Activating Distributed Visual Region within LLMs for Efficient and Effective Vision-Language Training and Inference

Anonymous ACL submission

Abstract

Large Vision-Language Models (LVLMs) typically learn visual capacity through visual instruction tuning, involving updates to both a projector and their LLM backbones. Inspired by the concept of a visual region in the human brain, we investigate the existence of an analogous *visual region* within LLMs that functions as a cognitive core, and explore the potential of efficient training of LVLMs via selective layers tuning. Using Bunny-Llama-3-8B-V for detailed analysis and other three LVLMs for validation across diverse visual and textual tasks, we find that selectively updating 25% of LLMs layers, when sparsely and uniformly distributed, can preserve nearly 99% of visual performance and maintain or improve textual task results, while effectively reducing training time. Based on this targeted training approach, we further propose a novel visual region-based pruning paradigm, removing non-critical layers outside the visual region, which can achieve minimal performance loss. This study offers an effective and efficient strategy for LVLM training and inference by activating a layer-wise visual region within LLMs, which proves consistently effective across different models.

1 Introduction

Large Vision-Language Models (LVLMs) (Li et al., 2023c; Zhu et al., 2023; Bai et al., 2023; Liu et al., 2024) have emerged as an increasing research interest for interpreting and interacting with the world through both visual and linguistic channels. Existing LVLMs generally utilize advanced Large Language Models (LLMs), like FlanT5 (Chung et al., 2022) and Vicuna (Chiang et al., 2023), as their cognitive core, and align visual features from visual encoders with LLMs' knowledge and reasoning abilities. This alignment has demonstrated remarkable performance across diverse visual tasks (Lu et al., 2022; Liu et al., 2023b; Fu et al., 2024).

LVLMs are primarily trained through visual instruction tuning (Liu et al., 2023a), which involves training both a projector and LLMs on visual instruction datasets, with optional updates to the visual encoder. Despite its efficacy, fully tuning all LLMs layers remains computationally costly, even when using efficient strategies like Low-Rank Adaptation (LoRA) (Hu et al., 2021) and its quantized variant (QLORA) (Dettmers et al., 2024). Additionally, extensive multimodal training risks degrading LLMs' pre-trained linguistic knowledge and reasoning capabilities (Dai et al., 2024; Agrawal et al., 2024), as evidenced by LVLMs' increased perplexity on textual tasks compared to their LLM backbone in the purple section of Fig. 1.

Inspired by specialized visual regions in the human brain (Grill-Spector and Malach, 2004) and LLMs' brain-like versatility across tasks, we propose an analogous concept of a *visual region* within LLMs. We hypothesize that visual alignment to LLMs can only activate this specific *visual region* while preserving LLMs' core language abilities, potentially manifesting as a layer-wise structure considering layer redundancy in LLMs (Men et al., 2024; Gromov et al., 2024). We further detailedly analyze LVLMs' layer redundancy in Fig. 1 (green part), shows that reverting certain layers of a LVLM to its backbone LLM' parameters minimally impacts downstream visual performance. This suggests certain layers within LLMs are non-essential for visual tasks, thereby supporting our hypothesis.

Although layer-wise freezing techniques (Zhang et al., 2024b) enable efficient LLM fine-tuning by adapting later layers for specific language tasks, they cannot be directly applied to visual tasks. Because visual alignment requires visual perception capabilities beyond textual understanding and reasoning. While Zhang et al. (2024a) propose parameter localization for visual tasks, it remains highly task-specific and data-dependent, limiting its generalizability to versatile multimodal learning and

Model Variants	Visual		Textual	
	OCRVQA	DocVQA	WikiText	Pile-10k
LLaVA	2.43	30.55	11.44	29.58
LLaVA$_\mathbf{r}$ (layer 0∼7)	1.87	38.49 [↑]	11.37 [↑]	29.19 [↑]
LLaVA$_\mathbf{r}$ (layer 8∼15)	1.93	32.35 [↑]	11.38 [↑]	29.21 [↑]
LLaVA$_\mathbf{r}$ (layer 16∼23)	2.18	16.47	11.35 [↑]	29.33 [↑]
LLaVA$_\mathbf{r}$ (layer 24∼31)	2.11	17.47	11.36 [↑]	29.27 [↑]
Vicuna (all layers)	80.75	175.10	11.32	28.38

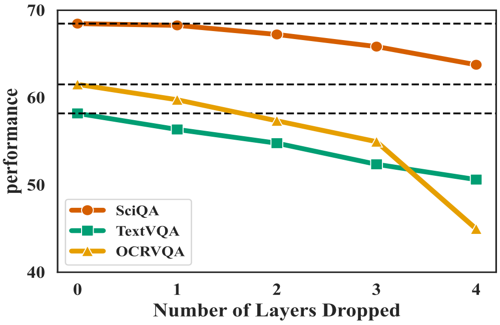

Figure 1: **Left:** Perplexity of LLaVA with selected layers (in parentheses) reverted to Vicuna parameters on visual and textual tasks. Arrows indicate perplexity increases relative to LLaVA (visual tasks) and Vicuna (textual tasks). (1) Perplexity increases in textual tasks after multimodal training compared to the LLM backbone, indicating multimodal training compromises LLMs' linguistic abilities. (2) Perplexity decreases in visual tasks reverting certain layers (e.g., reverting layers 16–23 or 24-31 in LLaVA), suggesting these layers are redundant. **Right:** Accuracy of LLaVA-1.5-7B when pruning certain layers based on angular distance scores (Gromov et al., 2024).

neglecting the preservation of linguistic capabilities. To bridge this gap, we identify a general-purpose visual region within LLMs for efficient LVLM training across diverse tasks without diminishing linguistic performance. Specifically, we aim to investigate two key questions: (1) Where is this visual region located within LLMs? (2) What is the necessary scale of layers in this visual region to ensure effective and efficient LVLMs training?

To this end, we embark on empirical experiments with Bunny-Llama-3-8B-V (He et al., 2024) across diverse visual tasks. Our findings indicate that **sparsely and uniformly distributed layers within LLMs are the optimal position for visual learning** while simultaneously preserving textual performance. This strategic visual region selection also outperforms previous layer importance strategies. Notably, **updating only 25% of layers achieves nearly 99% performance on visual tasks** while effectively saving training time. We further validate this conclusion with LLaVA-1.5-7B, LLaVA-1.5-13B (Liu et al., 2023a) and Bunny-Phi3-mini-4B-V, demonstrating its consistent applicability across varying models and parameter scales. Specifically, we achieve time reductions of nearly 23% for LLaVA-1.5-7B and LLaVA-1.5-13B, and 12% for Bunny-Llama-3-8B-V.

Additionally, as shown in Figure 1 (right), we find that commonly used layer-pruning strategies are ineffective for LVLMs, with even minimal layer removal causing significant performance degradation. In response, we propose a visual region-based pruning paradigm that selectively prunes less-important layers outside the visual region after targeted training. Specifically, we follow the angular distance based layer importance strategy (Gromov et al., 2024) outside the visual region, and experimental results demonstrate that our paradigm is effective to minimizes performance decline. Overall, our work highlights promising potential for more efficient LVLMs training and inference. Notably, our approach is flexibly complementary to other efficient training techniques, such as LoRA, as demonstrated in our experiments.

2 Preliminary of LVLMs

2.1 Model Architecture

Mainstream LVLMs consist of three components: a LLM, a visual encoder, and a projector or connection module, aim to effectively leverage the capabilities of both the pre-trained visual model and LLMs. The visual encoder extracts visual features from images, commonly utilizing pre-trained models such as CLIP ViT-L/14 (Radford et al., 2021). The connection module then projects these extracted features into word embedding space understandable by LLMs, commonly employing techniques such as linear projection (Tsimpoukelli et al., 2021), Q-former (Li et al., 2023c), or cross-attention layers (Alayrac et al., 2022). This enables LVLMs based on LLMs cores, like Vicuna (Chiang et al., 2023), FlanT5 (Chung et al., 2022), and LLaMA (Touvron et al., 2023) to process visual information in a similar manner as text.

2.2 Model Training

The training of LVLMs can be broadly divided into two phases: pre-training and supervised fine-tuning. Unlike LLMs, both phases utilize supervised image-text pairs for visual instruction tuning. Pre-training primarily uses large-scale captioning instruction data, guiding the model to briefly describe images. This phase enables the model to interpret image content, usually with LLMs' weights frozen and the visual encoder optionally updated. Some works such as Qwen-VL (Bai et al., 2023), expand the pre-training to include additional tasks like visual question answering, updating the LLM component accordingly. Supervised fine-tuning employs high-quality instruction data to enhance the LVLMs' ability to following diverse visual instructions and engaging in conversations. The visual encoder in this stage is typically kept static while the LLMs are tuned. During both stages, the projector is consistently updated, ensuring the model effectively bridges visual and textual data.

3 Experimental Setup

In this study, we conduct empirical experiments on Bunny-Llama-3-8B-V to investigate our hypothesis regarding the existence of a specific *visual region* within LLMs (Sec. 4.1∼ 4.3), and apply our findings on LLaVA-1.5-7B, LLaVA-1.5-13B and Bunny-Phi3-mini-4B-V to validate its general applicability across different models (Sec. 5.1).

3.1 LVLM Implementation

We employ Bunny-Llama-3-8B-V for investigation, which builds upon the 32-layer Llama3-8B (Touvron et al., 2023), and LLaVA-1.5-7B/13B, built on the 32/40-layer Vicuna-1.5-7B/13B (Chiang et al., 2023), Bunny-Phi3-mini-4B-V based on 32-layer Phi-3-mini for validation. Since the LLM components remain frozen during pre-training, we focus on the supervised fine-tuning stage using 695K and 665K language-image instruction-following instances for Bunny and LLaVA. Considering computational constraints, we use LoRA (Hu et al., 2021), highlighting that *our approach is complementary to other efficient training methods*. Additional implementation details are available in the Appendix.

3.2 Evaluation Tasks

Our investigation spans 10 visual tasks involving both perception and cognition, to comprehensively evaluate models and examine our hypothesis.

Visual perception tasks assess models' ability to interpret and understand surface-level visual features, like object identification and scene recognition, mirroring human sensory perception process. (1) OCRVQA (Mishra et al., 2019): VQA by reading text in images through optical character recognition (OCR). We follow(Bai et al., 2023) for accuracy calculation on the test set, allowing a margin of error. (2) DocVQA (Mathew et al., 2021): VQA by interpreting document images. We use the same evaluation method and metric as OCRVQA on the validation set. (3) RefCOCOg (Yu et al., 2016): A variant of RefCOCO (Kazemzadeh et al., 2014) featuring more complex object referring expressions. We assess the reference expression generation on the test set using Intersection over Union metric. (4) TDIUC (Kafle and Kanan, 2017): covering 12 categories, primarily perception tasks (e.g., object presence, counting, recognition) with some cognition tasks (e.g., positional reasoning, affordance). Accuracy is measured on the validation set.

Visual cognition tasks require deeper reasoning based on visual stimuli, drawing on prior knowledge and decision-making abilities learned within LLMs, mirroring human cognitive thinking and manipulation. (5) MMBench (Liu et al., 2023b): focuses on cognition tasks, with some fine-grained perception tasks requiring knowledge and reasoning. For model variant comparison, we report accuracy on the dev subset instead of submitting to the evaluation server. (6) GQA (Hudson and Manning, 2019): real-world visual reasoning and compositional question answering. (7) ScienceQA (Lu et al., 2022): sourced from elementary and high school science curricula, requiring external knowledge and reasoning. We evaluate only image-based questions. (8) TextVQA (Singh et al., 2019): requiring reasoning about text in images. (9) MMMU (Yue et al., 2024): covering math, science, and commonsense reasoning with accuracy calculated. (10) SEED-IMG: The image-based QA from SEED-Bench (Li et al., 2023a).

4 Visual Region Investigation

We first analyze the position and scale of the layerwise-structure vision region within its LLM core on Bunny-Llama-3-8B-V, to answer the following two questions.

4.1 Where are visual region layers located within LLMs for effective visual learning?

To demonstrate the optimal positioning of the visual region in LLMs for effective and efficient visual learning, we re-train Bunny-Llama-3-8B-V by

Model Version	OCRVQA	DocVQA	RefCOCOg	TDIUC	MMBench	GQA	ScienceQA	TextVQA	MMMU	SEED-IMG	Avg
All layers	64.26%	29.45%	50.12%	83.84%	74.74%	64.29%	79.28%	62.11%	40.6%	73.13%	62.18%
Heuristic Selections											
Sparse & Uniform	62.65%	29.51%	**48.33%**	83.68%	**73.88%**	**63.68%**	78.78%	62.43%	42.1%	72.61%	**61.82%**
Consecutive Lower	61.38%	22.47%	46.49%	83.27%	73.63%	62.33%	75.26%	62.26%	**42.6%**	72.66%	60.24%
Consecutive Lower-middle	62.54%	26.13%	48.17%	83.77%	72.51%	62.81%	77.14%	60.96%	38.8%	72.16%	60.50%
Consecutive Upper-middle	62.32%	28.06%	43.12%	83.40%	70.27%	61.28%	**78.83%**	59.33%	38.3%	70.45%	59.54%
Consecutive Top	60.48%	26.47%	39.92%	83.22%	67.96%	60.30%	77.54%	58.71%	37.0%	71.00%	57.26%
Hybrid Top-Lower	57.63%	**29.76%**	41.79%	83.26%	72.25%	62.71%	77.99%	62.74%	40.1%	72.59%	60.09%
Importance-based Selections											
Image Attention Score	63.65%	24.53%	43.62%	**83.90%**	72.59%	62.82%	77.59%	61.99%	39.3%	72.29%	60.23%
Parameter Change Ratio	**63.94%**	26.94%	47.67%	83.88%	73.54%	63.21%	78.68%	61.73%	42.0%	72.85%	61.45%
Block Influence Score	62.38%	28.45%	46.37%	83.73%	71.13%	61.93%	77.34%	59.93%	38.9%	71.66%	60.18%
Multimodal BI Score	61.48%	28.80%	46.68%	83.74%	73.02%	63.23%	77.24%	62.23%	41.0%	72.25%	60.97%
Angular Distance	60.95%	27.71%	46.74%	83.49%	73.88%	62.11%	77.14%	**62.76%**	39.9%	**73.01%**	60.77%

Table 1: Performance comparison of Bunny-LLaMA-3-8B-V tuned with ***different layer selection methods (8 layers)***. Bold numbers indicate the best performance in each column (excluding "all layers").

updating 25% of layers (8 layers) [1] under various selection configurations. As pre-training does not involve LLM optimization, we focus on supervised fine-tuning, starting from the pre-trained checkpoint. We specifically explore different positional selection strategies as detailed below.

- **Heuristic Layer Selection** (1) We intuitively hypothesize that tuning *sparsely and uniformly distributed layers* (0,4,8,12,18,22,26,30) preserves LLMs' existing knowledge and reasoning abilities while enabling visual learning. (2) We experiment with tuning *consecutive 8-layer blocks* at different positions in LLMs: lower layers (0∼7), lower-middle layers (8∼15), upper-middle layers (16∼23), and top layers (24∼31), with the latter being a common practice of **efficient domain-specific fine-tuning** (Liao et al., 2024). (3) We test a hybrid of lower and top layers (0∼3, 28∼31).

- **Importance-based Layer Selection** We compare layer selection strategies based on varying importance metrics. (1) *Image Attention Score*: We compute the average attention score on all image tokens at each layer to gauge the layer's affinity for image information. The top 8 layers with the highest scores are selected (1,2,3,4,5,27,29,31). (2) *Parameter Change Ratio* (Zhao et al., 2023): 8 layers with the highest relative parameter change ratios (averaged all parameters in each layer) in Bunny-Llama-3-8B-V compared to its backbone Llama are selected (0,2,9,12,23,24,25,26). (3) *Block Influence (BI) Score* (Men et al., 2024): Using Flickr30k dataset, we calculate hidden state transformations at each layer as the BI score, and select 8 layers with the highest scores (12,15,18,25,27,29,30,31). (4) *Multimodal BI Score*: We propose a multimodal variant that average hidden state transformations respectively of visual tokens and textual tokens, and select 8 layers with highest scores (0,1,2,3,4,5,9,31). (5) *Angular Distance Score* (Gromov et al., 2024): The top 8 layers with the highest angular distances between consecutive layer inputs are selected (0,1,2,3,5,6,7,8). Detailed calculations for these metrics are provided in Appendix A.

The results are shown in Table 1. We observe that tuning sparsely and uniformly distributed layers achieves the best overall performance across perception and cognition tasks, closely matching the all-layers upper bound. In contrast, consecutive layers generally underperform, likely due to limited diversity in similar representations across adjacent layers (Kornblith et al., 2019), which hinders adaptability to various tasks. This further underscores the superiority of sparsely and uniformly distributed layers. Notably, tuning top layers yields the worst performance, deviating from the conventional practice in domain-specific fine-tuning, where the last few layers are typically adjusted for downstream tasks (Liao et al., 2024). This highlights a significant distinction between adapting to new modalities and new downstream domains.

While some importance-based selections, such as parameter change ratio, yield close performance, all importance-based methods operate post-hoc that require a fully trained model to compute im-

[1] We use the 8-layer configuration as a testbed for its balance of efficiency and effectiveness.

Model Scale	OCRVQA	DocVQA	RefCOCOg	TDIUC	MMBench	GQA	ScienceQA	TextVQA	MMMU	SEED-IMG	Avg
32 layers	**64.26%**	29.45%	**50.12%**	83.84%	74.74%	**64.29%**	**79.28%**	62.11%	40.6%	**73.13%**	**62.18%**
16 layers	62.42%	26.43%	49.15%	**84.04%**	74.83%	64.10%	78.93%	**62.96%**	**42.6%**	72.75%	61.82%(99.42%)
8 layers	62.65%	29.51%	48.33%	83.68%	73.88%	63.68%	78.78%	62.43%	42.1%	72.61%	61.78%(99.36%)
6 layers	62.25%	**29.76%**	47.71%	84.01%	**75.00%**	62.93%	77.54%	62.92%	40.6%	72.67%	61.55%(98.99%)
4 layers	62.40%	28.89%	46.00%	83.99%	73.71%	62.66%	77.69%	62.74%	39.2%	72.14%	60.94%(98.01%)
2 layers	57.96%	28.49%	44.67%	83.15%	72.68%	61.00%	78.48%	60.35%	40.8%	72.35%	60.00%(96.49%)
1 layer	53.68%	24.33%	38.47%	82.92%	68.64%	59.19%	77.69%	58.32%	37.4%	70.69%	57.14%(91.89%)

Table 2: Performance comparison of Bunny-Llama-3-8B-V fine-tuned with ***different numbers of layers***. Bold numbers represent the best performance in each column. Values in parentheses denotes the percentage relative to the performance achieved by tuning all layers.

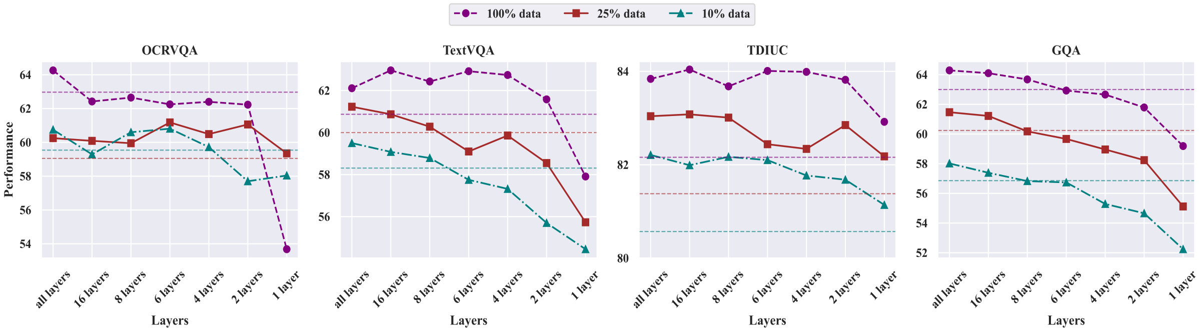

Figure 2: Performance variation of the re-trained Bunny-Llama-3-8B-V model across ***different training data scales*** during the supervised fine-tuning stage, with tuning varying number of layers. Dashed lines indicate 98% of the performance achieved by tuning all layers with the corresponding training data scale.

portance metrics for layer selection. This makes them primarily suitable for inference and applying them during LVLM training incurs significantly higher computational costs. In contrast, our heuristic method is training-free, allowing for greater flexibility and direct transferability across different models, enhancing its practical applicability. We compare importance-based selections to show that our sparsely and uniformly distributed layers even outperform these post-hoc strategies and also simplify the process.

4.2 What is the necessary scale of layers for effective and efficient LVLMs training?

To investigate the necessary scale of this visual region to enable LVLMs to receive visual signals and align with linguistic features, we re-train Bunny-Llama-3-8B-V by updating varying number of layers. We respectively experiment with configurations of 32, 16, 8, 6, 4, 2 and 1 layers, with all selected layers uniformly distributed across all layers [2]. This selection strategy is based on our finding that sparsely and uniformly distributed layers are the optimal position for effective visual learning.

The results of tuning varying scales of layers on visual perception and cognition tasks are summarized in Table 2. Tuning 20∼25% of the layers (6 and 8 layers) retains approximately 98% of the performance achieved by tuning all LLMs layers of Bunny-Llama-3-8B-V, with 25% (8 layers) preserving up to 99%. However, updating fewer than 4 layers leads to a significant performance drop, particularly in perception tasks that heavily relies on visual interpretation, highlighting the necessity of tuning at least 12.5% of the layers (4 layers) for effective visual alignment.

4.3 Trend between Data Size and Visual Region Scale

We further explore the trend between data size and the optimal layer count for effective visual instruction tuning. Using random subsets of 100%, 25% and 10% from a pool of 695K visual instruction-following instances, we tune Bunny-Llama-3-8B-V with varying numbers of layers following the

[2] Specifically, we select all even-numbered layers for the 16-layer configuration; layer 0, 4, 8, 12, 18, 22, 26, 30 for 8-layer; layer 0, 6, 12, 18, 24, 30 for 6-layer; and layer 0, 10, 20, 30 for 4-layer (Experiments show that layer 30 or 31 yields comparable results, and all odd-numbered selections perform slightly worse). Since 2-layer and 1-layer selection can not be uniform, we have tested various configurations and adopted the best-performing strategy: layer 0 and 31 for 2-layer, and layer 31 for 1-layer based on highest block influence scores.

Model Scale	OCRVQA	DocVQA	RefCOCOg	TDIUC	MMBench	GQA	ScienceQA	TextVQA	MMMU	SEED-IMG	Avg
LLaVA-1.5-7B											
32 layers	61.51%	19.46%	49.01%	83.40%	66.67%	62.98%	68.47%	58.19%	35.3%	67.52%	57.25%
16 layers	64.01%	20.75%	48.02%	83.47%	64.00%	62.43%	67.53%	58.27%	35.4%	67.22%	57.11%(99.76%)
8 layers	62.19%	21.10%	47.71%	83.10%	63.92%	61.60%	68.17%	57.35%	34.6%	67.23%	56.70%(99.04%)
6 layers	61.39%	22.84%	46.54%	83.31%	61.77%	61.08%	68.32%	56.19%	33.2%	65.69%	56.04%(97.87%)
4 layers	63.28%	21.01%	43.47%	83.14%	60.82%	60.48%	67.97%	54.48%	33.8%	64.08%	55.25%(96.51%)
2 layers	54.54%	19.10%	41.90%	81.47%	57.22%	57.38%	65.84%	53.27%	33.7%	63.19%	52.76%(92.16%)
1 layer	53.16%	16.96%	33.29%	81.20%	51.89%	55.83%	64.50%	45.51%	30.1%	57.64%	49.01%(85.61%)
LLaVA-1.5-13B											
40 layers	67.60%	25.19%	50.26%	83.61%	68.38%	63.29%	71.64%	60.21%	37.2%	68.70%	59.61%
10 layers	65.17%	23.56%	48.27%	83.57%	66.58%	62.01%	70.75%	59.13%	36.9%	67.39%	58.33%(97.85%)
9 layers	66.47%	23.65%	49.29%	83.74%	65.61%	62.31%	72.14%	59.71%	37.7%	67.29%	58.80%(98.64%)
Bunny-Phi3-mini-4B-V											
32 layers	63.62%	29.19%	48.07%	83.69%	72.94%	62.35%	76.75%	60.64%	42.4%	72.09%	61.17%
8 layers	61.96%	27.21%	46.95%	83.11%	71.74%	61.38%	75.71%	59.69%	42.3%	71.53%	60.16%(98.35%)

Table 3: Performance of LVLMs with varying LLM backbones and parameter scales tuned with different numbers of layers. Values in parentheses denotes the percentage relative to the performance achieved by tuning all layers.

same selection strategy as the full dataset. We report the performance trends across four datasets, OCRVQA, TextVQA, TDIUC and GQA. As shown in Figure 2, tuning 25% of the layers consistently achieves over 98% of full performance across different data sizes while reducing training time. This approach offers a resource-efficient pathway for optimizing hyperparameters and training data selection by tuning such a visual region before finalizing the model with all layers. Moreover, even with smaller datasets, tuning fewer than 4 layers still results in notable performance declines.

5 Further Analysis

5.1 Generalizability Validation

To validate our findings of the visual region beyond Bunny-Llama-3-8B-V, we take LLaVA-1.5-7B, LLaVA-1.5-13B and Bunny-Phi3-mini-4B-V as additional testbeds to assess the generalizability across LVLMs with different LLM backbones and parameter scales. Following the setup in Sec. 4.2, we re-train these models with different number of layers that are sparsely and uniformly distributed within their respective backbones, including Vicuna-1.5-7B, Vicuna-1.5-13B and Phi-3-mini-4B (Abdin et al., 2024). Results presented in Table 3 show that under our visual region positioning strategy, tuning approximately 25% of the layers consistently yield 98% of the full performance. This demonstrates that our approach generalizes effectively across varying LVLMs.

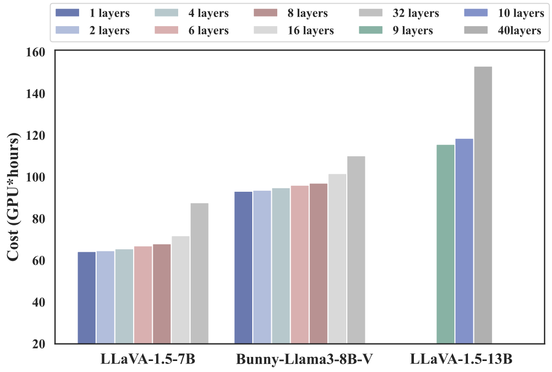

Figure 3: Computational costs for tuning LLaVA-1.5-7B, Bunny-Llama-3-8B-V, and LLaVA-1.5-13B with different number of layers using LoRA.

5.2 Computational Cost

To demonstrate the efficiency of visual region-based tuning, we report the computational costs associated with tuning different numbers of layers across various models using the LoRA strategy. For fair comparison across setups with different numbers of GPUs (specifically A800 GPUs in this analysis), we compute the product of the number of GPUs and running hours as a measure of computational cost. From Figure 3, Table 2 and Table 3, tuning a visual region comprising up to 25% of layers (8 layers for LLaVA-1.5-7B and Bunny-Llama3-8B-V, 10 layers for LLaVA-1.5-13B) can achieve 98% of full performance while achieving substantial reductions in computational overhead. Specifically, we reduce training time by 23% for LLaVA models and 13% for Bunny. These results highlight that the effectiveness of visual region-based tuning in training LVLMs efficiently with minimal performance trade-offs. Moreover, this relative reduction in computational cost would be more significant as dataset and model sizes scale.

5.3 Evaluation of Textual Tasks

As highlighted in (Dai et al., 2024; Agrawal et al., 2024) and illustrated in Figure 1, multi-modal training risks degradation of LLMs' pre-trained linguistic knowledge and reasoning capabilities. To verify whether training our sparsely and uniformly distributed visual region affects the model linguistic capacity, we extend our evaluation to four text-only question answering datasets, MMLU (Hendrycks et al., 2020), C-Eval (Huang et al., 2023), CMMLU (Li et al., 2023b), and BIG-bench-Hard (Suzgun et al., 2022), covering diverse topics and fields. We use "Answer with the option's letter from the given choices directly" as the prompts for the first three and "Please answer this question in a word or phrase" for BIG-bench-Hard, and allow models to provide explanations alongside its responses. We adopt a five-shot prompting strategy for MMLU, C-Eval and CMMLU, and a zero-shot strategy for BIG-bench-Hard.

Model Version	MMLU	BIG-Bench-H	C-Eval	CMMLU
Bunny-LLaMA3-8B-V				
Fully-trained (32layers)	60.27%	30.93%	45.84%	45.68%
Partial-trained (8layers)	63.36%	31.50%	49.70%	48.39%
LLM-Backbone	66.01%	57.93%	50.52%	50.70%
LLaVA-1.5-7B				
Fully-trained (32layers)	50.52%	26.85%	38.34%	37.27%
Partial-trained (8layers)	50.74%	31.64%	39.08%	37.71%
LLM-Backbone	49.78%	29.33%	38.78%	36.60%

Table 4: Performance on text-only tasks. The LLm backbones of Bunny-LLaMA3-8B-V and LLaVA-1.5-7B are respectively LLaMA3-8B and Vicuna-1.5-7B.

As shown in Table 4, fully-trained LVLMs generally exhibit decreased performance on text-only tasks compared to their LLM backbones, particularly with more powerful LLaMA3-8B and on the challenging BIG-bench-Hard dataset. In contrast, our selectively trained LVLMs minimally compromise models' linguistic capacity, which consistently outperform fully-trained LVLMs, and sometimes even surpass their LLMs backbones. These results support our hypothesis that positioning the visual region strategically by tuning sparsely and uniformly distributed layers better preserves LLMs' linguistic knowledge and reasoning capabilities, whereas full training may cause minor disruptions.

6 Visual Region-Based Layer Pruning

Beyond layer selection for efficient LVLMs training, we explore whether the visual region can also benefit LVLM efficient inference. Although layer pruning techniques (Men et al., 2024; Ma et al., 2023) have been widely developed for LLM inference, they prove ineffective for LVLMs. As shown in Figure 1 (right), minimal layer removal causing significant performance degradation on visual tasks even using advanced angular distance based pruning strategy (Gromov et al., 2024).

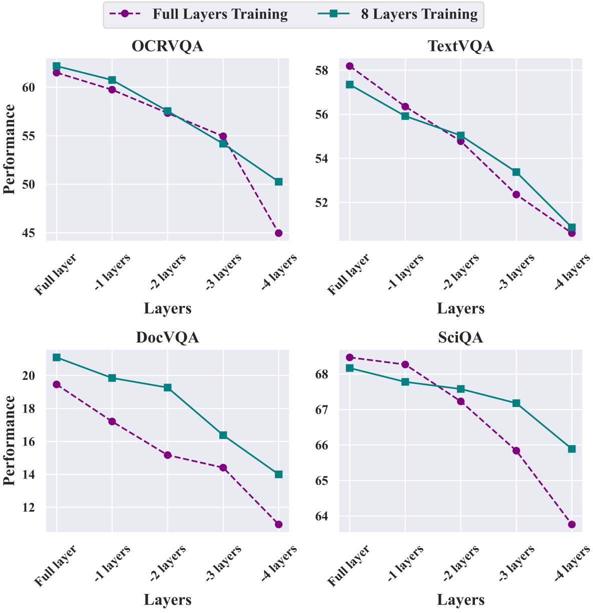

Figure 4: Results of pruning LLaVA-1.5-7B using angular distance-based strategy with 0∼4 layers removed. Dashed lines represent pruning applied to the fully trained model while solid layers denote our visual region-based pruning within the targeted trained model.

Building on our visual region targeted training, we propose a visual region-based pruning paradigm that selectively prunes less-important layers outside the visual region after training. Specifically, we follow the angular distance based layer importance metric and select 0∼4 layers with the lowest angular distance outside the visual region. We do not evaluate pruning beyond this range as removing additional layers in LVLMs would lead to significant performance collapse. We evaluate this approach on LLaVA-1.5-7B across four datasets: OCRVQA, TextVQA, DocVQA and SciQA. As shown in Figure 4, our paradigm generally maintain higher performance, especially when pruning 3∼4 layers, even though the visual region targeted trained model performs slightly worse than fully trained model without pruning. This result demonstrates that our paradigm effectively minimizes performance degradation compared to pruning in full-layer trained LVLMs, serving as an initial exploration into LVLM-specific pruning strategies.

7 Related Work

7.1 Efficient Training and Inference

Recent research community has witnessed an emergent interest in LLMs (Touvron et al., 2023; Chiang et al., 2023) and LVLMs (Li et al., 2023c; Zhu et al., 2023; Bai et al., 2023; Liu et al., 2024) due to their remarkable ability to interpret and interact with the world via linguistic and visual channels. With the sustainably increased scale of LLMs and LVLMs, training or inference using all model parameters are cost for practical deployment. There are numerous techniques for efficient model training and inference. For instance, quantization reduce the memory footprint of models by decreasing the precision of model weights (Dettmers et al.; Dettmers and Zettlemoyer, 2023; Xiao et al., 2023). Low rank adapters enable cost-effective fine-tuning by updating only a small subset of the adapter parameters (Hu et al., 2021; Karimi Mahabadi et al., 2021).

Moreover, LLMs exhibit significant redundancy at the layer level, making training or inference with all layers computationally wasteful, and this redundancy is established for LVLMs as well, where LLMs serve as the core cognitive brain for visual learning. In responding, layer-wise freezing techniques (Zhang et al., 2024b; Liang et al., 2023; Pan et al., 2024) and layer pruning strategies (Men et al., 2024; Ma et al., 2023; Gromov et al., 2024) are proposed to enable efficient LLM fine-tuning and inference. However, they are designed for LLMs and fail to generalize effectively to visual learning, often resulting in substantial performance degradation. While Zhang et al. (2024a) introduce parameter localization for visual tasks, their approach is highly task-specific and data-dependent, limiting its applicability to versatile visual learning and neglecting the preservation of linguistic capabilities. In contrast, we propose a more efficient layer-selected strategy for LVLMs training and inference.

7.2 Functional Regions in LLMs

The existing literature on cognitive science and brain localization indicates that different regions among the human brain are dedicated to specific functions (Fedorenko and Varley, 2016), such as frontotemporal language processing region localized by Scott et al. (2017). Grill-Spector and Malach (2004) highlight the existence of visual regions in neuroscience (Grill-Spector and Malach, 2004). These insights have inspired an analogy with LLMs, increasingly viewed as cognitive core for remarkable performance across diverse tasks, mirroring the human brain's functionality in terms of overall planning and processing. For example, Aw et al. (2023) propose that LLMs can be aligned to the human brain through instruction-tuning. Building upon this parallel, Zhao et al. (2023) unveil a core linguistic region within LLMs, accounting approximately 1% of the model's parameters. Li and Li (2024) identify a duality between Tulving's synergistic ecphory model (SEM) of memory and LLMs' emergent abilities. Drawing inspiration from these, our research focuses on defining a vision region within LLMs, suggesting a more effective and efficient pipeline to optimizing LVLMs for visual tasks.

8 Conclusion

In this study, we introduce an effective and efficient training paradigm for LVLMs by activating a specific *visual region* within LLMs. This offers a new pipeline for advancing LVLMs which first identify such *visual region* using limited data followed by efficient continual training. Specifically, we investigating the necessity of tuning all layers within LLM cores, and propose the concept of a specialized *visual region* within LLMs. We conduct extensive empirical experiments with Bunny-LLaMA-3-8B-V, covering a range of visual and textual tasks. Our results reveal that selectively updating no more than 25% of sparsely and uniformly layers, can preserve nearly 99% visual performance, while also yielding comparable results in textual tasks. This targeted LVLMs' training approach is consistently effective for different models and parameter scales, effectively reducing training time by 23% for LLaVA models and 12% for Bunny-LLaMA-3-8B-V. Additionally, we propose a visual region-based layer pruning by strategy removing non-critical layers outside the visual region and achieve minimal performance drop. Overall, our work presents a promising pathway for more efficient LVLMs training and inference, while complementing existing efficient training methods.

Limitations

Experimented Models Our work primarily focuses on LLaVA-1.5 family, Bunny-LLama3-8B-V and Bunny-Phi3-mini-4B-V to demonstrate the effectiveness and efficiency of our proposed training and inference paradigms for LVLMs. Future work will expand to a broader range of models to further validate the generalizability of our approach. Additionally, we will explore extensions to other modalities such as speech, and investigate the existence of other modality-specific regions to develop more versatile and scalable multimodal models.

Sparse Architectures While our approach effectively reduces training and inference costs by activating the *visual region*, it currently operate in a layer-wise dense manner. Future efforts will focus on integrating our method with sparse model architectures to optimize *visual region* activation. For example, explore routing mechanisms targeting modality-specific partitions within models to implement sparse mixture-of-expert architectures with specialized functional areas, analogous to the functional regions of the human brain.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Devendra Chaplot, Jessica Chudnovsky, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, et al. 2024. Pixtral 12b. *arXiv preprint arXiv:2410.07073*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Khai Loong Aw, Syrielle Montariol, Badr AlKhamissi, Martin Schrimpf, and Antoine Bosselut. 2023. Instruction-tuning aligns llms to the human brain. *arXiv preprint arXiv:2312.00575*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Wenliang Dai, Nayeon Lee, Boxin Wang, Zhuolin Yang, Zihan Liu, Jon Barker, Tuomas Rintamaki, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nvlm: Open frontier-class multimodal llms. *arXiv preprint arXiv:2409.11402*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm. int8 (): 8-bit matrix multiplication for transformers at scale, 2022. *CoRR abs/2208.07339*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning*, pages 7750–7774. PMLR.

Evelina Fedorenko and Rosemary Varley. 2016. Language and thought are not the same thing: evidence from neuroimaging and neurological patients. *Annals of the New York Academy of Sciences*, 1369(1):132–153.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Kalanit Grill-Spector and Rafael Malach. 2004. The human visual cortex. *Annu. Rev. Neurosci.*, 27:649–677.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. 2024. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*.

Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. 2024. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36:62991–63010.

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

Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. 2015. Guiding long-short term memory for image caption generation. *Preprint*, arXiv:1509.04942.

Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In *Proceedings of the IEEE international conference on computer vision*, pages 1965–1973.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

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

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023a. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023b. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Jitang Li and Jinzheng Li. 2024. Memory, consciousness and large language model. *arXiv preprint arXiv:2401.02509*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023c. Blip-2: Bootstrapping language-image pretraining with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pages 20852–20867. PMLR.

Baohao Liao, Shaomu Tan, and Christof Monz. 2024. Make pre-trained model reversible: From parameter to memory efficient fine-tuning. *Advances in Neural Information Processing Systems*, 36.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2023b. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. Docvqa: A dataset for vqa on document images. *Preprint*, arXiv:2007.00398.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. Lisa: Layerwise importance sampling for memory-efficient large language model fine-tuning. *arXiv preprint arXiv:2403.17919*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Terri L Scott, Jeanne Gallée, and Evelina Fedorenko. 2017. A new fun and robust version of an fmri localizer for the frontotemporal language system. *Cognitive neuroscience*, 8(3):167–176.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Wenxuan Zhang, Paul Janson, Rahaf Aljundi, and Mohamed Elhoseiny. 2024a. Overcoming generic knowledge loss with selective parameter update. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24046–24056.

Yulin Zhang, Yanhua Li, and Junhan Liu. 2024b. Unified efficient fine-tuning techniques for open-source large language models.

Jun Zhao, Zhihao Zhang, Yide Ma, Qi Zhang, Tao Gui, Luhui Gao, and Xuanjing Huang. 2023. Unveiling a core linguistic region in large language models. *arXiv preprint arXiv:2310.14928*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Details of Layer Importance Metrics

To demonstrate the effectiveness of our heuristically identified sparsely and uniformly distributed visual region, we conduct a comparative analysis against several other layer importance metrics (originally for layer pruning) by selecting 8 layers and re-training Bunny-Llama-3-8B-V. Below are the details of how these metrics are calculated.

- **Block Influence (BI) Score** (Men et al., 2024): serves as an indicator of layer importance by measuring the transformation of hidden states. We utilize the Flickr30k dataset (Jia et al., 2015) to calculate the BI score for each layer within LVLMs. The BI score of i^{th} layers is calculated as following:

$$BI_i = 1 - \mathbb{E}_{X,t} \frac{X_{i,t}^T X_{i+1,t}}{\|X_i\|_2 \|X_{i+1}\|_2}$$

  where X_i represents the hidden states of the i^{th} layer and $X_{i,t}$ denotes the hidden states of the t^{th} token at the i^{th} layer. By calculating the average cosine similarity of token states before and after passing through a layer, we measure the change magnitude across all tokens.

- **Multimodal BI Score**: As the above method treats visual image and text as a single modality, we propose a multimodal variant that separately calculates the hidden state transformations of visual tokens and textual tokens, and take its average as a multimodal BI score. The Multimodal BI score of i^{th} layers is calculated as follows.

$$\begin{aligned} BI_i' = 1 - \frac{1}{2}(\mathbb{E}_{X,t} \frac{X_{i,t}^T X_{i+1,t}}{\|X_i\|_2 \|X_{i+1}\|_2} \\ + \mathbb{E}_{Y,l} \frac{Y_{i,l}^T Y_{i+1,l}}{\|Y_i\|_2 \|Y_{i+1}\|_2}) \end{aligned}$$

  $X_{i,t}$ and $Y_{i,l}$ respectively mean the hidden states of the t^{th} visual token and the l^{th} text token at the i^{th} layer. We calculate the cosine similarity of each modality tokens before and after passing through a layer, then average the results. This balances the token quantity across various modalities.

- **Parameter Change Ratio** (Zhao et al., 2023): We calculate the relative change ratio of the parameters in LVLM against its backbone LLM across each layer (by averaging all parameters within each layer). The parameter change ratio of i^{th} layers is calculated as follows:

$$R_i = \mathbb{E}_{\theta \in L_i, j} | \frac{\theta'_j - \theta_j}{\theta_j} |$$

  where θ_j and θ'_j respectively mean the j^{th} parameter of layer L_i in LLM and LVLM.

- **Angular Distance** (Gromov et al., 2024): We calculate the Angular Distance of the parameters in LVLM against its backbone LLM across each layer (by averaging all parameters within each layer). The Angular Distance of i^{th} layers is calculated as follows:

$$D_i = \frac{1}{\pi} \arccos \left(\frac{\theta'_j \cdot \theta_j}{\|\theta'_j\| \|\theta_j\|} \right)$$

  where θ_j and θ'_j respectively mean the j^{th} parameter of layer L_i in LLM and LVLM, $\| \cdot \|$ denotes the L^2-norm and the factor of $\frac{1}{\pi}$ is a constant.

- **Image Attention Score**: We calculate image attention score to measure each layer's affinity for image information. We utilize the DocVQA, OCRVQA, TDIUC, and RefCOCOg datasets, sampling 50 instances from each dataset to calculate the attention scores of the all image tokens for each layer within Bunny-Llama-3-8B-V. The heat map of image attention Score of every instances for each layers in Bunny-Llama-3-8B-V is showed in Figure 5. The image attention score of one instance in i^{th} layers A_i is calculated as follows:

$$A_i = \frac{\sum_{t=\text{k}}^{\text{k}+N_{\text{img}}-1} \sum_{h=1}^{H} \sum_{j=1}^{T} \text{Attn}[i][h, j, t]}{N_{\text{img}} H}$$

  where H represents the number of attention heads per layer and T denotes the total number of tokens at the i^{th} layer. N_{img} is the number of image tokens of the instance. The index range for the image tokens is from k to $k + N_{\text{img}} - 1$. While $\text{Attn}[h, j, t]$ means the attention score of the h^{th} attention head for the j^{th} token to the t^{th} token.

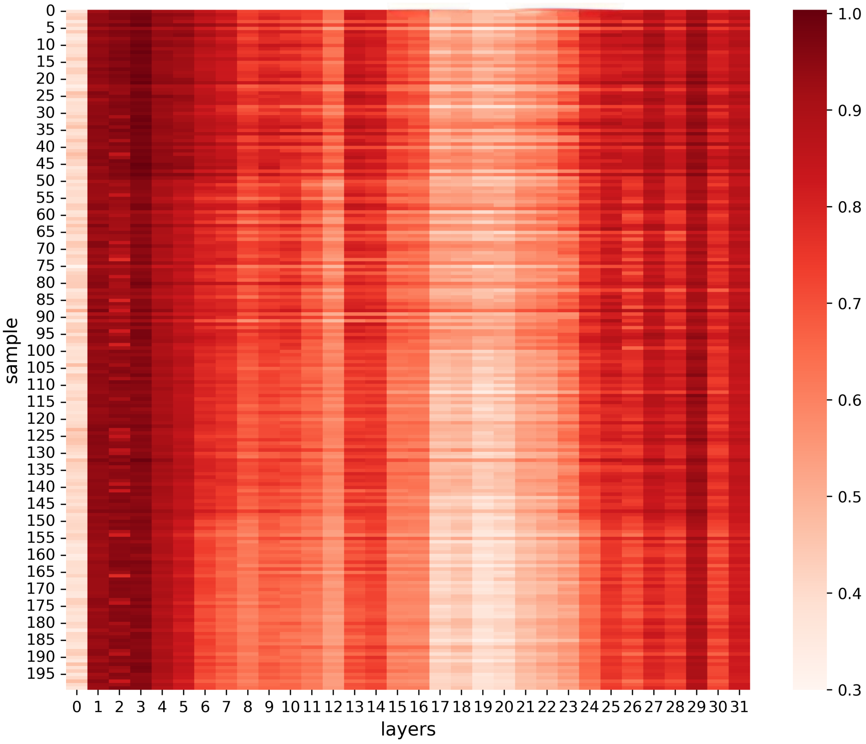

Figure 5: Visualization of Image Attention Scores for every instances across all layers