Multimodal Language Models See Better When They Look Shallower

Haoran Chen[1,2,3*], **Junyan Lin**[2,3*], **Xinghao Chen**[2,3], **Yue Fan**[4], **Jianfeng Dong**[1], **Xin Jin**[2,3], **Hui Su**[5], **Jinlan Fu**[6], **Xiaoyu Shen**[2,3†]
[1]Zhejiang Gongshang University
[2]Ningbo Key Laboratory of Spatial Intelligence and Digital Derivative
[3]Institute of Digital Twin, Eastern Institute of Technology, Ningbo
[4]Genmo.ai, [5]Meituan Inc., [6]National University of Singapore
haoranchr321@gmail.com, xyshen@eitech.edu.cn

Abstract

Multimodal large language models (MLLMs) typically extract visual features from the final layers of a pretrained Vision Transformer (ViT). This widespread deep-layer bias, however, is largely driven by empirical convention rather than principled analysis. While prior studies suggest that different ViT layers capture different types of information—shallower layers focusing on fine visual details and deeper layers aligning more closely with textual semantics, the impact of this variation on MLLM performance remains underexplored. We present the first comprehensive study of visual layer selection for MLLMs, analyzing representation similarity across ViT layers to establish shallow, middle, and deep layer groupings. Through extensive evaluation of MLLMs (1.4B–7B parameters) across 10 benchmarks encompassing 60+ tasks, we find that while deep layers excel in semantic-rich tasks like OCR, shallow and middle layers significantly outperform them on fine-grained visual tasks including counting, positioning, and object localization. Building on these insights, we propose a lightweight feature fusion method that strategically incorporates shallower layers, achieving consistent improvements over both single-layer and specialized fusion baselines. Our work offers the first principled study of visual layer selection in MLLMs, showing that *MLLMs can often see better when they look shallower*.

1 Introduction

Multimodal Large Language Models (MLLMs) extend the capabilities of traditional Large Language Models (LLMs) by enabling joint reasoning over both visual and textual inputs (Hong et al., 2024; Bai et al., 2023; Chen et al., 2024a; Shen et al., 2024). Typically, these models integrate a pretrained Vision Transformer (ViT) to extract image features, which are then projected into the language embedding space of an LLM. This architecture enables unified multimodal understanding and powers a wide range of applications, including robotic navigation, medical diagnostics, and visual question answering (Alayrac et al., 2022; Chen et al., 2024b; Bai et al., 2023; Tong et al., 2024a; Li et al., 2025).

While recent advancements have notably improved the language reasoning capabilities of MLLMs, the visual processing pipeline, particularly the selection of ViT layers for visual representation, remains insufficiently explored. In practice, MLLMs often default to using features from the deepest layers of ViT models. For instance, `Qwen-VL` (Bai et al., 2025) and `InternVL-6B v1.2/1.5` use the final layer of CLIP-ViT (Radford et al., 2021), while other `InternVL` variants select the fourth-to-last layer (Chen et al., 2024b). The LLaVA series (Liu et al., 2023b,a, 2024a) relies on the penultimate layer. However, these choices are largely heuristics rather than systematic evaluation (Yao et al., 2024; Jiang et al., 2023; Tong et al., 2024b; Cao et al., 2024; Li et al., 2026).

Previous work has shown that ViT layers encode a hierarchy of semantic information—from low-level edge detectors in shallow layers to abstract object representations in deeper layers (Gandelsman et al., 2024; Yao et al., 2024; Tong et al., 2024b). Yet, how these layer-wise representations affect MLLM performance remains poorly understood. This paper addresses this gap by systematically investigating *which ViT layers provide the most effective visual features for MLLMs*.

We begin by analyzing Layer-wise Representation Similarity (LRS) across CLIP-ViT's hidden states using cosine similarity, revealing three semantically coherent layer groups: shallow (layers 1–12), middle (13–20), and deep (21–24) (Fig. 1). This categorization provides a foundation for structured layer selection and fusion.

Building upon this foundation, we first systemati-

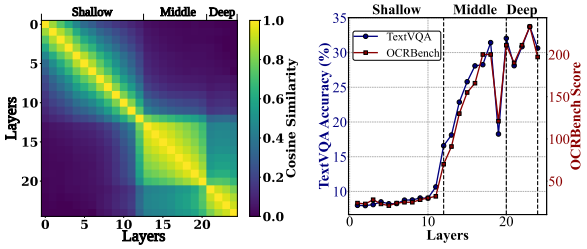
*Equal contribution.
†Corresponding Author. Code available at https://github.com/EIT-NLP/VisualProbing-for-MLLM

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 6689–6707
November 4-9, 2025

cally assess the efficacy of different deep vision layers. Our analysis reveals that *while the penultimate layer does not universally achieve peak performance in every scenario, it demonstrates consistent superiority across all evaluated model scales* (1.4B, 2.7B, and 7B parameters). This advantage stems from the penultimate layer's unique balance of preserving fine-grained visual details while maintaining strong alignment with textual representations. Notably, the performance gap between the penultimate layer and other deep layers widens as model scale increases. This suggests that *simply using larger LLMs cannot compensate for suboptimal visual feature selection*, underscoring the critical importance of visual layer choice in MLLMs.

Having established the penultimate layer's strength among deep layers, we ask a more fundamental question: *Can shallower ViT layers offer complementary or even superior information?* Our analysis shows that shallow and middle layers outperform deep layers in approximately one-third of sub-tasks in the MME benchmark (Fu et al., 2024) (Fig. 3), particularly in tasks involving fine-grained localization and counting. For instance, layer 18 outperforms the penultimate layer by 20% on position tasks (Fig.9). Similar trends are observed in MMVet (Yu et al., 2023). Although shallow layers generally show lower average performance, they still excel on a significant subset of tasks (Fig.2). In contrast, deeper layers remain crucial for tasks with high-level semantic demands such as OCR. To assess robustness, we evaluate across three training data scales (665k, 737k, and 1M samples). Despite some fluctuations, our findings consistently demonstrate that *shallow and middle layers carry underutilized yet valuable information*.

Motivated by these insights, we propose a simple yet effective fusion strategy that combines visual features from shallow, middle, and deep layers. Our method uses a single linear projection layer, keeping computational overhead minimal while achieving substantial performance gains. This minimalist approach offers a principled alternative to existing ad-hoc layer selection and fusion methods. Unlike prior works (Yao et al., 2024; Hong et al., 2024; Cao et al., 2024) that explore hierarchical feature fusion or LLM-aligned selection heuristically, our study provides the first systematic analysis of layer-wise information variation within ViTs, grounded in both intrinsic representation structure and downstream performance. Our key contributions are summarized as follows:



(a) (b)

Figure 1: (a) Average cosine similarity of visual representations across different layers in CLIP-ViT. (b) Layer-wise performance on OCR tasks. The results highlight three distinct representation regions and their influence on performance.

(1) We identify three semantically coherent groups of ViT layers (shallow, middle, deep) based on representation similarity. We show that shallow and middle layers, which are often overlooked, can outperform the commonly used deep layers (Sec. 4).

(2) Through extensive experiments across different data sizes and model scales, we confirm the generalization of our findings. Even as gains diminish with scaling, shallow and middle layers continue to exhibit unique strengths over deep layers in certain sub-tasks (Sec. 5).

(3) We design a linear-layer-based fusion method that integrates features from all three layer groups. It outperforms both specialized fusion designs (e.g., DenseConnector (Yao et al., 2024), MMFuser (Cao et al., 2024)) and standard practices in current MLLMs (e.g., using only the penultimate layer) (Sec. 6).

2 Related Work

Visual Encoder in Multimodal LLMs Serving as the "eyes" of MLLMs, the vision encoder sets the upper bound of the model's perceptual capabilities. CLIP, through image-text contrastive learning effectively aligns visual representation with text space and is widely adopted as the visual encoder in models such as LLaVA (Liu et al., 2023b,a), Qwen-VL (Wang et al., 2024), Flamingo (Alayrac et al., 2022), and BLIP (Li et al., 2023b). Other foundational vision models, such as DINOv2 (Oquab et al., 2023), SigLIP (Zhai et al., 2023), ConvNeXT (Liu et al., 2022), are also utilized to build MLLMs. In this paper, we select the widely used CLIP-ViT model as the focus of our layer-wise analysis.

Visual Layer Selection Recent studies have explored incorporating shallow visual features within

the ViT of multimodal language models, such as DenseConnector (Yao et al., 2024), MMFuser (Cao et al., 2024). Lin et al. (2025) have further investigated internal fusion strategies by integrating multiple visual layers with language representations, highlighting the critical role of visual layer selection in effective multimodal integration.

Previous methods have largely relied on intuitive, heuristic-based strategies, such as evenly sampling layers. Although some approaches have explored the distinct characteristics of different ViT layers (Gandelsman et al., 2024), the specific roles of layers at different depths in multimodal tasks remain unclear. This study conducts a comprehensive analysis of layer-wise visual representations in MLLMs, aiming to inform the selection of visual layers and guide the design of future visual fusion strategies.

3 Overall Setup

Problem Formulation MLLMs typically comprise three core components: a vision encoder, a connector that maps visual features to the language space, and a large language model (LLM) (Lin et al., 2024). This architecture empowers MLLMs to handle a diverse array of perception and reasoning tasks across both visual and textual modalities.

Most modern MLLMs adopt a pre-trained CLIP-ViT (Radford et al., 2021) as their image encoder. A ViT encodes an image into a sequence of token embeddings through a stack of transformer blocks. Each block (or *layer*) progressively refines the visual representations, with earlier layers focusing on low-level spatial details and later layers capturing more abstract, semantic information.

Formally, given an image I, the vision encoder produces a set of layer-wise outputs:

$$\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \ldots, \mathbf{H}^{(L)} \quad \text{where} \quad \mathbf{H}^{(l)} \in \mathbb{R}^{T \times d}$$

$\mathbf{H}^{(l)}$ denotes the embedding at the l-th layer, T is the number of tokens, and d is the dimension.

Despite the availability of rich multi-level features, most MLLMs select a single layer—often the penultimate or final one—to represent the entire image. This practice may overlook complementary signals from shallower layers that encode fine-grained visual details. In this work, we systematically investigate the impact of using different ViT layers for visual input and explore *how selecting appropriate layers can improve MLLM performance across diverse tasks*.

Partitioning of Visual Representations To examine the behavioral patterns of different visual layers, we analyze the relationships between them based on cosine similarities. Inspired by prior findings (Sun et al., 2024) that LLMs exhibit several distinct representation spaces through such analysis, we similarly identify three significantly different representation spaces within CLIP-ViT.

As shown in Fig.1a, three distinct representation spaces emerge among the visual layers (see Appendix A for computational details). Experiments on OCR and TextVQA (Fig. 1b) further indicate that shallow layers contribute little to performance, which rises substantially in the middle layers and peaks in the deep layers. Layers within the same representation space exhibit similar behaviors.

Based on behavioral similarity, we categorize the 24 CLIP-ViT visual layers into three groups: *shallow layers (1 to 12), middle layers (13 to 20), and deep layers (21 to 24)*.

Implementation Details We employ CLIP ViT-L/14 (336px) (Radford et al., 2021) as the visual encoder and 1.4B MobileLLaMA (Chu et al., 2024) as the language model for efficiency analysis, with a one-layer MLP serving as the connector. Training follows a two-phase strategy aligned with LLaVA (Liu et al., 2023b). AdamW optimizer with a cosine annealing scheduler is used, with learning rates of 1e-3 (phase one) and 2e-5 (phase two), and batch sizes of 256 and 128. Training on four NVIDIA A100 80GB GPUs takes 2 hours for phase one and 8 hours for phase two. We adopt the LLaVA 1.5 (Liu et al., 2023b) dataset, comprising 558K image-caption pairs which is carefully filtered from CC3M (Sharma et al., 2018) for pre-training and 665K conversational instances for instruction tuning. Unless explicitly noted, the experimental setup remains the same.

Evaluation Benchmarks To comprehensively explore and evaluate various visual representations, we classified the benchmarks into four categories following previous work (Tong et al., 2024a): General tasks, OCR tasks, Vision-centric tasks, and Hallucination tasks.

The **General tasks** category assesses basic vision-language reasoning abilities, including MME (Fu et al., 2024) (yes/no questions on attributes like existence and color), MMBench (Liu et al., 2024c) (multiple-choice across diverse aspects), SEEDBench (Li et al., 2023a) (spatial and temporal reasoning), and GQA (Hudson and Man-

Layers	General					OCR		Vision-Centric						Hallu
	MME^P	MME^C	MMB	SEEDB	GQA	TVQA	OCRB	CVB	CVB^{2D}	CVB^{3D}	RWD	MMVet	RefCOCO	POPE
1	750.1	211.4	0	25.30	40.55	7.99	24	40.14	34.87	45.42	38.17	9.9	5.73	70.21
2	790.2	212.5	0.34	25.76	41.24	7.96	23	40.21	34.34	46.08	37.12	10.1	5.36	71.01
3	742.7	219.2	0.17	25.00	41.76	8.10	28	42.69	37.89	47.50	36.08	10.4	6.87	72.33
4	788.4	239.6	0	25.53	42.26	8.50	23	42.69	36.71	48.67	37.25	10.2	7.61	72.34
5	813.2	220.7	0.17	25.26	42.74	8.22	21	41.30	33.69	48.92	36.86	10.9	8.25	72.91
6	838.8	227.8	0	25.23	43.16	8.26	24	41.69	35.97	47.42	36.86	11.5	9.31	75.18
7	815.6	235.7	0	25.74	44.90	8.75	25	43.02	37.12	48.92	37.39	10.6	11.10	75.44
8	857.7	237.5	0	25.48	46.14	8.77	25	41.43	36.85	46.00	36.99	11.2	10.79	76.20
9	889.7	232.8	0.17	27.72	47.02	9.05	28	40.53	36.23	44.83	37.12	13.0	10.06	77.84
10	903.4	228.2	0.17	26.61	48.39	9.03	30	41.8	36.19	47.42	37.39	11.4	13.50	77.46
11	935.3	224.3	0.52	26.58	49.85	10.65	32	42.51	37.27	47.75	36.86	14.2	12.14	79.24
12	980.1	232.1	0.09	26.85	50.39	16.58	70	41.81	36.7	46.92	38.56	12.6	11.20	80.63
13	964.0	252.5	0.09	26.33	51.14	18.12	91	41.71	35.75	47.67	37.25	11.6	12.50	81.39
14	984.2	**265.4**	0.69	34.07	51.83	22.86	130	42.69	36.12	49.25	39.08	13.8	14.37	81.97
15	1042.8	227.5	0.17	28.98	52.89	25.79	155	43.85	36.37	51.33	37.12	13.6	13.77	83.11
16	1069.5	225.4	0	27.95	52.81	28.08	166	43.26	36.61	49.92	38.04	13.7	15.41	84.32
17	1074.8	230.4	0.26	32.81	53.86	28.25	200	47.26	39.43	**55.08**	39.22	15.4	**18.49**	84.46
18	1088.7	237.1	29.38	52.06	54.37	31.44	200	**47.29**	**41.17**	53.42	39.48	14.3	17.04	84.26
19	945.1	236.8	20.02	44.64	48.32	18.27	121	45.69	37.21	54.17	35.95	13.2	18.22	81.47
20	1118.2	232.1	26.03	51.72	**54.83**	32.05	211	**47.29**	40.32	54.25	38.82	16.3	**18.49**	84.76
21	1041.4	212.5	0.95	35.42	49.47	28.10	190	44.37	39.32	49.42	39.87	14.5	17.09	81.91
22	1123.6	238.9	23.28	49.60	54.52	30.84	211	44.37	36.73	52.00	39.87	17.3	16.32	**84.79**
23	**1142.7**	245.0	**35.31**	**52.84**	54.61	**33.73**	**233**	44.26	38.02	50.50	**45.36**	**18.0**	17.08	84.00
24	1114.1	243.5	32.65	51.09	53.61	30.63	197	46.68	39.78	53.58	43.92	16.1	17.08	83.65

Table 1: Performance across layers 1–24. MME^P and MME^C represent the MME perception and cognition tasks respectively. SEEDB, GQA, OCRB and CVB refer to SEEDBench, General QA tasks, OCRBench and CVBench, with CVB^{2D} and CVB^{3D} indicating the 2D/3D subtasks of CVBench, respectively. RWD stands for RealWorldQA. This table provides a detailed analysis of all 24 layers, highlighting that *many optimal performances are found in the middle layers*, which are marked in bold.

ning, 2019a) (complex real-world VQA). The **OCR category** evaluates a model's ability to recognize textual content from images, featuring TextVQA (Singh et al., 2019) and OCRBench (Liu et al., 2024d). The **Vision-centric category** emphasize fine-grained perception and localization, including CVBench (Tong et al., 2024a) (evaluating spatial relations and depth), RealWorldQA (real-world QA), MMVet (Yu et al., 2023) (general multimodal assessment), and RefCOCO (Yu et al., 2016) (visual grounding). Finally, the **Hallucination category** includes POPE (Li et al., 2023c), which evaluates whether MLLMs generate false or invented content not grounded in the image.

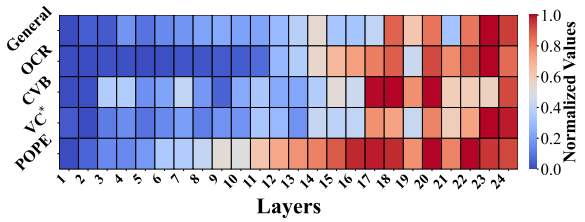


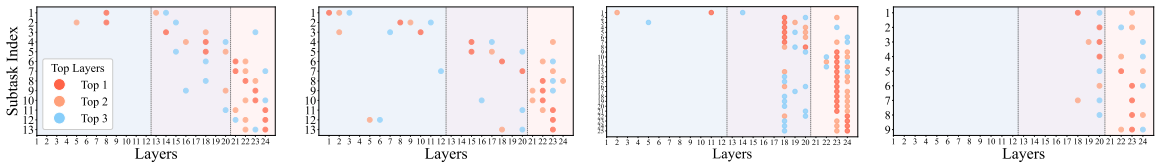
Figure 2: Averaged performance of layers 1 to 24 across various tasks. **General** represents tasks from MME, MMBench, GQA, and SEEDBench. **OCR** includes includes TextVQA and OCRBench. **CVB** corresponds to CVBench, whereas **VC*** includes RefCOCO, RealWorldQA, and MMVet. Results show that the final layer underperforms the penultimate layer, and middle layers sometimes surpass deeper ones.

4 Experiment: Layer-wise Exploration

Previous studies have primarily used techniques such as linear probing and attention head decomposition to analyze CLIP-ViT representations (Gandelsman et al., 2024). While these methods reveal what types of information are present in different ViT layers, they do not assess whether such information can be effectively utilized by MLLMs. The mere presence of information in a particular layer does not guarantee its usefulness when integrated into an MLLM. In contrast, our work goes beyond probing for representational content—we systematically evaluate how each ViT layer contributes to downstream MLLM performance. To this end, we conduct a layerwise exploration by individually connecting each visual layer to the language model, training the corresponding MLLM, and benchmarking its task performance. The layerwise performance is shown in Tab. 1 and Fig. 2.

4.1 Deep-to-Deep Layer Comparison

A common practice is to use deep layers from ViT as input to the MLLM. In this section, we investigate the effectiveness of this approach.



(a) MME (b) MMVet (c) MMBench (d) SEEDBench

Figure 3: Layer-wise performance distribution across four benchmarks: **(a) MME**, **(b) MMVet**, **(c) MMBench**, and **(d) SEEDBench**. The x-axis corresponds to layer indices and the y-axis indicates the sub-tasks index (see Tab. 12 for details). Top-performing layers for each sub-task are highlighted with color-coded markers: • (1st place), • (2nd place), and • (3rd place).

The final layer is not the optimal choice: As shown in Tab. 1 and Fig. 2, the final layer does *NOT* perform the best on any benchmark. For general tasks, a noticeable performance drop is observed at the final layer, with OCR tasks exhibiting particularly severe degradation. A similar trend is evident in POPE. However, vision-centric tasks partially show this decline. Overall, these results indicate that the final layer is *NOT* the optimal choice for representation across tasks.

The underlying reason might lie in the CLIP model's training mechanism, where supervision primarily focuses on aligning the final layer [CLS] token with text embeddings. The [CLS] token in the final layer is optimized by CLIP's contrastive loss, making it highly specialized for the image-text matching task. However, this optimization process can significantly suppress local image features in the final layer, which can harm performance when finer-grained details are required.

Penultimate layer as the optimal choice. As shown in Fig. 2, the penultimate layer consistently achieves the best performance across tasks. Notably, it outperforms other deep layers on General, OCR, and vision-centric tasks. This superiority stems from its ability to retain rich visual information while maintaining strong text alignment, which is second only to the final layer. Such a balance offers an optimal trade-off between visual expressiveness and semantic alignment, making it particularly well-suited for multimodal tasks.

4.2 Deep-to-Shallower Layer Comparison

Afterwards, we investigate the effects of using shallow and middle layers in MLLMs. We highlight the following key observations.

Deep layers are essential for OCR. As shown in Fig. 1, shallow layers provide negligible text information for the LLM. A clear boundary exists between the shallow and middle layers, with layer 12 marking the transition point. Layers before this point fail to contribute meaningfully to text processing, while a sharp performance gain occurs immediately afterward. This might be attributed to two essential requirements for OCR tasks:

1. *Rich fine-grained visual features in visual representation*: In OCR tasks, a strong perception of details is often required. Therefore, these fine-grained details must be embedded in the representation with sufficiently strong signals to be effectively utilized by the LLM.

2. *Well textually aligned visual features*: Despite containing rich visual details, shallow layers lack intrinsic alignment with textual representations, limiting their usefulness for OCR tasks. Fig. 1a confirms this with notably low cosine similarity between shallow and deep layers. This discrepancy poses a challenge for the connector, which can only align features originating from the deep (text) space or adjacent middle layers.

Limited impact of representation quality on cognitive tasks. Cognitive tasks, such as "Code Reasoning" and "Numerical Calculation", require both perception and high-level reasoning capabilities. Interestingly, we observe that even the shallow layers, which generally yield lower quality visual representations can rival or even outperform deeper layers in tasks under the MME-Cognitive. layer 3 achieves superior performance on "Code Reasoning", "Numerical Calculation', and "Text Recognition" compared to both middle and deep layers. (see Tab.7 and Tab.9 in Appendix) *These findings suggest that, for cognitive tasks, visual feature quality is not the primary limiting factor.*

Potential of middle layers We first conduct an investigation into the middle representation spaces. The performance of these two spaces, as shown in Tab. 1, several key insights emerge from the results:

(1) *The middle layer has the potential to perform best:* Although the middle layer's informa-

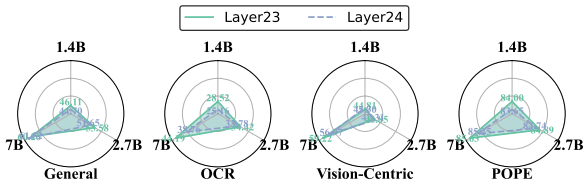


Figure 4: Radar charts comparing the performance of Layers 23 and 24 across four different tasks under three LLM scales: 1.4B, 2.7B, and 7B. The results consistently show that *the penultimate layer outperforms the final layer in all tasks*. This trend remains stable across different model scales.

tion has not been fully processed, it still achieves the best performance on one-third of the benchmarks. Specifically, compared to the penultimate layer, layer 14 achieves a 20-point higher score on MME-Cognitive, layer 18 outperforms by 3% on CVBench, layer 17 surpasses by 1.4% on RefCOCO, and layer 20 exceeds by 0.2% on GQA.

(2) *The middle layers generally perform better on vision-centric tasks:* Fig. 3a illustrates the performance of different layers across subtasks in the MME dataset, showing that position and existence tasks benefit more from middle layer representations. As depicted in Fig. 3b, the penultimate layer achieves top performance in only three out of eleven subtasks, whereas the shallow and middle layers yield optimal results in seven. Similarly, in Fig. 3c, one-third of the best performing results, such as those in spatial relations, physical relations, and cross fine-grained perception originate from the shallow and middle layers. A comparable trend is observed in SEEDBench (Fig. 3d), where middle layers produce optimal results in nearly half of the subtasks, including Instance Attribute, Instance Location, Instance Interaction, and Text Recognition.

The hallucination problem is more pronounced in shallow layers but is effectively mitigated in the middle layers. As shown in Tab. 1, POPE results indicate that hallucination issues are most prominent in the shallow representation space, with minimal variation between the middle and deep layers. Notably, in the middle representation space, half of the layers outperform the penultimate layer on the POPE. This phenomenon likely stems from the fact that the challenge of this task lies more in visual perception than in semantic comprehension. In Sec.5.3, we provide a detailed analysis showing that further experiments with larger LLMs consistently support this finding.

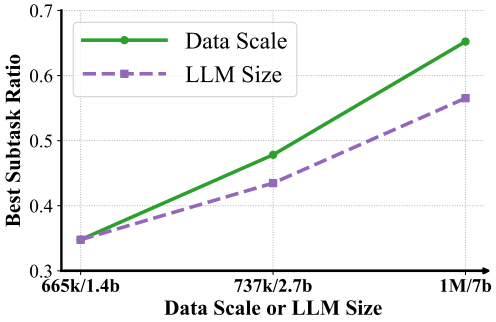


Figure 5: Proportion of subtasks achieving their best performance at the penultimate layer on MME and SEEDBench, demonstrating a clear upward trend.

5 Effect of Data and Model Scale

To further assess the generality of our findings, we extend our experiments to larger model scales and training datasets in this section and analyze the resulting performance trends.

5.1 Settings

More training Data. We investigate the impact of training data scales using the same training procedure detailed in Sec. 3. In the first stage, we use the LLaVA 558k dataset (Liu et al., 2023b). For the second stage, we evaluate three dataset configurations: **(1)** LLaVA 665K, **(2)** Cambrian-1 737K (Tong et al., 2024a), an expansion of the 665K dataset with additional OCR data, and **(3)** a custom 1M dataset that builds on the 737K dataset by incorporating data specifically curated for vision-centric tasks. The dataset composition can be found in Appendix B.

Scaling LLM sizes. Building on the original 1.4B experiments, we extend our study to include MobileLLaMA (Chu et al., 2024) 2.7B and Vicuna v1.5 7B. These LLMs are selected due to their similar architectures, making them well-suited for investigating the impact of different LLM sizes.

Due to computational constraints, we do not conduct a full layer-wise analysis across different data scales. Instead, leveraging insights from Sec 4.2, we select representative layers from the shallow (layer 3), middle (layer 18), and deep (layers 23 and 24) representation spaces to examine how variations in data scale affect model performance.

5.2 Deep-to-Deep Layer Comparison

As shown in Tab. 2, our findings indicate that the key conclusions in Sec. 4.2 remain valid across different training data scales. As data scales up, we uncover the following key insights:

Data Scale	Layers	General					OCR		Vision-Centric				Hallu
		MME^P	MME^C	MMB	SEEDB	GQA	TVQA	OCRB	CVB	CVB^{2D}	CVB^{3D}	RWQA	POPE
665k	**3**	742.7	219.3	0.17	25.00	41.76	8.10	28	42.69	37.89	47.50	36.08	72.33
	18	1088.8	237.1	29.38	52.06	54.37	31.44	200	**47.29**	**41.17**	53.42	39.48	**84.26**
	23	**1142.8**	**245.0**	**35.31**	**52.84**	**54.61**	**33.73**	**233**	44.26	38.02	50.50	**45.36**	84.00
	24	1114.1	243.6	32.65	51.09	53.61	30.63	197	46.68	39.78	**53.58**	43.92	83.65
737k	**3**	845.9	225.4	0.26	26.33	44.12	8.34	27	42.36	35.81	48.92	37.52	74.98
	18	1093.1	226.4	43.04	**56.33**	**56.98**	35.98	**270**	**48.87**	46.57	**51.17**	43.40	86.18
	23	**1163.7**	230.0	**48.37**	55.55	56.77	**36.41**	265	48.09	**47.59**	48.58	41.83	**86.22**
	24	1121.7	**258.6**	46.05	55.43	56.34	36.09	255	43.63	38.01	49.25	**44.44**	85.09
1M	**3**	871.8	215.4	13.40	40.21	45.67	8.03	26	43.04	37.74	48.33	39.08	74.08
	18	1145.9	213.2	42.44	56.72	57.74	35.68	**267**	**54.08**	**56.33**	51.83	43.14	84.06
	23	**1214.4**	**249.3**	**52.92**	**58.58**	**57.91**	**37.24**	263	53.48	53.96	**53.00**	43.27	84.03
	24	1192.0	245.0	47.34	57.62	57.21	36.45	264	47.88	42.84	52.92	**44.97**	**84.58**

Table 2: Performance comparison of visual representations across different data scales, demonstrating the consistency of our key findings. Even as gains diminish with scaling, middle layers continue to exhibit unique strengths over deep layers in OCRBench, SEEDBench, GQA and CVBench.

The penultimate layer remains the optimal choice in deep space regardless of LLM size As shown in Fig. 4, the penultimate layer consistently outperforms the final layer across LLMs of 1.4B, 2.7B, and 7B, reinforcing our findings. This indicates that CLIP-ViT's final-layer visual degradation, driven by its training paradigm, cannot be offset by a stronger LLM.

Furthermore, increasing LLM size does not yield significant improvements in POPE performance, indicating that hallucination bottlenecks in MLLMs stem primarily from the quality of the visual representation. In contrast, vision-centric tasks benefit more from scaling LLM size, indicating that even for tasks grounded in visual understanding, strong perception alone does not suffice, as robust reasoning capabilities remain essential.

The more data and the larger the model, the more the deep layers benefit. As both the training data scale and LLM size increase, a clear trend emerges in Fig. 5: the proportion of subtasks where the penultimate layer achieves the best performance consistently grows.

Since the visual encoder remains frozen during training, this suggests that compared to the middle layers, fine-grained information is less explicitly preserved in the deep layers. In other words, fine-grained details in the middle layers are more readily utilized by the LLM, whereas those in the deep layers are harder to extract, requiring larger amounts of data to activate the LLM to effectively capture these fine-grained features.

5.3 Deep-to-Shallower Layer Comparison

We observe that while conclusions from small models may not fully generalize to larger ones, the main findings still hold, as detailed below:

The potential of shallower layers persists across model and data scales The consistency of the previous conclusion is validated under larger training data and increased LLM size. As detailed in Appendix 9 and 10, under the 2.7B model, the penultimate layer fails to outperform shallower layers on several MME subtasks, such as Count, Position, and Existence. Similar patterns emerge in SEEDBench and persist with the 7B model, where shallower layers (e.g., Layer 18) achieve better results on tasks like Spatial Relation. On the 665K dataset, layer 18 outperforms the penultimate layer by up to 3% on CVBench and maintains a slight advantage on OCR and vision-centric tasks. This trend persists on the 737K dataset, where layer 18 continues to lead on SEEDBench and GQA. Although the performance gap narrows on the 1M dataset, the middle layer still surpasses the deep layer on both OCRBench and CVBench.

Limited gains on OCR tasks despite data and LLM scaling. Layer 3 exhibits comparable performance on OCR tasks both before and after incorporating OCR-specific training data, and across models of different scales (2.7B vs. 7B). This suggests that *increasing task-specific data alone cannot overcome the inherent limitations of shallow representations*. The lack of improvement can be attributed to their poor alignment with the textural feature space required for OCR understanding. In contrast, the middle layers, though only partially aligned, still exhibit performance gains under additional supervision.

6 Visual Feature Fusion

Building on the above-mentioned findings that deeper layers are not universally optimal and that

Models	General					OCR		Vision-Centric				Hallu	
	MME^P	MME^C	MMB	SEEDB	GQA	TextVQA	OCRB	CVB	CVB^{2D}	CVB^{3D}	RWQA	POPE	Win
Baseline(23)	1142.8	245.0	35.31	52.84	52.84	33.73	233	44.26	**38.02**	50.50	45.36	84.00	9/10
DenseConnector	1145.0	**253.2**	47.85	57.16	56.92	37.54	257	45.60	35.83	**54.92**	45.10	**84.95**	7/10
MMFuser	1149.5	238.9	**49.65**	56.21	56.59	35.43	245	**45.70**	36.89	54.50	44.83	84.53	8/10
Ours*	**1157.2**	236.1	49.22	**57.23**	**57.35**	**37.70**	**265**	44.56	36.53	52.58	**45.75**	84.82	-

Table 3: Study on different layer fusion strategies. "Ours" represents $\mathcal{L}_5 = 23, 18, 3$. "Win" denotes the proportion of datasets where our method achieves superior performance. Our method outperforms DC and MMFuser on 7 and 8 out of 10 benchmarks.

shallower layers offer valuable complementary information, we explore the most effective way to enhance visual representation by combining visual features from multiple layers. To be specific, we employ a simple fusion strategy to merge features from different layers and conduct a preliminary study on various layer combinations, aiming to highlight the potential benefits of layer fusion.

6.1 Method

The equation below shows the simplest visual feature fusion mechanism,

$$f = \text{Concat}\left(\mathcal{H}^{(i)} \mid i \in \mathcal{L}\right) \tag{1}$$

where $\mathcal{L}$ denotes the set of selected layers, and each $\mathcal{H}^{(i)}$ represents the feature representation extracted from layer l with a dimension of $N \times D$. Here, N is the number of visual tokens, and D is the feature dimensionality of each token. The concatenation function $\text{Concat}(\cdot)$ merges these representations along the feature dimension, producing an output f of size $N \times (D \times |\mathcal{L}|)$, where $|\mathcal{L}|$ denotes the number of concatenated layers. The resulting f is then fed into the Connector. Subsequently, we explore various layer combinations.

6.2 Exp-I: Ablation of Fusion Layer Selection

We select representative layers from shallow, middle, and deep layers for the fusion ablation study to preliminarily explore the effect of different representation spaces in fusion methods. We systematically construct different layer combinations $\mathcal{L}$ to analyze their impact on fusion performance.

Multiple stages bring generalization: Fig. 6 illustrates six different configurations, ranging from using only the end stage to incorporating all stages. Compared to two-layer fusion ($\mathcal{L}_1$) and three-layer fusion covering two stages ($\mathcal{L}_3$), three-layer fusion ($\mathcal{L}_{2,3}$) that spans all stages ($\mathcal{L}_2$ and Ours {23, 3, 18}) leads to more consistent performance improvements. Notably, $\mathcal{L}_2$ performs worse than Ours {23, 3, 18} on OCR tasks. This is likely because the first layer feature is too raw, making them less suitable for extracting the low-level visual cues required in OCR. By incorporating more stable representations from multiple stages, models can achieve the most robust performance across tasks.

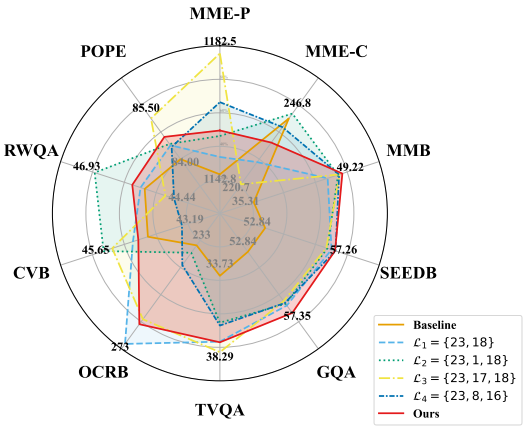


Figure 6: Performance comparison of different layer fusion combinations on four tasks. $\mathcal{L}_1$–$\mathcal{L}_4$ denote representative strategies for layer selection. "Ours" is $\mathcal{L}$ ={23, 3, 18}.

6.3 Exp-II: Fusion Method Comparison

To investigate the impact of different layer fusion strategies, we compare our selected layers {23, 3, 18} and a simple concatenation strategy, against other carefully designed fusion methods.

Less is more: We compare two state-of-the-art ad-hoc methods, DenseConnector (DC) and MMFuser with our method. As shown in Tab. 3, *our method outperforms DC and MMFuser on 7 and 8 across 10 widely used benchmarks*, respectively. These results suggest that complex fusion strategies may be unnecessary, as the simplest concatenation already meets the performance requirements.

7 Conclusion

This study presents a comprehensive layer-wise analysis, revealing that shallow and middle representation spaces can surpass the performance of deep layers. Evaluations across diverse data and model scales further substantiate our findings. Fur-

thermore, we introduce a straightforward yet highly effective fusion strategy for visual feature integration, delivering substantial improvements over the baseline. Our findings offer a foundation for advancing future research in fusion methodologies.

Limitations

Due to the high computational cost of layer-wise analysis, we adopt a linear probing-inspired strategy: most experiments are conducted on the 1.4B model, with selective validation on the 2.7B and 7B variants. However, our study does not extend to larger-scale LLMs. In terms of visual encoders, we focus exclusively on CLIP-ViT-L/14, given its widespread adoption, and leave the exploration of alternative backbones to future work.

Moreover, vision–language fusion strategies can be broadly classified into internal and external methods. Our analysis focuses only on external fusion approaches and does not directly compare them with internal alternatives. We ensure consistent experimental conditions across all settings, which enables a fair assessment of visual representation quality. However, our current design does not explore how different connector architectures may affect performance. We regard this as a valuable direction for future research.

Acknowledgement

We are grateful to EIT and IDT High Performance Computing Center for providing computational resources that greatly facilitated this research. This work was financially supported by the 2035 Key Research and Development Program of Ningbo City (Grant No. 2025Z034), the Pioneer and Leading Goose R&D Program of Zhejiang (Grant No. 2024C01110), and the National Natural Science Foundation of China (Grant No. 62472385).

References

Manoj Acharya, Kushal Kafle, and Christopher Kanan. 2019. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8076–8084.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Yue Cao, Yangzhou Liu, Zhe Chen, Guangchen Shi, Wenhai Wang, Danhuai Zhao, and Tong Lu. 2024. Mmfuser: Multimodal multi-layer feature fuser for fine-grained vision-language understanding. *arXiv preprint arXiv:2410.11829*.

Kaibing Chen, Dong Shen, Hanwen Zhong, Huasong Zhong, Kui Xia, Di Xu, Wei Yuan, Yifei Hu, Bin Wen, Tianke Zhang, Changyi Liu, Dewen Fan, Huihui Xiao, Jiahong Wu, Fan Yang, Size Li, and Di Zhang. 2024a. Evlm: An efficient vision-language model for visual understanding. *Preprint*, arXiv:2407.14177.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, and Chunhua Shen. 2024. Mobilevlm v2: Faster and stronger baseline for vision language model. *Preprint*, arXiv:2402.03766.

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.

Yossi Gandelsman, Alexei A. Efros, and Jacob Steinhardt. 2024. Interpreting CLIP's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and 1 others. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290.

Drew A. Hudson and Christopher D. Manning. 2019a. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Preprint*, arXiv:1902.09506.

Drew A Hudson and Christopher D Manning. 2019b. 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.

Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. 2023. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656.

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.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, and 1 others. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73.

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Xu Li, Yi Zheng, Haotian Chen, Xiaolei Chen, Yuxuan Liang, Chenghang Lai, Bin Li, and Xiangyang Xue. 2026. Instruction-guided fusion of multi-layer visual features in large vision-language models. *Pattern Recognition*, 170:111932.

Yanshu Li, Hongyang He, Yi Cao, Qisen Cheng, Xiang Fu, and Ruixiang Tang. 2025. M2iv: Towards efficient and fine-grained multimodal in-context learning in large vision-language models. *arXiv preprint arXiv:2504.04633*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023c. Evaluating object hallucination in large vision-language models. *Preprint*, arXiv:2305.10355.

Junyan Lin, Haoran Chen, Yue Fan, Yingqi Fan, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. 2025. Multi-layer visual feature fusion in multimodal llms: Methods, analysis, and best practices. *Preprint*, arXiv:2503.06063.

Junyan Lin, Haoran Chen, Dawei Zhu, and Xiaoyu Shen. 2024. To preserve or to compress: An in-depth study of connector selection in multimodal large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5666–5680, Miami, Florida, USA. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, and Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. 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, Kai Chen, and Dahua Lin. 2024c. Mmbench: Is your multi-modal model an all-around player? *Preprint*, arXiv:2307.06281.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. 2024d. Ocrbench: On the hidden mystery of ocr in large multimodal models. *Preprint*, arXiv:2305.07895.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. *Preprint*, arXiv:2201.03545.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, and 1 others. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer.

ShareGPT. 2023. `https://sharegpt.com/`.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. 2024. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. *arXiv preprint arXiv:2411.16044*.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. *Preprint*, arXiv:1904.08920.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2024. Transformer layers as painters. *arXiv preprint arXiv:2407.09298*.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Yang, and 1 others. 2024a. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024b. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Huanjin Yao, Wenhao Wu, Taojiannan Yang, YuXin Song, Mengxi Zhang, Haocheng Feng, Yifan Sun, Zhiheng Li, Wanli Ouyang, and Jingdong Wang. 2024. Dense connector for mllms. *arXiv preprint arXiv:2405.13800*.

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

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *Preprint*, arXiv:2308.02490.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.

Appendix

We provide some additional information as supplementary material. This material is divided into four sections:

- A detailed analysis and formal computation process of visual representations is provided in Appendix A.
- Experimental details covering dataset composition, evaluation protocols, visualization of fusion performance, and analyses on model factors such as LLM choice, data scale, layer selection, and feature fusion are provided in Appendix B.
- We provide additional details of our reasoning and rebuttal explanations in Appendix C.
- Ethics statement is provided in Appendix D.

A Visual Representations

A.1 Formal Computation of LRS

To examine the behavioral patterns of different visual layers, we introduce **Layer-wise Representation Similarity (LRS)** to analyze the relationships between various visual layers.

Specifically, we compute the cosine similarity between hidden states of ViT layers, where values closer to 1 indicate higher similarity. The formalized division process is detailed in the following equation:

We define the hidden state matrix H of CLIP-ViT as:

$$H \in \mathbb{R}^{L \times d}$$

where L is the number of layers, and each row $H_i \in \mathbb{R}^d$ represents the hidden state of the i-th layer. The cosine similarity matrix S is computed as:

$$S = \left| \frac{HH^T}{\|H\|_2 \|H^T\|_2} \right|$$

where HH^T computes the pairwise dot products between layer hidden states. The denominator normalizes the values using the L_2-norm, defined as:

$$\|H_i\|_2 = \sqrt{\sum_{k=1}^{d} (H_i^{(k)})^2}$$

ensuring that values lie within $[-1, 1]$. The absolute value guarantees that all elements are within the range $[0, 1]$. Then, we compute the average of S across four tasks.

We also explore the variations in visual representation spaces across four tasks. As illustrated in Figure 7, the partitioning of visual representations is minimally influenced by the nature of the tasks. In other words, the shallow, middle, and deep representation spaces exhibit remarkable stability, maintaining consistent structures across various tasks.

A.2 Replacing the Penultimate Layer

As shown in Table 4, when the penultimate-layer visual representations are replaced by those from other layers of the visual encoder without additional training, layers 20 to 24 (belonging to the deep representation space) do not suffer catastrophic performance degradation. Moreover, among the two layers closest to the penultimate one, layer 22 exhibits more stable performance than the final layer.

Layer	MME-P	MME-C	OCRB	TextVQA	RefCOCO
24	1153.5	306.0	266	41.16	47.56
23	1509.9	365.3	314	46.10	49.04
22	1451.1	366.7	304	44.76	47.46
21	1368.8	293.2	287	41.59	40.47
20	1259.2	265.7	271	39.11	44.31
19	1183.3	267.1	240	36.76	42.83
18	1083.7	237.8	205	32.18	36.04
17	993.6	255.7	156	27.92	31.07
16	901.0	256.7	116	23.37	19.96
15	790.0	253.9	94	17.96	14.85

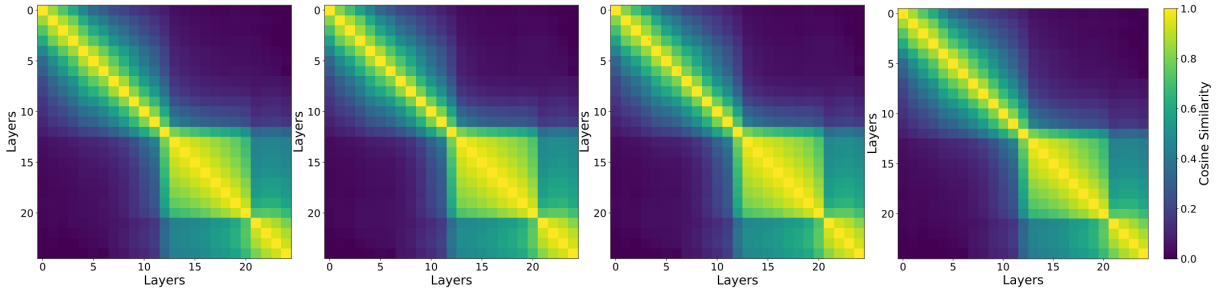
Table 4: Performance metrics across different layers on various benchmarks for non-training methods are presented. Specifically, MME-P denotes MME Perception, MME-C corresponds to MME Cognition, and OCRB represents OCR-Bench. The performance on RefCOCO is evaluated using Intersection over Union (IOU) as the metric.

B Experiment Details

B.1 Composition of Three Scale Datasets

As shown in Table 5, the following datasets are incorporated to enhance the model's capabilities across multiple multimodal tasks, with all datasets accessed and utilized strictly under their licenses.

- **AI2D (Allen Institute for AI Diagram Dataset) (Kembhavi et al., 2016)** AI2D is designed for visual reasoning and diagram understanding, featuring annotated diagrams with textual descriptions and Q&A pairs. It is



(a) General (b) OCR (c) CV-Centric (d) Hallucination

Figure 7: A visualization of the average cosine similarity of visual representations across different layers in CLIP-ViT for four tasks, namely General, OCR, CV-Centric, and Hallucination. Values closer to 1 indicate greater similarity.

Data	Size
LLaVA (Liu et al., 2024b)	158K
+ ShareGPT (ShareGPT, 2023)	40K
+ VQAv2 (Goyal et al., 2017)	83K
+ GQA (Hudson and Manning, 2019b)	72K
+ OKVQA (Marino et al., 2019)	9K
+ OCRVQA (Mishra et al., 2019)	80K
+ A-OKVQA (Schwenk et al., 2022)	66K
+ TextCaps (Sidorov et al., 2020)	22K
+ RefCOCO (Kazemzadeh et al., 2014; Mao et al., 2016)	48K
+ VG (Krishna et al., 2017)	86K
LLaVA-1.5 (Liu et al., 2023a)	665K
+ AI2D (Kembhavi et al., 2016)	16K
+ DocVQA (Mathew et al., 2021)	15K
+ DVQA (Kafle et al., 2018)	13K
Cambrian-737k (Tong et al., 2024a)	737K
+ CLEVR (Johnson et al., 2017)	215k
+ TallyQA (Acharya et al., 2019)	77K
Customized-1M	1M

Table 5: The mixture detail of fine-tuning dataset for LLaVA-1.5 665K, Cambrian-1 737K and customized 1M.

particularly useful for multimodal reasoning and visual question answering (VQA) tasks.

- **DocVQA (Document Visual Question Answering) (Mathew et al., 2021)** DocVQA focuses on visual question answering over document images, where questions pertain to scanned documents, OCR-recognized text, and textual reasoning. This dataset is valuable for document comprehension, text recognition, and multimodal reasoning.

- **DVQA (Diagrammatic Visual Question Answering) (Kafle et al., 2018)** DVQA is designed for visual question answering over diagrams and charts, covering questions related to bar charts, pie charts, and scientific illustrations. It evaluates the model's ability to read structured visual information and perform reasoning based on graphical representations.

- **CLEVR (Compositional Language and Elementary Visual Reasoning) (Johnson et al., 2017)** CLEVR is a synthetic dataset for visual reasoning, containing 3D-rendered scenes with structured questions that require reasoning based on attributes, object relationships, and compositional logic. It is widely used to assess a model's capability in compositional and multi-step reasoning.

- **TallyQA (Acharya et al., 2019)** TallyQA is a dataset specifically designed for object counting tasks, where questions require the model to accurately count objects in an image. It evaluates the model's ability to attend to relevant objects, integrate global and local information, and perform numerical reasoning.

B.2 Evaluation Metrics

We provide a comprehensive explanation of the evaluation methods, categorizing them into three distinct types based on the evaluation metrics:

- For benchmarks such as MME-Perception, MME-Cognition, OCRBench, and MMVet, we adopt the common approach of directly using the dataset-defined scores. We follow this established approach to maintain consistency and comparability in evaluations.

- Using Accuracy directly as the evaluation metric. This applies to benchmarks such as MMBench, SEEDBench, GQA, TextVQA, CVBench, RealworldQA, and POPE.

- In evaluating the RefCOCO dataset, we use CIDEr (Consensus-based Image Description Evaluation) as the primary evaluation metric.

To facilitate evaluation, we use lmms-eval as our primary evaluation tool. Evaluations must be conducted on the official platform for the MMVet dataset by uploading the necessary data. Regarding the CVBench 3D tasks, where models generally exhibit weaker instruction-following performance, we employ the DeepSeek API as the judge. This tool provides results consistent with GPT-4o but is significantly more cost-effective.

B.3 The impact of LLMs

Additional experiments are conducted on different sizes of large language models to investigate their impact on visual information processing. We validate our conclusions on LLMs of 2.7b and 7b sizes. Due to computational resource constraints, we selected representative layers from the three representation spaces to conduct experiments on subtasks of MME and SEEDBench. As shown in Table 6, the penultimate layer does not consistently achieve the best performance on MME. The commonly used penultimate layer achieves optimal performance on 6 out of 14 subtasks, while other layers, such as Layers 3, 18, and 24, demonstrate superior performance on the remaining subtasks. This observation aligns with prior findings, suggesting that middle layers can exhibit superior performance over deeper layers on certain CV-centric tasks. Notably, layer 18 outperforms the penultimate layer in tasks such as Count, Position, and Existence.

As illustrated in Table 7, while performance varies slightly across the subtasks of SEEDBench, the penultimate layer achieves the best performance on only 3 out of 9 subtasks. These results provide strong empirical evidence that shallow and middle layers can outperform deeper layers on specific subtasks.

B.4 The impact of Data Scale

The conclusion remains valid across different data scales. Under the 737k data scale, half of the subtasks in the MME dataset achieve optimal performance using the penultimate layer. However, for tasks like Count, Position, and Existence, the middle visual representation layer (Layer 18) demonstrates either superior or comparable performance. Similarly, results under the 1M data scale also show that half of the optimal performances are achieved on layers other than the penultimate one.

The results for SEEDBench subtasks, as presented in Table 10, further support this observation. At the 737k data scale, Layer 18 from the middle representation space achieves the best performance on 5 out of 9 subtasks, while the penultimate layer excels in only 3 subtasks. Likewise, under the 1M data scale, half of the subtasks continue to achieve their best performance on layers other than the penultimate one. These findings consistently demonstrate across varying data scales that shallow and middle layers have the potential to outperform deep layers in certain scenarios.

B.5 Layer Selections and Feature Fusion

In the shallow layer, we select layers 1 and 3 as representatives. Layer 1, being the most chaotic, primarily captures early-stage visual features, while layer 3 is still in a chaotic state but performs relatively well. In the middle layer, we choose layers 18 and 17, as they achieve the first and second-best performance within this representation space. For the deep layer, we select layer 23, as it demonstrates the highest overall performance. The baseline configuration considers only layer 23 as the visual representation.

Full version of comparison study as shown in Figure 8. We evaluate four state-of-the-art fusion methods, including three from DenseConnector (DC) (Yao et al., 2024) and one from MMFuser (Cao et al., 2024). As shown in Table 3, our method outperforms STI, SCI, and DCI on 8, 7, and 7 benchmarks, respectively. Compared to MMFuser (Cao et al., 2024), our approach demonstrates supe-

Model Size	2.7b				7b			
Layers	3	18	23	24	3	18	23	24
Code Reasoning	**52.50**	47.50	47.50	40.00	**50.00**	40.00	42.50	45.00
Artwork	53.00	65.00	**65.75**	64.50	50.00	69.25	**71.00**	70.75
Celebrity	46.76	49.12	**64.12**	58.82	51.76	59.41	**74.71**	74.41
Numerical Calculation	**50.00**	**50.00**	42.50	25.00	**47.50**	45.00	37.50	37.50
Text Translation	50.00	50.00	50.00	50.00	65.00	65.00	47.50	**67.50**
Count	50.00	**65.00**	61.67	58.33	56.67	**85.00**	**85.00**	80.00
Color	53.33	83.33	86.67	**88.33**	78.33	**91.67**	**91.67**	**91.67**
Commonsense Reasoning	52.86	60.71	**64.29**	62.14	57.86	69.29	**73.57**	72.86
Position	48.33	**71.67**	**71.67**	**71.67**	61.67	71.67	**75.00**	80.00
OCR	50.00	67.50	**72.50**	65.00	55.00	**77.50**	75.00	70.00
Landmark	59.50	75.25	**80.25**	76.75	66.25	78.00	**86.00**	84.50
Scene	73.75	87.75	87.75	**89.00**	80.00	85.50	85.50	**87.50**
Existence	83.33	**98.33**	96.67	95.00	81.67	**96.67**	**96.67**	95.00
Posters	37.41	59.18	**65.65**	64.29	51.02	74.83	81.63	**83.33**

Table 6: Performance of LLaVA architectures with 2.7B and 7B LLMs on MME subtasks, evaluated across four layers from three representative spaces.

Model Size	2.7b				7b			
Layers	3	18	23	24	3	18	23	24
Scene Understanding	36.04	68.84	68.87	**69.79**	50.70	73.50	**73.91**	73.59
Instance Identity	32.33	62.26	**62.92**	62.59	41.40	67.78	70.29	**70.56**
Instance Attribute	40.35	**62.19**	59.07	60.46	50.48	69.09	**68.70**	68.21
Instance Location	37.53	52.25	49.69	**53.78**	43.46	61.04	**61.45**	59.71
Instance Counting	25.70	43.07	**47.57**	45.20	33.02	56.89	57.13	**57.29**
Spatial Relation	33.03	42.47	40.64	**43.99**	41.55	**52.05**	49.62	51.45
Instance Interaction	34.02	**64.95**	54.64	64.95	48.45	63.92	67.01	**71.13**
Visual Reasoning	35.05	67.07	**72.51**	72.21	53.47	75.23	**78.85**	77.04
Text Recognition	**44.71**	21.18	21.18	24.71	38.82	34.12	**47.06**	43.53

Table 7: Performance of LLaVA architectures with 2.7B and 7B LLMs on SEEDBench subtasks, evaluated across four layers from three representative spaces.

rior performance on 8 benchmarks. These results highlight the significant potential of visual feature fusion strategies in enhancing MLLMs and offering guidance for developing future fusion strategies.

C Detailed Discussion

C.1 Alignment Guideline

What defines a good visual representation in multimodal models? Firstly, an ideal visual representation must simultaneously provide rich visual information and effectively align with textual modal. An MLLM can only correctly answer queries when the information corresponding to the given instruction is explicitly embedded within the visual representation. However, when the visual representation fails to deliver the necessary information, the model becomes prone to hallucination problems. Secondly, alignment with the textual modality is essential for a large language model to understand and process information from a different modality. This alignment ensures that the rich visual content is effectively leveraged. In a word, the CLIP series models currently offer the best trade-off between these two dimensions.

C.2 Comparison with Prior Work

Although (Liu et al., 2023b; Yao et al., 2024; Cao et al., 2024; Li et al., 2026) have briefly considered features from different layers, none have conducted a systematic or in-depth analysis of layer selection or the functional roles of visual layers in MLLMs.

- (Liu et al., 2023b) presents a limited comparison between the penultimate and final layers on ScienceQA, without broader analysis. In contrast, we generalize this observation across over 10 benchmarks and various model and

Models	General					OCR		Vision-Centric				Hallu	
	MME^P	MME^C	MMB	SEEDB	GQA	TextVQA	OCRB	CVB	CVB^{2D}	CVB^{3D}	RWQA	POPE	**Win**
Baseline(23)	1142.8	245.0	35.31	52.84	52.84	33.73	233	44.26	38.02	50.50	45.36	84.00	9/10
+ *18*	1148.5$^{5.7\uparrow}$	228.9$^{16.1\downarrow}$	46.91$^{11.6\uparrow}$	57.01$^{4.2\uparrow}$	56.80$^{4\uparrow}$	37.66$^{3.9\uparrow}$	273$^{40\uparrow}$	44.73$^{0.5\uparrow}$	35.79$^{2.2\downarrow}$	53.67$^{3.2\uparrow}$	45.49$^{0.1\uparrow}$	84.51$^{0.5\uparrow}$	8/10
+ *1+18*	1155.4$^{12.6\uparrow}$	246.8$^{1.8\uparrow}$	48.54$^{13.2\uparrow}$	56.75$^{3.9\uparrow}$	56.68$^{3.8\uparrow}$	36.53$^{2.8\uparrow}$	236$^{3\uparrow}$	45.65$^{1.4\uparrow}$	36.21$^{1.8\downarrow}$	55.08$^{4.6\uparrow}$	46.93$^{1.6\uparrow}$	84.56$^{0.6\uparrow}$	7/10
+ *17+18*	1182.5$^{39.7\uparrow}$	220.7$^{24.3\downarrow}$	48.80$^{13.5\uparrow}$	56.68$^{3.8\uparrow}$	56.48$^{3.6\uparrow}$	38.29$^{4.6\uparrow}$	263$^{30\uparrow}$	45.38$^{1.1\uparrow}$	36.25$^{1.8\downarrow}$	54.50$^{4\uparrow}$	44.71$^{0.7\downarrow}$	85.50$^{1.5\uparrow}$	6/10
DC-STI	1142.4$^{0.4\downarrow}$	218.9$^{26.1\downarrow}$	48.02$^{12.7\uparrow}$	57.23$^{4.4\uparrow}$	56.86$^{4.0\uparrow}$	36.42$^{2.7\uparrow}$	226$^{7\downarrow}$	43.83$^{0.4\downarrow}$	35.10$^{2.9\downarrow}$	52.58$^{2.1\uparrow}$	44.44$^{0.9\downarrow}$	86.38$^{2.4\uparrow}$	8/10
DC-SCI*	1166.5$^{23.7\uparrow}$	241.8$^{3.2\downarrow}$	48.71$^{13.4\uparrow}$	57.26$^{4.4\uparrow}$	56.61$^{3.8\uparrow}$	36.70$^{3.0\uparrow}$	241$^{8\uparrow}$	43.19$^{1.1\downarrow}$	34.96$^{3.1\downarrow}$	51.42$^{0.9\uparrow}$	44.44$^{0.9\downarrow}$	84.45$^{0.5\uparrow}$	7/10
DC-DCI	1145.0$^{2.2\uparrow}$	253.2$^{8.2\uparrow}$	47.85$^{12.5\uparrow}$	57.16$^{4.3\uparrow}$	56.92$^{4.1\uparrow}$	37.54$^{3.8\uparrow}$	257$^{24\uparrow}$	45.60$^{1.3\uparrow}$	35.83$^{2.2\downarrow}$	54.92$^{4.4\uparrow}$	45.10$^{0.3\downarrow}$	84.95$^{1.0\uparrow}$	7/10
MMFuser	1149.5$^{6.7\uparrow}$	238.9$^{6.1\downarrow}$	49.65$^{14.3\uparrow}$	56.21$^{3.4\uparrow}$	56.59$^{3.8\uparrow}$	35.43$^{1.7\uparrow}$	245$^{12\uparrow}$	45.70$^{1.4\uparrow}$	36.89$^{1.1\downarrow}$	54.50$^{4.0\uparrow}$	44.83$^{0.5\downarrow}$	84.53$^{0.5\uparrow}$	8/10
Ours*	1157.2$^{14.4\uparrow}$	236.1$^{8.9\downarrow}$	49.22$^{13.9\uparrow}$	57.23$^{4.4\uparrow}$	57.35$^{4.5\uparrow}$	37.70$^{4.0\uparrow}$	265$^{32\uparrow}$	44.56$^{0.3\uparrow}$	36.53$^{1.5\downarrow}$	52.58$^{2.1\uparrow}$	45.75$^{0.4\uparrow}$	84.82$^{0.8\uparrow}$	-

Table 8: Study on different layer fusion strategies. The results reveal that nearly all fusion methods significantly outperform the baseline, with performance variations depending on the combination of different layers. (*) 'DC-SCI' is the same as $\mathcal{L}_4$ and 'Ours' represents $\mathcal{L}_5$. 'Win' denotes the proportion of datasets where our method achieves superior performance.

Data Scale	**737k**				**1M**			
Layers	**3**	**18**	**23**	**24**	**3**	**18**	**23**	**24**
Code Reasoning	**50.00**	47.50	45.00	42.50	**47.50**	**47.50**	45.00	**47.50**
Artwork	51.00	59.25	**64.50**	61.75	53.75	65.00	**68.00**	66.50
Celebrity	48.82	55.88	**64.12**	62.65	52.35	62.06	**68.53**	65.59
Numerical Calculation	**47.50**	37.50	35.00	**47.50**	**50.00**	30.00	45.00	47.50
Text Translation	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Count	**55.00**	**55.00**	**55.00**	50.00	58.33	58.33	**60.00**	**60.00**
Color	60.00	75.00	**76.67**	75.00	61.67	78.33	**80.00**	**80.00**
Commonsense Reasoning	54.29	56.43	60.71	**61.43**	52.14	57.14	**62.14**	58.57
Position	51.67	**70.00**	63.33	**70.00**	50.00	70.00	**73.33**	70.00
OCR	50.00	55.00	**57.50**	52.50	**55.00**	52.50	**55.00**	**55.00**
Landmark	63.25	71.00	**77.50**	74.25	61.75	72.50	**77.50**	76.00
Scene	69.25	83.50	**85.00**	84.25	74.00	84.50	84.00	**85.75**
Existence	85.00	**96.67**	**96.67**	**96.67**	80.00	**96.67**	**96.67**	95.00
Posters	38.44	50.34	55.44	**56.80**	37.41	52.72	**55.44**	54.76

Table 9: Experimental results on MME subtasks under data scales of 737k and 1M, with the model settings consistent with those described in the main text.

data scales, and offer an interpretation in Section 4.1 grounded in CLIP's contrastive learning objective.

- (Yao et al., 2024) uses attention maps to suggest that different layers attend to different regions of interest. However, such analysis is inherently qualitative and subjective, lacking a principled explanation of why these differences arise or how they relate to downstream tasks.
- (Cao et al., 2024) proposes a shallow-to-deep fusion strategy but does not explain the underlying rationale for selecting specific layers, and its choices are based on empirical heuristics (e.g., 1/3, 2/3, 3/3), with no analysis of representations itself.
- While (Li et al., 2026) focuses on fusion design, it lacks consideration of the role of visual layers in MLLMs. Our work provides novel insights, such as the unexpected correlation between inter-layer similarity and downstream performance. Moreover, they analyze only 1/5 of ViT layers, without considering data scale or model size.

In contrast, our work is the first to systematically group ViT layers based on representation similarity, and to establish clear correlations between layer types and multimodal task performance across 60+ tasks and multiple model sizes (1.4B–7B) and data scales (665k-1M).

D Ethics Statement

This work made limited use of ChatGPT for language polishing and code debugging support. All scientific contributions, analyses, and conclusions are those of the authors.

Data Scale	737k				1M			
Layers	3	18	23	24	3	18	23	24
Scene Understanding	26.85	67.35	**68.24**	67.57	44.02	65.71	**68.97**	67.57
Instance Identity	23.65	**56.36**	55.60	55.87	35.06	57.89	**61.82**	58.77
Instance Attribute	26.03	**59.93**	58.36	57.69	44.27	**61.93**	61.45	60.59
Instance Location	26.69	**50.20**	47.75	48.26	39.37	48.67	**49.69**	49.08
Instance Counting	27.34	**41.93**	40.13	41.19	32.04	42.38	44.91	**45.97**
Spatial Relation	29.68	**41.86**	38.81	38.96	37.90	**42.92**	39.57	39.57
Instance Interaction	21.65	46.39	**51.55**	49.48	38.14	50.52	51.55	**52.58**
Visual Reasoning	27.79	62.54	67.98	**69.18**	43.20	60.42	**68.88**	67.07
Text Recognition	21.18	34.12	**48.24**	43.53	41.18	17.65	**55.29**	42.35

Table 10: Experimental results on SEEDBench subtasks under data scales of 737k and 1M, with the model settings consistent with those described in the main text.

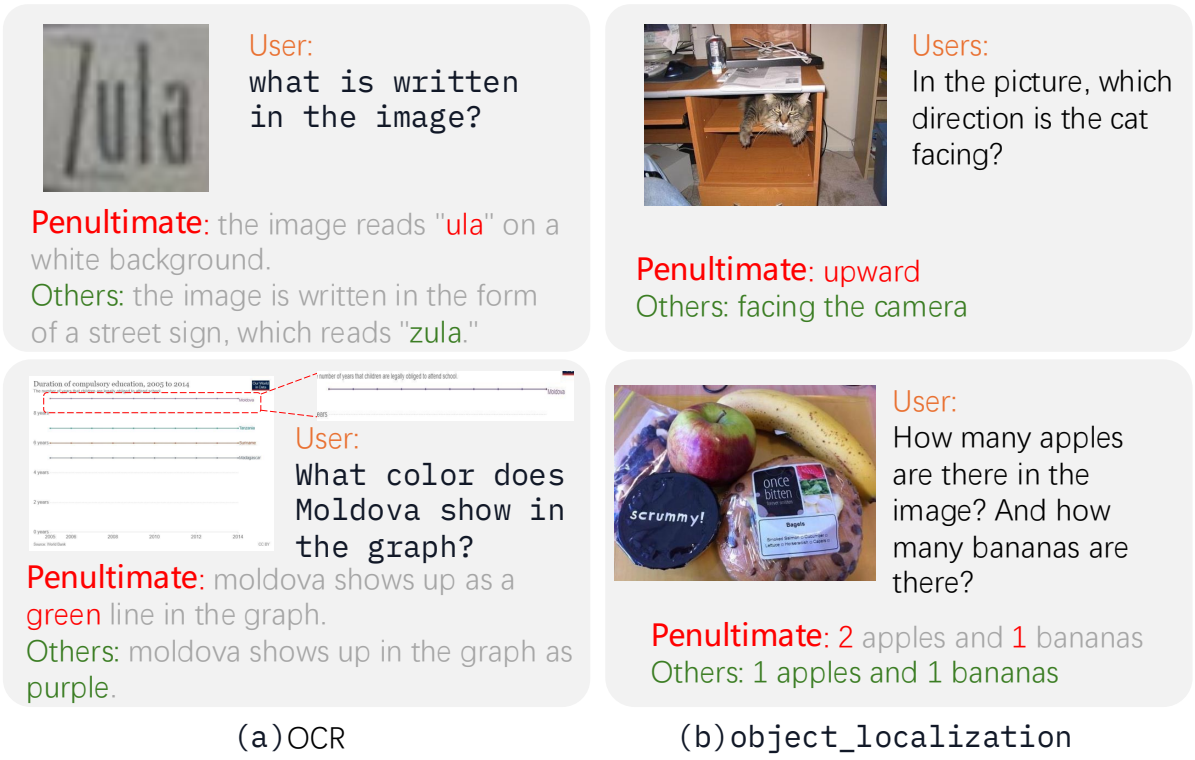


Figure 8: Case study illustrating four examples where the penultimate layer provides incorrect answers, but these errors can be resolved by using shallow and middle layers. In all four cases, Layer 18 of CLIP-ViT as the visual representation could successfully provide the correct answers.

Model	MMB	SEEDB	GQA	OCRB	RefCOCO
Baseline (Vicuna-7B)	65.1	65.2	63.2	320	52.3
DenseConnector (Yao et al., 2024)	63.3	66.6	62.9	323	-
MMFuser (Cao et al., 2024)	67.5	60.8	62.8	-	-
Instruction-Guided Fusion (Li et al., 2026)	66.9	68.3	63.1	-	-
Ours	67.4	67.7	63.5	325	54.0

Table 11: Performance comparison on 7B-scale LLMs. Dense2Connector and MMFuser are retrained under the same settings, while results of Instruction-Guided Fusion are copied from (Li et al., 2026). The results highlight the importance of visual layer selection.

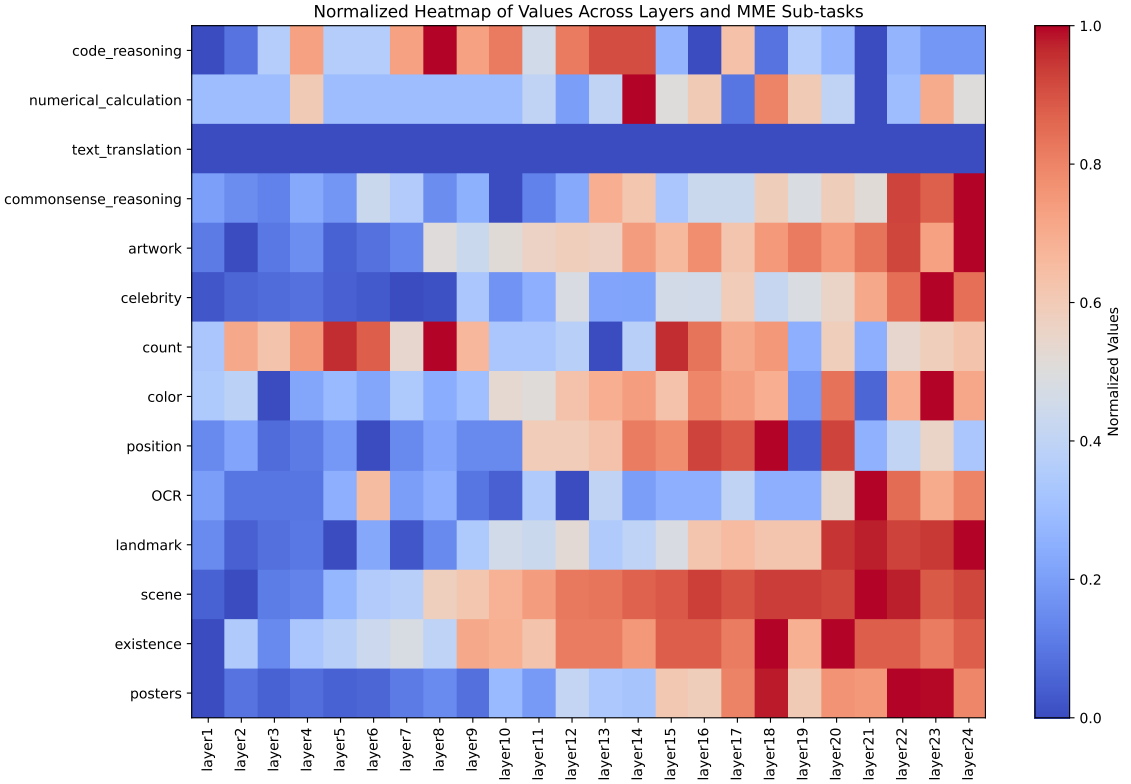


Figure 9: Heatmap showing the relative performance of CLIP-ViT layers 1-24 on MME subtasks. Consistent with the experimental settings in the main text, the results demonstrate that while deep layers generally achieve the best performance, shallow and middle layers can surpass deep layers on specific tasks, providing further support for our findings.

Index	MME	MMVet	MMBench	SEEDBench
1	Code Reasoning	Rec_OCR_Spat_Math	Attribute Comparison	Instance Attribute
2	Count	Rec_Spat	Attribute Recognition	Instance Location
3	Numerical Calculation	Rec_OCR_Know_Gen	Physical Property	Instance Interaction
4	Position	OCR_Spat	Spatial Relation	Text Recognition
5	Existence	OCR_Know_Spat	Image-Text Understanding	Instance Identity
6	Scene	Rec	Fine-grained Perception (Cross-instance)	Scene Understanding
7	OCR	OCR_Gen_Spat	Image Style	Instance Counting
8	Poster	Rec_Know_Gen	Physical Relation	Spatial Relation
9	Color	Rec_Know	Image Scene	Visual Reasoning
10	Celebrity	Rec_OCR_Gen_Spat	Celebrity Recognition	—
11	Landmark	OCR	Identity Reasoning	—
12	Artwork	Rec_OCR_Gen	Image Emotion	—
13	Commonsense Reasoning	Rec_OCR	Image Topic	—
14	—	—	Natural Relation	—
15	—	—	Object Localization	—
16	—	—	OCR	—
17	—	—	Social Relation	—
18	—	—	Attribute Recognition	—
19	—	—	Coarse Perception	—
20	—	—	Fine-grained Perception (Single-instance)	—
21	—	—	Relation Reasoning	—
22	—	—	Action Recognition	—
23	—	—	Function Reasoning	—
24	—	—	Future Prediction	—
25	—	—	Logical Reasoning	—

Table 12: Subtask-to-Index Mapping Table. Each column corresponds to the substask taxonomy defined in MME (Fu et al., 2024), MMVet (Yu et al., 2023), MMBench (Liu et al., 2024c) and SEEDBench (Li et al., 2023a).