From One-to-One to Many-to-Many: Dynamic Cross-Layer Injection for Deep Vision-Language Fusion

Cheng Chen[1,2*], Yuyu Guo[2*], Pengpeng Zeng[1], Jingkuan Song[1,3†], Peng Di[2], Hang Yu[2†], and Lianli Gao[4]

[1] Tongji University, China
[2] Ant Group, China
[3] Shanghai Innovation Institute, China
[4] Independent Researcher, China
{cczacks,yuyuguo1994,jingkuan.song}@gmail.com, hyu1@e.ntu.edu.sg

Abstract. Vision-Language Models (VLMs) create a severe visual feature bottleneck by using a crude, asymmetric connection that links only the output of the vision encoder to the input of the large language model (LLM). This static architecture fundamentally limits the ability of LLMs to achieve comprehensive alignment with hierarchical visual knowledge, compromising their capacity to accurately integrate local details with global semantics into coherent reasoning. [5] To resolve this, we introduce **Cross-Layer Injection (CLI)**, a novel framework that forges a dynamic "**many-to-many**" bridge between the two modalities. CLI consists of two synergistic, parameter-efficient components: an **Adaptive Multi-Projection (AMP)** module that harmonizes features from diverse vision layers, and an **Adaptive Gating Fusion (AGF)** mechanism that empowers the LLM to selectively inject the most relevant visual information based on its real-time decoding context. We validate the effectiveness and versatility of CLI by integrating it into LLaVA-OneVision and LLaVA-1.5. Extensive experiments on 28 diverse benchmarks demonstrate significant performance improvements, establishing CLI as a scalable paradigm that unlocks deeper multimodal understanding by granting LLMs on-demand access to the full visual hierarchy. Code is available at `https://github.com/codefuse-ai/CLI`.

Keywords: Vision-Language Model · Multimodal Large Language Model

1 Introduction

Vision-Language Models (VLMs) [1, 18, 22, 23] are increasingly viewed as a pivotal step toward Artificial General Intelligence. By endowing powerful Large

[1] ∗ Equal contribution.
[2] † Corresponding author.
[5] Note that *reasoning* throughout this paper denotes the model's general inferential ability to interpret and integrate visual and linguistic cues, rather than any explicit "thinking" paradigm.

Language Models (LLMs) with sight from Vision Transformers (ViTs) [34, 46], VLMs can perceive and reason about the world in a manner that more closely mirrors human cognition. Beneath this ambition, however, lies a profound architectural **symmetry** between ViTs and LLMs that most current designs shatter. Both progressively refine representations [2, 35, 36], yet they are connected by a crude, **asymmetric** bridge where only the top of the vision hierarchy is linked to the base of the language hierarchy. Such static and single-sided connection breaks the intrinsic hierarchical symmetry between ViTs and LLMs [2, 35, 36], leading to limited fine-grained perception and shallow multimodal reasoning.

This limitation has tangible consequences, as shown in Fig. 1 (a). A baseline VLM, relying only on final-layer features, recognizes the object as "roller skates" but overlooks critical wheel details, thus failing to assess its usability. This failure highlights a fundamental **functional mismatch**: the static, single-layer connection denies the LLM access to the full spectrum of visual evidence, preventing it from integrating local details with global semantics for nuanced reasoning.

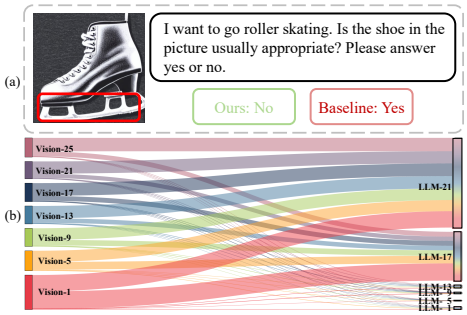

Fig. 1: (a) An illustrative failure case where our baseline model, relying solely on final-layer features, incorrectly identifies an ice skate as 'roller skates'. This error underscores the necessity of accessing fine-grained details from early vision encoder layers. (b) Visualized gating weights from our CLI framework validate the efficacy of "criss-cross connections". The heatmap reveals that deeper LLM decoder layers (right) dynamically query features from the full spectrum of vision encoder layers (left)—a many-to-many interaction that confirms our proposed fusion architecture.

An effective integration must accommodate the diverse and dynamic needs of the LLM processing hierarchy. The most intuitive requirement is a parallel alignment: shallow LLM layers, processing local syntax [36], need access to early ViT layers capturing edges and textures [2] to ground basic nouns. Similarly, deep LLM layers handling abstract reasoning [35] would logically leverage high-level scene semantics from deep ViT layers [2]. Yet, even this parallel structure is still insufficient. The demands of sophisticated multimodal reasoning also necessitate **criss-crossed connections**. For instance, when processing the prompt in Fig. 1, a shallow LLM layer parsing the word "shoe" requires immediate access to the high-level, semantic concept of the object—an abstraction formed only in the deep ViT layers—to correctly ground the noun. Simultaneously, to answer the ultimate question, "Is the shoe...appropriate?", a deep LLM layer performing this evaluative reasoning must "zoom in" on the fine-grained structural details of the wheels—information captured only in the early ViT layers. This single example powerfully illustrates the need for both types of criss-crossed connections. Indeed, our own experiments confirm this ne-

cessity, revealing a complex pattern of learned connections (Fig. 1 (b)) that move far beyond a simple one-to-one mapping.

Solving this requires a fundamental shift from the prevailing one-to-one connections to a truly dynamic many-to-many framework. However, prior research attempting to bridge this gap offers only incomplete solutions: approaches like DeepStack [32] provide a form of brute-force one-to-many injection. Other models, from EVLM [6] to the recent Qwen3-VL [33], impose a heavyweight and rigid one-to-one fusion, hard-wiring specific vision layers to pre-configured LLM layers. All such methods create a static "straitjacket", failing to provide the dynamic, arbitrary access that sophisticated reasoning demands.

We argue that the solution is to empower the LLM to become an **active observer**, capable of dynamically querying the entire visual hierarchy at the precise level of detail required by its current reasoning step. To this end, we are among the **first** to propose **Cross-Layer Injection (CLI)** (as shown in Fig. 2 (b)), a framework designed to realize an adaptive **many-to-many** architecture for VLMs. CLI consists of two synergistic, parameter-efficient components. First, **Adaptive Multi-Projection** uses Low-Rank Adaptation (LoRA) [14] to efficiently harmonize features from diverse vision layers into a shared semantic space. Second, and most critically, our **Adaptive Gating Fusion** mechanism acts as an intelligent, context-sensitive controller. At each injection point, it allows the LLM to query this multi-level visual repository and selectively integrate only the most relevant information—be it fine-grained textures from a shallow ViT layer or abstract concepts from a deep one.

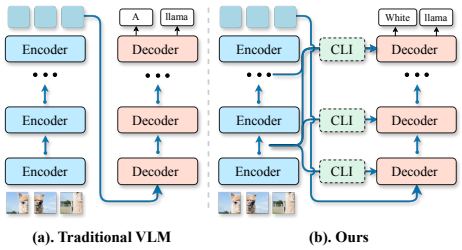

Fig. 2: (a) Conventional VLM Pipeline. Maps visual features from the encoder's final layer to the text embedding space via a projector. (b) "Many-to-Many" Bridge. Adaptively fuses features from multiple vision layers into multiple LLM layers.

To validate the effectiveness and broad applicability of our CLI framework, we integrated it into two distinct VLM architectures: LLaVA-OneVision and LLaVA-1.5. Across 28 diverse and challenging benchmarks—covering key capabilities like document analysis, chart reasoning, and general visual perception—our method significantly outperforms the baseline models and other deep fusion strategies. In particular, when applied to LLaVA-OV-7B, our method achieves performance improvements of **6.5**, **3.3**, and **4.7** points on the LLaVA-in-the-Wild, MME, and the OCR-Bench benchmarks, respectively. In summary, our main contributions are as follows:

– A novel framework, CLI, that systematically addresses the underutilization of hierarchical visual features in VLMs.

- Two synergistic and parameter-efficient modules, Adaptive Multi-Projection and Adaptive Gating Fusion, that collectively enable the alignment and dynamic, context-sensitive fusion of multi-level visual features.
- Extensive experiments on two distinct VLM architectures and a broad range of diverse benchmarks, demonstrating consistent and significant performance gains that establish a new state-of-the-art (SOTA) on several key tasks.

2 Related Works

2.1 The Dominant VLM Paradigm

The paradigm shift in AI driven by LLMs [4, 37] has extended into the visual domain, leading to the rapid emergence of VLMs. A dominant architectural blueprint, established by seminal works like Flamingo [1] and BLIP-2 [18] and popularized by LLaVA [22, 23], is rooted in parameter-efficient adaptation. This approach involves freezing the powerful pre-trained vision and language backbones and training only a lightweight "bridge" module between them. This powerful and efficient design crystallized the "pre-training + fine-tuning" paradigm that now dominates the field, sparking a wave of open-source innovation, including InstructBLIP [11], MiniGPT-4 [5, 50], Qwen-VL [3], and InternVL-2.5 [8]. Beneath this success, however, lies the fundamental architectural flaw we identify in our introduction: these models almost universally treat the output of the final vision encoder layer as the sole representation of the image, creating a severe information bottleneck and limiting the LLM's perceptual depth.

2.2 Deeper Fusion: Hierarchical Approaches

To address the above problem, subsequent research has explored deeper visual-language fusion strategies.

One-to-Many Injection: Brute-Force and Context-Blind. The first category attempts to inject features from a single image (one) into multiple LLM layers (many) through direct, non-adaptive methods. DeepStack [32] is a prominent example of this approach, which partitions high-resolution visual tokens into distinct groups and feeds each group to a different LLM decoder layer via simple element-wise addition. This brute-force mechanism is context-blind; the non-adaptive summation risks disrupting the LLM's carefully learned representations by injecting unfiltered, and potentially irrelevant, visual information. CogVLM [38] integrates "Visual Expert" modules into each LLM layer, these modules all process the same, single feature map from the vision encoder's final layer. In addition, a distinct approach is taken by CogAgent [13], whose primary goal is high-resolution efficiency. It employs a parallel architecture with a separate, lightweight encoder for high-resolution details. These features are statically broadcast to each decoder layer, meaning every layer accesses the same, fixed level of visual information. Other work, such as FUSION [25], deepens the fusion

process by introducing "interaction layers" where learnable tokens recursively engage with both textual and visual features. However, this approach focuses on the interaction and still operates on a single level of visual representation.

Statically-Wired One-to-One Fusion: Rigid and Inflexible. The second category employs dedicated modules to create fixed connections between specific vision and LLM layers, forming a kind of one-to-one mapping at different points in the hierarchy. The work of [21] systematically compared various internal fusion methods, demonstrating that simple, static approaches often fail. Flamingo-style architectures like EVLM [6] exemplify this by inserting a new cross-attention layer before each LLM layer and feeding it features from a specific, pre-assigned ViT layer. This philosophy of sparsely inserting specialized fusion blocks into a pre-configured structure is also seen in models like mPLUG-Owl3 [42], which places "Hyper Attention" blocks at predetermined layers to run cross-attention.

More recent models like Qwen3-VL [33] similarly rely on a predetermined, hard-wired connection scheme. DEHVF [39] further explores this direction by imposing a rigid, pre-defined "one-to-one" mapping between groups of vision and LLM layers. All these approaches lock the model into a manually configured, static "straitjacket" that predetermines information flow.

The proposed CLI framework is designed to overcome key limitations of prior multimodal methods. Unlike DeepStack or CogVLM, our AGF module integrates information with selectivity and precision. Furthermore, though the concurrent work DEHVF employs a rigid group mapping, our framework empowers the LLM to learn this alignment by selectively querying the entire visual hierarchy. Similarly, in contrast to the hard-wired architectures of EVLM and mPLUG-Owl3, our framework provides on-demand access to the complete visual hierarchy at all injection points. This design enables the stable and dynamic many-to-many fusion essential for sophisticated multimodal reasoning.

3 Method

This section details the Cross-Layer Injection (CLI) framework. By replacing the conventional crude connections with a dynamic **many-to-many** bridge, CLI overcomes the information bottleneck that perceptually impoverishes VLMs. We first review the standard VLM pipeline in Sec. 3.1 to contextualize its limitations. Subsequently, Sec. 3.2 details the proposed mechanism. As shown in Fig. 3, it consists of two core components. The first component is **Adaptive Multi-Projection (AMP)**, a parameter-efficient module that uses LoRA to align multi-level visual features with the language space. The second, **Adaptive Gating Fusion (AGF)**, is a query-based attention gate that dynamically integrates these aligned features into the LLM during the decoding process.

3.1 Preliminaries

The traditional VLM architecture (see Fig. 2 (a)) comprises a visual encoder, a projector, and an LLM. First, the visual encoder extracts features from an input

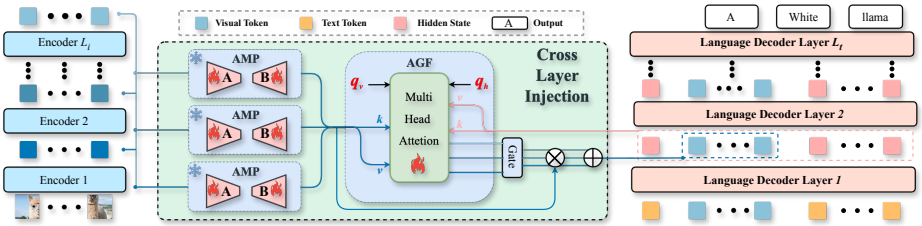

Fig. 3: An overview of the proposed Cross-Layer Injection (CLI) Framework. The left panel illustrates the overall "many-to-many" information flow, where hierarchical features from multiple vision encoder layers are dynamically injected into the LLM at multiple decoder layers. The right panel details the core **Adaptive Gating Fusion** (AGF) module that enables this process. Multi-Head Attention (MHA) first distills key information from both the incoming visual features and the current LLM hidden state. A gate controller then uses these distilled representations to compute a dynamic weight that governs the selective fusion of the new visual information. This gated process ensures the feature injection is both adaptive and context-sensitive, preventing the LLM from being overwhelmed by irrelevant data.

image I:

$$V = \text{ImageEncoder}(I), \quad V \in \mathbb{R}^{N_i \times D_i}, \tag{1}$$

where V consists of N_i visual tokens, each with a dimension of D_i. A projector module P, typically an MLP, then aligns these visual tokens with the language embedding space:

$$\hat{V} = P(V) = \text{MLP}(V), \quad \hat{V} \in \mathbb{R}^{N_i \times D_t}, \tag{2}$$

where the output dimension D_t matches the dimension of the text tokens. Simultaneously, the input text, T, containing user instructions and prompts, is embedded into a sequence of N_t text tokens. The projected visual tokens $\hat{V}$ and the text tokens $\hat{T}$ are concatenated to form a unified multimodal context, $\mathbf{H} = [\hat{V}; \hat{T}]$, on which the LLM performs auto-regressive decoding:

$$P(w_{t+1}|w_{1:t}, \mathbf{H}) = \text{LLM}(w_{1:t}, \mathbf{H}). \tag{3}$$

This dominant paradigm's critical flaw, as argued in our introduction, is its reliance on only the final-layer visual features. This rigid one-to-one connection creates a severe information bottleneck by discarding the rich hierarchical information captured in earlier ViT layers.

3.2 Many-to-Many Cross-Layer Injection

To overcome this limitation, we introduce Many-to-Many Cross-Layer Injection, a framework that restores this symmetry by forging a dynamic many-to-many bridge between the vision encoder and the LLM decoder, as shown in Fig. 3. Instead of a single point of contact, CLI empowers the LLM to dynamically

access and integrate multi-level visual features at various stages of its decoding process, conditioning its reasoning on a more comprehensive visual context.

Our method begins by extracting a hierarchical set of visual features. Instead of using only the final layer, we sample intermediate token matrices from L_V distinct layers of the vision encoder, forming a collection $\mathbb{V} = \{V_l\}_{l=1}^{L_V}$. This strategy ensures that the model can access both low-level structural features from early layers and high-level semantic concepts from later ones. For instance, sampling from layers 1, 7, and 14 would produce the feature set $\mathbb{V} = \{V_1, V_7, V_{14}\}$. To integrate these multi-level features during reasoning, we introduce cross-layer injection points at multiple layers within the LLM's decoder. This is operationalized through our novel **AMP** and **AGF**. At a specific layer in the LLM, these modules take the entire set of hierarchical visual features $\mathbb{V}$ as input, aggregate this information, and inject a synthesized representation. This allows the LLM to dynamically condition its output on the most relevant visual details, regardless of their layer of origin.

Adaptive Multi Projection (AMP). A central component of our mechanism is a specialized projection module designed to align multi-level visual features with the language embedding space. Aligning multi-level visual features presents a fundamental projection dilemma. On one hand, a single, rigid projector cannot reconcile the significant distributional variance across vision layers—from low-level textures to high-level semantics—leading to severe feature misalignment. On the other hand, the seemingly obvious alternative of training a dedicated projector for each layer is computationally prohibitive due to the extensive pre-training required. Therefore, our solution is to make the projector *adaptive* in a parameter-efficient manner. We employ LoRA to modify the behavior of the original, pre-trained projector. Specifically, for the visual tokens $V_k \in \mathbb{V}$ drawn from a sampled layer k, we augment the pre-trained MLP within the projector with a unique, layer-specific LoRA projector. This adaptive projection is:

$$\hat{V_k} = P(V_k) + \text{LoRA}(V_k) = \text{MLP}(V_k) + B_k A_k(V_k), \quad V_k \in \mathbb{V}, \tag{4}$$

where A_k and B_k are the low-rank matrices trained specifically for features from vision layer k. This adaptive projection process is applied to each feature map $V_k \in \mathbb{V}$, yielding a new set of aligned token maps $\hat{\mathbb{V}} = \{\hat{V_k}\}_{k=1}^{K}$. Each component $\hat{V_k}$ within this set is now harmonized in the text embedding dimension, making the entire hierarchical collection ready for effective integration into the LLM.

Adaptive Gating Fusion (AGF). With the visual tokens aligned, the next step is their effective integration. Rather than using a simple summation, which can disrupt the LLM's state, we propose an adaptive injection mechanism. This approach acknowledges that the LLM's hidden state, h, already contains contextual information and thus requires a more nuanced update. Accordingly, for each injection layer L_t in the LLM, a gating module is proposed to dynamically assess the relevance of the new visual features, $\hat{V}_k \in \hat{\mathbb{V}}$, based on the current decoding context h_t. The gate's logic is driven by cross-attention. Two learnable query vectors, q_v and q_h, act as probes to distill the essence of the new visual

information and the existing hidden state:

$$\hat{V}_{\text{att}} = \text{MultiHeadAttention}(q_v, \hat{V}_k, \hat{V}_k), \tag{5}$$

$$h_{\text{att}} = \text{MultiHeadAttention}(q_h, h_t, h_t). \tag{6}$$

The resulting context vectors are fused and passed through a gate controller—a linear layer followed by a Sigmoid activation—to yield a dynamic weight:

$$W = \text{Sigmoid}(\text{Gate}([\hat{V}_{\text{att}}; h_{\text{att}}])), \tag{7}$$

where $W \in [0, 1]$. This weight governs the selective update of the hidden state. To ensure precision, we use a binary "mask" that isolates the positions of visual tokens within h_t. The non-visual portions are preserved, while the visual portions are updated via a weighted sum with the new features:

$$h'_t = h_t \odot (1 - \text{mask}) + (h_t \odot \text{mask} + W \odot \hat{V}), \tag{8}$$

where $\odot$ denotes element-wise product. This fusion process enriches the hidden state h_t with a hierarchical representation of the visual input through processing all the $\hat{V}_k$ in $\hat{\mathbb{V}}$. This gating and update cycle is repeated at designated injection points throughout the LLM's decoder. This method facilitates an iterative refinement of the model's visual understanding, allowing it to "re-examine" visual evidence at varying granularities throughout the generation process.

4 Experiments

This section presents a comprehensive empirical validation of our Cross-Layer Injection (CLI) framework. We first detail the experimental setup, including our implementation of CLI on two distinct VLM architectures and the benchmarks used for evaluation. Then, the main results are presented, demonstrating that CLI consistently and significantly outperforms strong baselines and competing fusion strategies across 28 diverse benchmarks. Finally, a series of in-depth ablation studies and analyses are conducted to dissect the specific contributions of CLI's core components and validate our "many-to-many" design philosophy.

4.1 Experiment Setup

Model Configuration. To demonstrate the versatility and general applicability of CLI, we integrate it into two distinct VLM architectures, LLaVA-OneVision [17] and LLaVA-1.5 [22]. The specific components for each are as follows:

- **CLI on LLaVA-OneVision**: This model uses a Qwen-2 series model [41] as the LLM backbone, a Siglip-so400m-patch14-384 [46] vision encoder, and a two-layer MLP projector. To mitigate potential data contamination and ensure a fair evaluation, we initialize our fine-tuning from the public checkpoint released after its "High-Quality Knowledge Learning" stage.

– **CLI on LLaVA-1.5**: This variant is built upon Vicuna-7B [10] as the LLM and CLIP-Large [34] as the vision encoder, with a two-layer MLP projector using a GELU activation. Following the methodology of LLaVA-1.5, we fine-tune the model starting from the pre-trained checkpoint and using the identical training data.

We adopt the original training protocols from LLaVA-OneVision and LLaVA-1.5. Detailed hyper-parameters for our experiments are listed in Appendix A.

Dataset Configuration. The foundation of our instruction tuning set consists of the single-image datasets from LLaVA-OneVision, which provide a broad base for general visual understanding (see Appendix B for details).

Evaluation Benchmarks. To ensure a standardized and reproducible comparison, we evaluate the LLaVA-OneVision and LLaVA-1.5 models across a comprehensive suite of single-image benchmarks using the LMMs-Eval framework [47]. These benchmarks are grouped into three primary categories to assess a wide range of capabilities (details can be found at Appendix C):

– **Chart, Diagram, and Document Understanding**: AI2D [16], ChartQA [29], DocVQA [31], and InfoVQA [30]. These tasks demand fine-grained perception of text and structural elements.
– **Perception and Multidisciplinary Reasoning**: MME [12], MMBench [24], MMVet [44], MathVerse [49], MathVista [26], MMMU [45], GQA [15], OK-VQA [28], ScienceQA [27], SEED-Bench [43], MM-Star [7], and POPE [19]. These benchmarks test abstract reasoning, knowledge integration, and resistance to hallucination.
– **Real-world Understanding and Visual Chat**: RealworldQA [40], LLaVA-in-the-Wild [22]. These qualitative benchmarks evaluate the model's practical conversational and reasoning abilities in open-ended scenarios.

Comparison Methods. We evaluate our method against two sets of competitors. First, to situate our model's performance in the broader landscape, we compare it against SOTA VLMs at both similar (IXC-2.5-7B [48], InternVL-2-8B [9]) and larger scales (VILA-13B [20], InternVL-2-26B [9]). Additionally, to directly assess our fusion strategy, we conduct a controlled study comparing CLI against three fusion paradigms, all implemented on the same base models and trained on the identical dataset. Our primary comparison is against the **Baseline Projector**, the original single-layer projection method from the LLaVA architectures, which we re-trained to serve as a fair baseline (referred to as LLaVA-OV-0.5B/7B). We also compare against **DeepStack** [32], a brute-force approach that partitions high-resolution visual tokens and injects these different token groups into different LLM layers, and **Shallow-Layer Injection (SLI)**, a rigid statically-wired method used by models like Qwen3-VL [33] that creates a fixed one-to-one mapping from n vision layers sampled at a uniform stride of 8 to the corresponding initial n LLM decoder layers.

Table 1: Performance evaluation of the proposed CLI framework across a suite of nine benchmarks. We integrate our proposed Cross-Layer Injection (CLI) framework into LLaVA-OV (0.5B, 7B) and compare against baselines and alternative fusion methods.

Model	AI2D	ChartQA	DocVQA	InfoVQA	RealWorldQA	LLaVA-W	POPE	OK-VQA	GQA	Sum
	test	test	val	val	test	test	test	val	test	
VILA-13B [20]	57.6	32.8	20.0	10.0	41.9	58.6	0.4	1.6	17.6	240.5
IXC-2.5-7B [48]	39.1	81.2	90.3	68.1	57.5	63.2	88.5	29.2	57.7	574.8
InternVL-2-8B [9]	82.2	82.5	90.0	66.5	64.4	72.7	87.8	52.1	62.7	660.9
InternVL-2-26B [9]	83.0	84.4	90.4	68.9	67.2	90.6	88.8	48.3	65.1	686.7
LLaVA-OV-0.5B	56.5	64.5	64.1	**47.5**	56.0	**61.7**	88.8	48.4	53.7	541.2
w/ DeepStack [32]	53.2$_{-3.3}$	54.0$_{-10.5}$	54.6$_{-9.5}$	41.0$_{-6.5}$	52.9$_{-3.1}$	57.1$_{-4.6}$	88.2$_{-0.6}$	46.2$_{-2.2}$	53.3$_{-0.4}$	490.5$_{-50.7}$
w/ SLI [33]	**57.2**$_{+0.7}$	64.7$_{+0.2}$	63.6$_{-0.5}$	46.6$_{-0.9}$	55.1$_{-0.9}$	55.2$_{-6.5}$	89.0$_{+0.2}$	**48.8**$_{+0.4}$	54.5$_{+0.8}$	534.7$_{-6.5}$
w/ **CLI**	56.7$_{+0.2}$	**65.2**$_{+0.7}$	**64.7**$_{+0.6}$	47.1$_{-0.4}$	**56.7**$_{+0.7}$	61.1$_{-0.6}$	**89.1**$_{+0.3}$	48.6$_{+0.2}$	**53.7**	542.9$_{+1.7}$
LLaVA-OV-7B	77.5	78.5	82.5	69.5	**68.4**	68.0	88.6	58.5	59.4	650.9
w/ DeepStack [32]	76.0$_{-1.5}$	63.8$_{-14.7}$	72.6$_{-9.9}$	62.2$_{-7.3}$	61.4$_{-7.0}$	70.3$_{+2.3}$	88.7$_{+0.1}$	58.3$_{-0.2}$	58.5$_{-0.9}$	611.8$_{-39.1}$
w/ SLI [33]	77.6$_{+0.1}$	77.3$_{-1.2}$	79.9$_{-2.6}$	67.6$_{-1.9}$	68.2$_{-0.2}$	68.5$_{+0.5}$	**89.0**$_{+0.4}$	58.4$_{-0.1}$	59.3$_{-0.1}$	645.8$_{-5.1}$
w/ **CLI**	**77.9**$_{+0.4}$	**78.7**$_{+0.2}$	**82.8**$_{+0.3}$	**70.5**$_{+1.0}$	**68.4**	**74.5**$_{+6.5}$	88.9$_{+0.3}$	**59.2**$_{+0.7}$	**59.7**$_{+0.3}$	**660.6**$_{+9.7}$

4.2 Main Results

We evaluate our CLI by incorporating it into LLaVA-OneVision and LLaVA-1.5 architectures. Quantitative results compared with SOTA methods and other fusion works are shown in Tabs. 1 and 2. Overall, these results show that our CLI framework significantly outperforms the Baseline Projector (labeled LLaVA-OV-0.5B and -7B) across both model scales. This validates our central hypothesis: empowering an LLM with dynamic, on-demand access to the full visual hierarchy unlocks a deeper level of multimodal understanding. Furthermore, the performance lift from CLI is substantially larger than that of other deep fusion strategies, and it is the only method to deliver robust gains over the baseline. DeepStack, with its context-blind addition of high-resolution patch tokens, often degrades performance, falling well below the baseline and demonstrating that unfiltered feature injection is disruptive. Its strategy of partitioning visual tokens and injecting them into fixed layers via simple addition is non-selective; the LLM is forced to process pre-determined visual information regardless of its contextual relevance, which often disrupts its internal representations. Shallow-Layer Injection delivers only marginal or inconsistent gains, failing to significantly improve upon the baseline because it confines multi-level visual data to the LLM's shallowest layers, leaving the deeper, reasoning-intensive layers perceptually impoverished. In stark contrast, CLI's dynamic many-to-many architecture, governed by AGF, allows every designated LLM layer to query the entire visual hierarchy. This context-aware fusion enables a more sophisticated integration of visual evidence, leading to demonstrably superior performance.

Specifically, the benefits of CLI's architecture are evident across diverse task categories. For **fine-grained document understanding** on benchmarks like AI2D, ChartQA, DocVQA, and InfoVQA, CLI delivers cumulative gains of +1.1% and +1.9% over the Baseline Projector for the 0.5B and 7B models, respectively, by granting access to early-layer ViT features rich in essential textural and structural details. This strength extends to **complex multimodal reason-**

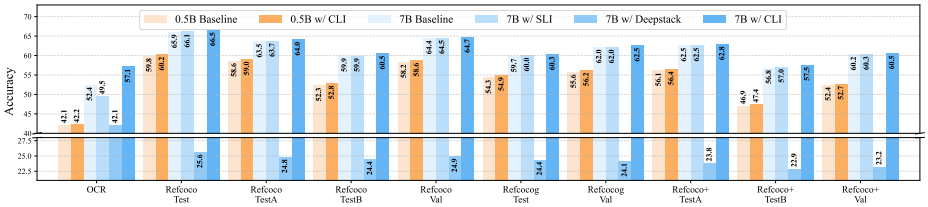

Fig. 4: Performance gains from CLI on fine-grained visual reasoning. On both OCR and a comprehensive suite of visual grounding benchmarks (RefCOCO/+/g), CLI delivers consistent improvements for both the 0.5B and 7B models.

Table 2: Main results on benchmarks for perception, multidisciplinary reasoning, and integrated capabilities. We compare our CLI-enhanced models against their baselines and alternative fusion methods on nine challenging benchmarks.

Model	MathVerse	MathVista	MMBench	MME	MMStar	MMMU	MMVet	SeedBench	ScienceQA	Sum
	mini-vision	testmini	en-dev	test	test	val	test	image	test	
VILA-13B [20]	26.4	33.7	57.9	45.2	32.8	31.5	31.2	66.5	71.0	396.2
IXC-2.5-7B [48]	13.03	60.4	66.1	90.3	14.8	33.1	46.3	47.1	22.0	380.1
InternVL-2-8B [9]	31.2	63.0	81.7	93.0	58.7	48.5	53.6	76.1	97.0	602.8
InternVL-2-26B [9]	27.3	61.1	81.6	94.9	60.4	47.5	53.1	76.7	97.5	600.1
LLaVA-OV-0.5B	17.7	36.3	45.8	63.2	38.5	30.7	28.0	65.6	65.2	391.0
w/ DeepStack [32]	17.8$_{+0.1}$	32.5$_{-3.8}$	**51.2**$_{+5.4}$	59.8$_{-3.4}$	38.5	**31.6**$_{+0.9}$	23.1$_{-4.9}$	62.9$_{-2.7}$	63.5$_{-1.7}$	380.9$_{-10.1}$
w/ SLI [33]	18.0$_{+0.3}$	**37.9**$_{+1.6}$	47.3$_{+1.5}$	61.1$_{-2.1}$	42.6$_{+4.1}$	30.2$_{-0.5}$	26.1$_{-1.9}$	65.9$_{+0.3}$	64.6$_{-0.6}$	393.7$_{+2.7}$
w/ **CLI**	**18.3**$_{+0.6}$	37.1$_{+0.8}$	48.3$_{+2.5}$	**63.3**$_{+0.1}$	**39.3**$_{+0.8}$	30.1$_{-0.6}$	27.0$_{-1.0}$	**66.7**$_{+1.1}$	**65.5**$_{+0.3}$	**395.6**$_{+4.6}$
LLaVA-OV-7B	**28.8**	55.4	79.2	85.1	54.1	47.0	**44.1**	76.1	**86.4**	556.2
w/ DeepStack [32]	27.0$_{-1.8}$	55.5$_{+0.1}$	78.1$_{-1.1}$	84.1$_{-1.0}$	50.8$_{-3.3}$	45.6$_{-1.4}$	40.3$_{-3.8}$	74.3$_{-1.8}$	83.6$_{-2.8}$	539.3$_{-16.9}$
w/ SLI [33]	25.9$_{-2.9}$	**57.8**$_{+2.4}$	**79.6**$_{+0.4}$	84.4$_{-0.7}$	55.8$_{+1.7}$	44.7$_{-2.3}$	40.6$_{-3.5}$	**76.5**$_{+0.4}$	85.8$_{-0.6}$	551.1$_{-5.1}$
w/ **CLI**	27.1$_{-1.7}$	55.8$_{+0.4}$	78.9$_{-0.3}$	**88.4**$_{+3.3}$	**56.5**$_{+2.4}$	**47.6**$_{+0.6}$	43.8$_{-0.3}$	75.7$_{-0.4}$	85.6$_{-0.8}$	559.4$_{+3.2}$

ing, where the impressive +6.5% gain on LLaVA-in-the-Wild with open-ended questions for the 7B model showcases how the AGF mechanism allows the LLM to dynamically synthesize information from the entire visual hierarchy—a capability static methods lack. Consistent gains on other reasoning-heavy benchmarks like MME (+3.3% on 7B) and SEED-Bench (+1.1% on 0.5B) further confirm that rich, multi-level visual input is a critical component for high-level cognitive tasks. This enhancement further allows our CLI-enhanced models to compete with much larger systems. Notably, our LLaVA-OV-7B with CLI achieves a total score of 660.6, closing the performance gap to reach near-parity with the larger InternVL-2-8B model (660.9).

To further challenge the framework on tasks that depend critically on multi-level visual information, we evaluated it on a separate suite of fine-grained benchmarks: Optical Character Recognition (OCR) and visual grounding (as shown in Fig. 4). These tasks act as a stress-test for vision-language fusion, as they require the model to resolve fine character strokes (for OCR) or synthesize high-level semantic context with precise low-level localization (for visual grounding). Here, CLI's superiority is unambiguous. Its ability to tap into early-layer ViT features for high-fidelity detail proves decisive, yielding a remarkable **4.7%** improvement for the 7B model on OCR. Similarly, it excels at grounding by allowing the LLM to dynamically draw on both deep ViT layers for context and shallow layers for spatial detail, leading to robust gains across **all nine** grounding benchmarks.

Table 3: To validate the architecture-agnostic nature of our method, we integrated CLI into the LLaVA-1.5 architecture. The CLI-enhanced model demonstrates consistent and broad performance gains across a wide range of tasks, confirming the robustness and general applicability of our framework.

Model	AI2D	ChartQA	DocVQA	InfoVQA	RealWorldQA	LLaVA-W	POPE	OK-VQA	GQA	**Partial Sum**
LLaVA-1.5-7B	66.3	38.9	32.2	26.8	54.5	62.9	87.0	41.8	57.6	468.0
w/ **CLI**	$65.7_{-0.6}$	$39.6_{+0.7}$	$32.4_{+0.2}$	$26.3_{-0.5}$	$54.6_{+0.1}$	$65.9_{+3.0}$	$86.4_{-0.6}$	$47.0_{+5.2}$	57.6	$475.5_{+7.5}$
Model	MathVerse	MathVista	MMBench	MME	MMStar	MMMU	MMVet	SeedBench	ScienceQA	**Partial Sum**
LLaVA-1.5-7B	17.5	34.4	64.3	79.9	37.2	34.5	31.2	61.9	72.9	433.8
w/ **CLI**	$18.3_{+0.8}$	$35.6_{+1.2}$	$66.3_{+2.0}$	$79.4_{-0.5}$	$38.5_{+1.3}$	$35.7_{+1.2}$	$34.5_{+3.3}$	61.9	$72.2_{-0.7}$	$442.4_{+8.6}$

This stands in stark contrast to the inconsistent results of competing methods as shown in Fig. 4, confirming that a dynamic, many-to-many architecture can reliably satisfy these complex visual demands.

To offer a holistic and intuitive validation of our framework, Fig. 5 displays CLI's performance gains over the baseline across 28 benchmarks, sorted by impact. CLI achieves a +**19.06** aggregate score increase, with the largest gains in challenging domains like open-ended reasoning (LLaVA-W: +6.5) and fine-grained perception (OCR: +4.7). On the few tasks where low-level features could be detrimental, CLI's gating mechanism effectively mitigates interference, confirming its architectural robustness.

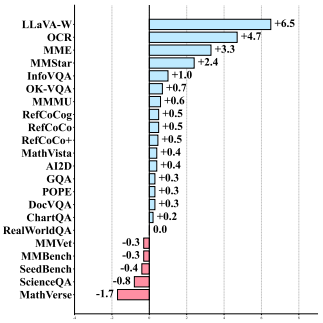

Fig. 5: Performance Gains of CLI on LLaVA-OV-7B. Absolute performance delta compared to the baseline across 28 benchmarks.

In addition, to validate its generalizability, we ported CLI to the LLaVA-1.5 architecture, which uses a different LLM, vision encoder, and pre-training pipeline. As reported in Tab. 3, CLI delivers comprehensive performance improvements to this architecture as well. The gains on LLaVA-1.5 are even more pronounced than on LLaVA-OV (+7.5 and +8.6 partial sum improvements vs. +9.7 and +3.2). This suggests a deeper advantage of our framework: CLI can act as a compensatory mechanism, mitigating pre-existing weaknesses in a base model's original visual-language alignment. By forging a richer, more dynamic connection, CLI establishes a more robust multimodal foundation, highlighting its broad utility as a plug-and-play enhancement. (A comparison with DeepStack and SLI can be found in Appendix D.)

4.3 Ablation Studies

We conducted ablation studies on LLaVA-OneVision-0.5B to analyze CLI's components and injection strategy. For efficiency, these experiments used a 50% subset of the instruction tuning data, a configuration justified by our scalability analysis (Appendix E.1). Key findings on component roles, injection density, and the AGF's internal query mechanism are summarized below, with full details provided in Appendix E.

Table 4: Ablation study of CLI components on LLaVA-OV-0.5B. Incrementally adding the AMP and AGF modules demonstrates the critical role of the gating mechanism and reveals a strong synergy between both components.

Variant	AI2D	ChartQA	DocVQA	InfoVQA	MathVerse	MathVista	MMBench	MME	MMMU	Sum	Para
	test	test	val/test	val/test	mini-vision	testmini	en-dev	test	val		
Baseline	48.87	55.88	58.87	42.21	16.71	29.40	28.52	56.38	29.88	366.72	100.00%
w/ AMP	49.13	55.76	59.19	42.75	17.04	29.60	28.26	56.28	29.88	367.89	102.25%
w/ AGF	48.67	56.60	59.47	42.48	17.01	29.10	30.58	56.47	30.22	370.60	102.12%
w/ AMP + AGF	48.64	55.40	59.32	42.52	17.85	30.00	30.41	56.71	30.66	371.51	104.37%
w/ Full AMP	49.38	56.12	59.27	42.38	17.13	28.50	29.12	56.07	29.77	367.74	107.95%
w/ Full AMP + AGF	**51.30**	**62.72**	**60.76**	**44.71**	**18.06**	**33.30**	**45.79**	**58.49**	**31.88**	**407.01**	110.07%

Dissecting the Contributions of CLI Components. A component-wise ablation (Tab. 4) reveals the critical, synergistic roles of AMP and AGF. Injecting multi-level features using only AMP provides marginal gains (367.89 vs. 366.72 baseline), confirming that simply flooding the LLM with unfiltered information is ineffective. In stark contrast, incorporating the AGF gating module alone yields a significant performance uplift (370.60), decisively validating our core hypothesis: the ability to dynamically select visual information is the most critical factor for successful fusion. The fine-tuned CLI framework, combining both modules, achieves further gains (371.51), demonstrating that AMP harmonizes the hierarchical features, providing a cleaner input for AGF to effectively gate. While a fully fine-tuned projector paired with AGF yields the highest absolute score (407.01), our LoRA-based AMP provides a much more compelling balance of performance and parameter efficiency (104.37% vs. 110.07% total params), justifying its use in our framework.

Impact of Injection Density. We next ablate the injection strategy to validate our many-to-many design (see Fig. 9 in Appendix E). Comparing various injection densities, we find that a single-point injection consistently underperforms, reaffirming that it creates an insurmountable information bottleneck. Conversely, a high-density strategy—injecting features frequently at multiple layers—achieves the best overall performance, validating our principle that on-demand access to the full visual hierarchy is essential. Intriguingly, a medium-density configuration underperforms a sparser one, suggesting a trade-off where intermittent updates may introduce disruptive cognitive overhead without the benefit of the near-continuous context provided by a high-density approach.

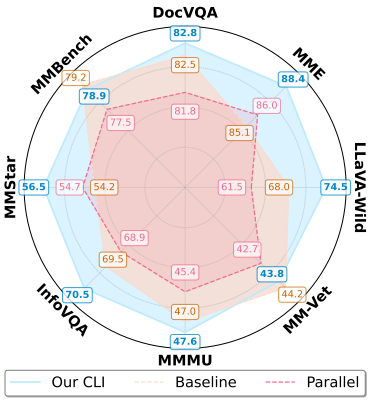

Fig. 6: Architectural Effectiveness of CLI vs. Static Parallel Fusion.

Impact of Architectural Effectiveness. To disentangle architectural benefits from parameter count, we conducted a controlled experiment with an equal-parameter setup. We designed a 'Parallel' model with a parameter count compa-

Fig. 7: Qualitative comparison of CLI on the LLaVA-OV-7B model across diverse benchmarks. Outputs from Baseline Projector are shown in red, while outputs from CLI are in green. The examples demonstrate that by integrating multi-level visual information, our model achieves more accurate perception and reasoning across various tasks, including MM-Star, LLaVA-in-the-Wild, and OCR-Bench.

rable to CLI but implemented a static, one-to-one injection scheme (i-th vision to i-th decoder layer). As shown in Fig. 6, this Parallel model frequently underperformed the baseline, dropping 6.5 points on LLaVA-Wild and yielding no significant gains on MME or MMStar. In stark contrast, CLI's dynamic architecture improved the LLaVA-Wild score by 6.5 points, creating a performance gap of over 13 points against the Parallel model, while also achieving consistent gains on other benchmarks. This result strongly indicates that CLI's performance gains stem from its architectural effectiveness, not merely an increased parameter count, thereby validating the design's complexity.

4.4 Qualitative and Efficiency Analysis

Qualitative Analysis. To offer insight into both the practical benefits and internal dynamics of our framework, we present qualitative and mechanistic visualizations. The qualitative examples in Fig. 7 highlight CLI's superior performance across diverse tasks. In complex compositional reasoning (MM-Star), fine-grained recognition (LLaVA-in-the-Wild), and challenging OCR, our model demonstrates more nuanced and accurate perception than the baseline. For instance, it correctly identifies a mangosteen where the baseline sees a durian, and reads the entire number '**3420**' where the baseline outputs a fragment, showcasing an ability to integrate both fine-grained details and global context. Additional visualizations and failure case analyses are provided in Appendix F.

The mechanism enabling this is revealed by visualizing the learned gating weights (Fig. 1 (b)). The heatmap uncovers a complex, non-parallel "criss-crossed" information flow, providing compelling evidence for our many-to-many fusion thesis. It demonstrates an evolving demand for visual information throughout the LLM's decoding process: shallow decoder layers query early vision layers to ground basic concepts, while deeper layers tasked with complex reasoning learn to query the entire visual hierarchy. This allows them to simultaneously "zoom in" on fine details (crucial for OCR) and "zoom out" for global context (vital for reasoning). Together, these visualizations confirm that granting the LLM dynamic, on-demand access to the full visual hierarchy is essential for sophisticated multimodal understanding and validates the efficacy of CLI's design.
Computational Efficiency. To quantify the resource implications of our framework, we report the computational costs for both training and inference in Tab. 5.

The results indicate that CLI introduces a modest and manageable computational overhead compared to the baseline. The memory overhead, in particular, is remarkably marginal, with the inference footprint increasing by a negligible 1.3%. This minimal impact is a critical advantage, demonstrating that our framework can be deployed without requiring significantly more VRAM. This slight increase in computational demand is decisively justified by the substantial performance improvements it unlocks, including the +19.06 aggregate score gain shown in Fig. 5. This analysis thus confirms a highly favorable trade-off and substantiates our claim that CLI is a parameter-efficient method with minimal memory impact.

Table 5: Computational Costs per sample in MMBench. (Memory indicates the Memory increases during forward).

Method	Inference			Train	
	Time	Memory	Flops	Memory	Flops
Baseline	0.66 s	241.2 MB	3.7T	251.3 MB	5.46T
CLI	0.77 s	244.4 MB	5.0T	277.6 MB	7.42T

5 Conclusion

This paper introduces CLI, a framework that resolves the information bottleneck in VLMs by creating a dynamic "many-to-many" bridge between vision and language hierarchies. By granting the LLM on-demand access to all visual layers, CLI achieves significant performance gains across 28 diverse benchmarks. Our work validates that dynamic, selective fusion is critical for sophisticated reasoning, establishing CLI as a scalable paradigm for deeper multimodal understanding.

6 Acknowledgements

This study is supported by grants from the New Generation Artificial Intelligence-National Science and Technology Major Project (No.2025ZD0123005), National Natural Science Foundation of China (No. U22A2097, No.62425208, No. 62402094), and Fundamental Research Funds for the Central Universities. In addition, this study is supported by Ant Group Research Intern Program

References

1. Alayrac, J., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J.L., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K.: Flamingo: a visual language model for few-shot learning. In: Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022 (2022)

2. Amir, S., Gandelsman, Y., Bagon, S., Dekel, T.: Deep vit features as dense visual descriptors. CoRR **abs/2112.05814** (2021)
3. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. CoRR **abs/2308.12966** (2023)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
5. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. CoRR **abs/2310.09478** (2023)
6. Chen, K., Shen, D., Zhong, H., Zhong, H., Xia, K., Xu, D., Yuan, W., Hu, Y., Wen, B., Zhang, T., Liu, C., Fan, D., Xiao, H., Wu, J., Yang, F., Li, S., Zhang, D.: EVLM: an efficient vision-language model for visual understanding. CoRR **abs/2407.14177** (2024)
7. Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., Zhao, F.: Are we on the right way for evaluating large vision-language models? In: Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024 (2024)
8. Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., Gu, L., Wang, X., Li, Q., Ren, Y., Chen, Z., Luo, J., Wang, J., Jiang, T., Wang, B., He, C., Shi, B., Zhang, X., Lv, H., Wang, Y., Shao, W., Chu, P., Tu, Z., He, T., Wu, Z., Deng, H., Ge, J., Chen, K., Dou, M., Lu, L., Zhu, X., Lu, T., Lin, D., Qiao, Y., Dai, J., Wang, W.: Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. CoRR **abs/2412.05271** (2024)
9. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., Li, B., Luo, P., Lu, T., Qiao, Y., Dai, J.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. CoRR **abs/2312.14238** (2023)
10. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (March 2023), `https://lmsys.org/blog/2023-03-30-vicuna/`
11. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.C.H.: Instructblip: Towards general-purpose vision-language models with instruction tuning. In: Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023)
12. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., Li, K., Sun, X., Ji, R.: MME: A comprehensive evaluation benchmark for multimodal large language models. CoRR **abs/2306.13394** (2023)
13. Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., et al.: Cogagent: A visual language model for gui agents. In: Proceedings

of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14281–14290 (2024)
14. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net (2022)
15. Hudson, D.A., Manning, C.D.: GQA: A new dataset for real-world visual reasoning and compositional question answering. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 6700–6709. Computer Vision Foundation / IEEE (2019)
16. Kembhavi, A., Salvato, M., Kolve, E., Seo, M.J., Hajishirzi, H., Farhadi, A.: 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. Lecture Notes in Computer Science, vol. 9908, pp. 235–251. Springer (2016)
17. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer. Trans. Mach. Learn. Res. **2025** (2025)
18. Li, J., Li, D., Savarese, S., Hoi, S.C.H.: BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, vol. 202, pp. 19730–19742. PMLR (2023)
19. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.: Evaluating object hallucination in large vision-language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023. pp. 292–305. Association for Computational Linguistics (2023)
20. Lin, J., Yin, H., Ping, W., Molchanov, P., Shoeybi, M., Han, S.: VILA: on pretraining for visual language models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024. pp. 26679–26689. IEEE (2024)
21. Lin, J., Chen, H., Fan, Y., Fan, Y., Jin, X., Su, H., Fu, J., Shen, X.: Multi-layer visual feature fusion in multimodal llms: Methods, analysis, and best practices. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 4156–4166 (2025)
22. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024. pp. 26286–26296. IEEE (2024)
23. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023)
24. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., Chen, K., Lin, D.: Mmbench: Is your multi-modal model an all-around player? In: Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 15064, pp. 216–233. Springer (2024)
25. Liu, Z., Liu, M., Chen, J., Xu, J., Cui, B., He, C., Zhang, W.: FUSION: fully integration of vision-language representations for deep cross-modal understanding. CoRR **abs/2504.09925** (2025)

26. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K., Galley, M., Gao, J.: Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. CoRR **abs/2310.02255** (2023)
27. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K., Zhu, S., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. In: Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022 (2022)
28. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: OK-VQA: A visual question answering benchmark requiring external knowledge. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 3195–3204. Computer Vision Foundation / IEEE (2019)
29. Masry, A., Long, D.X., Tan, J.Q., Joty, S.R., Hoque, E.: Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In: Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022. pp. 2263–2279. Association for Computational Linguistics (2022)
30. Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., Jawahar, C.V.: Infographicvqa. In: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022. pp. 2582–2591. IEEE (2022)
31. Mathew, M., Karatzas, D., Jawahar, C.V.: Docvqa: A dataset for VQA on document images. In: IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021. pp. 2199–2208. IEEE (2021)
32. Meng, L., Yang, J., Tian, R., Dai, X., Wu, Z., Gao, J., Jiang, Y.: Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. In: Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024 (2024)
33. Qwen: Qwen3-VL. `https://qwen.ai/` (2025)
34. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (2021)
35. Song, X., Wang, K., Li, P., Yin, L., Liu, S.: Demystifying the roles of LLM layers in retrieval, knowledge, and reasoning. CoRR **abs/2510.02091** (2025)
36. Starace, G., Papakostas, K., Choenni, R., Panagiotopoulos, A., Rosati, M., Leidinger, A., Shutova, E.: Probing llms for joint encoding of linguistic categories. In: Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023. pp. 7158–7179. Association for Computational Linguistics (2023)
37. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. CoRR **abs/2302.13971** (2023)
38. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., XiXuan, S., et al.: Cogvlm: Visual expert for pretrained language models. Advances in Neural Information Processing Systems **37**, 121475–121499 (2024)

39. Wei, X., Yang, G., Zhou, J., Yang, M., Li, L., Zhang, K., Qiu, C.: Dynamic embedding of hierarchical visual features for efficient vision-language fine-tuning. arXiv preprint arXiv:2508.17638 (2025)
40. X.AI Corp: Grok-1.5 vision preview: Connecting the digital and physical worlds with our first multimodal model. `https://x.ai/blog/grok-1.5v` (2024)
41. Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., Fan, Z.: Qwen2 technical report. CoRR **abs/2407.10671** (2024)
42. Ye, J., Xu, H., Liu, H., Hu, A., Yan, M., Qian, Q., Zhang, J., Huang, F., Zhou, J.: mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. In: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025. OpenReview.net (2025)
43. Ying, J., Chen, Z., Wang, Z., Jiang, W., Wang, C., Yuan, Z., Su, H., Kong, H., Yang, F., Dong, N.: Seedbench: A multi-task benchmark for evaluating large language models in seed science. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025. pp. 31395–31449. Association for Computational Linguistics (2025)
44. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities. In: Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net (2024)
45. Yue, X., Ni, Y., Zheng, T., Zhang, K., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024. pp. 9556–9567. IEEE (2024)
46. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023. pp. 11941–11952. IEEE (2023)
47. Zhang, K., Li, B., Zhang, P., Pu, F., Cahyono, J.A., Hu, K., Liu, S., Zhang, Y., Yang, J., Li, C., Liu, Z.: Lmms-eval: Reality check on the evaluation of large multimodal models. In: Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025. pp. 881–916. Association for Computational Linguistics (2025)
48. Zhang, P., Dong, X., Zang, Y., Cao, Y., Qian, R., Chen, L., Guo, Q., Duan, H., Wang, B., Ouyang, L., Zhang, S., Zhang, W., Li, Y., Gao, Y., Sun, P., Zhang, X., Li, W., Li, J., Wang, W., Yan, H., He, C., Zhang, X., Chen, K., Dai, J., Qiao, Y., Lin, D., Wang, J.: Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. CoRR **abs/2407.03320** (2024)
49. Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K., Qiao, Y., Gao, P., Li, H.: MATHVERSE: does your multi-modal LLM truly see the diagrams in visual math problems? In: Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings,

Part VIII. Lecture Notes in Computer Science, vol. 15066, pp. 169–186. Springer (2024)
50. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. In: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net (2024)