

# Mixpert: Mitigating Multimodal Learning Conflicts with Efficient Mixture-of-Vision-Experts

Xin He<sup>1\*</sup> Xumeng Han<sup>2,1\*</sup> Longhui Wei<sup>1†</sup> Lingxi Xie<sup>1</sup> Qi Tian<sup>1</sup>

<sup>1</sup>Huawei Inc. <sup>2</sup>University of Chinese Academy of Sciences

\*Equal Contribution <sup>†</sup>Corresponding Author

## Abstract

Multimodal large language models (MLLMs) require a nuanced interpretation of complex image information, typically leveraging a vision encoder to perceive various visual scenarios. However, relying solely on a single vision encoder to handle diverse task domains proves difficult and inevitably leads to conflicts. Recent work enhances data perception by directly integrating multiple domain-specific vision encoders, yet this structure adds complexity and limits the potential for joint optimization. In this paper, we introduce Mixpert, an efficient mixture-of-vision-experts architecture that inherits the joint learning advantages from a single vision encoder while being restructured into a multi-expert paradigm for task-specific fine-tuning across different visual tasks. Additionally, we design a dynamic routing mechanism that allocates input images to the most suitable visual expert. Mixpert effectively alleviates domain conflicts encountered by a single vision encoder in multi-task learning with minimal additional computational cost, making it more efficient than multiple encoders. Furthermore, Mixpert integrates seamlessly into any MLLM, with experimental results demonstrating substantial performance gains across various tasks.

## 1. Introduction

The breakthrough advancements in large language models (LLMs) [9, 44, 45, 52, 53] have sparked widespread interest in their potential for visual understanding and reasoning, driving researchers to develop models capable of *seeing* the real world. This demand has given rise to the emergence of multimodal large language models (MLLMs) [6, 25, 27, 29, 31, 37, 55, 60], which typically employ an architecture where images are converted by a visual encoder into a series of visual tokens and input into the LLM alongside text embeddings. The visual encoder typically employs models pre-trained on extensive image-text paired data, such as CLIP [46] and SigLIP [57], which embed images into

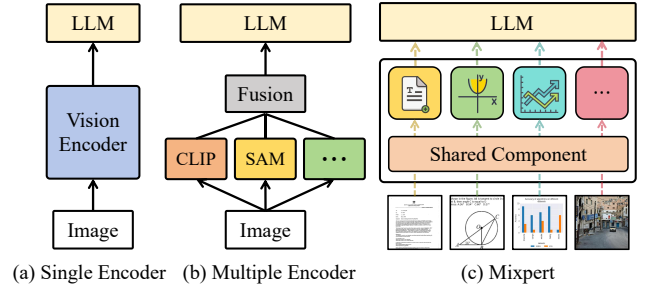


Figure 1. **Vision encoder structures.** (a) A single vision encoder handles various task scenarios. (b) Integrating multiple task-specific vision encoders to enhance data perception but introduces additional costs. (c) Our efficient mixture-of-vision-experts framework, **Mixpert**, assigns each expert to a specific domain. The input is routed to the expert most aligned with its type, incurring minimal computational overhead compared to a single encoder.

a language-consistent feature space.

Multimodal tasks encompass various visual challenges, each with significantly different image characteristics, making it difficult for a single vision encoder to achieve optimal balance across multiple domains. Recent studies [15, 29, 49, 51] employ multiple vision encoders, illustrated in Fig. 1 (b), to extract information from different perspectives, enriching visual content representation and enhancing the perception of various data types. This mode is shown to be effective and helps MLLMs better adapt to diverse visual characteristics and requirements. However, using multiple vision encoders undoubtedly increases computational and deployment burdens. Moreover, the independent nature of multiple visual encoders prevents them from gaining additional benefits through joint optimization with multimodal data, leaving each encoder reliant solely on its pre-trained capabilities. This separated structure limits the ability of vision encoders to fully leverage synergistic advantages in multimodal tasks, thereby reducing the overall optimization potential.

To address the above issues, we draw on the concept

of mixture-of-experts (MoE) [16, 17] structure and develop Mixpert to resolve domain conflicts encountered by MLLMs during multi-task joint learning in supervised fine-tuning (SFT). As show in Fig. 1 (c), we partition the vision encoder into a shared component and a multi-expert component. The shared component corresponds to the shallow layers of the vision encoder and is responsible for extracting features from all types of data, encompassing the foundational information for various visual tasks. The deeper layers and projector are structured into a mixture-of-vision-experts paradigm, where each expert focuses on a specific domain and performs more refined processing tailored to particular tasks.

In contrast to typical MoE models [13, 28], this paper manually defines the domains of experts, with each domain aligning with a specific multimodal data type. This approach allows each expert to be trained exclusively on its corresponding data type, enabling parameter decoupling for independent learning of different capabilities and mitigating potential conflicts in multitask joint optimization. Furthermore, we design a routing network to automatically select the most suitable expert for input images. The router is constructed using a simple two-layer MLP, which classifies the features extracted by the shared component of the vision encoder and assigns the image to the expert with the corresponding confidence. Since each image is allocated to only one expert, there is no additional computational overhead beyond the router. This design maintains efficient processing while significantly improving performance across tasks from different domains. The advantages of Mixpert can be summarized as follows:

- Compared to a single vision encoder, Mixpert effectively alleviates the conflicts arising from multi-task learning in SFT. During the fine-tuning of vision experts, Mixpert enables the flexible injection of task-specific data, obviating concerns around data balance across varied domains.
- For each image, Mixpert activates only the expert corresponding to the relevant data type. Consequently, the additional activation parameters and computational overhead are minimal, making it more efficient compared to solutions that rely on multiple vision encoders.
- Mixpert seamlessly integrates into any MLLM. We perform experimental validation on state-of-the-art models such as LLaVA-OV [24], InternVL2 [4], and Qwen2-VL [54], with results showing that Mixpert consistently delivers superior performance across all these models.

## 2. Related Work

**Multimodal Large Language Models (MLLMs).** In recent years, MLLMs [6, 25, 27, 29, 31, 37, 55, 60] have garnered sustained attention, primarily attempting to integrate various other modalities into LLMs. Among these, the most widely studied approaches involve encoding visual modalities

using visual encoders and then feeding them into LLMs alongside language tokens. For instance, BLIP2 [25] and InstructBLIP [6] employ a Q-Former structure to compress visual tokens extracted from CLIP-like encoders, while the LLaVA series [24, 31–33] utilize a simple MLP for projecting visual tokens into LLM embedding space. The Qwen-VL series [1, 54] and InternVL series [4, 5] utilize a pixel shuffle operation to reduce the number of visual tokens and then simply utilize MLP layers to project these tokens. In addition to methods that utilize a single vision encoder, there are also approaches that attempt to enhance the performance of MLLM by employing multiple visual encoders. SPHINX [29], IVE [15], and Cambrian-1 [51] have been proposed to incorporate multiple visual encoders to enhance the overall visual perception capability. Despite our approach also integrating multiple visual experts, it differs from the aforementioned methods by proposing to improve the perception capability for different types of visual inputs with minimal additional computation cost.

**Mixture-of-Experts (MoE).** The MoE [16, 17] models consist of multiple expert networks designed to handle different types of input data, with a router selecting the most suitable experts to process each sample. As an efficient architecture, MoE models [14, 23, 48, 59] have garnered widespread attention. To optimize resource utilization, many models [8, 10, 22] in the field of natural language processing (NLP) adopt sparse MoE architectures to handle large-scale tasks. In the quest to design more efficient MLLMs, MoCLE [13] and MoE-LLaVA [28] implement the MoE design in LLM with fewer parameter increases, but have achieved comparable performance to those with larger LLMs. In contrast to previous methods [13, 28, 30] which primarily borrowed schemes from NLP and design the MoE architectures in LLM, we have developed a more efficient MoE structure. Our approach involves designing the MoE in the visual encoder and projector layer to resolve domain conflicts of input images, and we achieve a notable reduction in parameters compared to earlier methods.

## 3. Methodology

### 3.1. Preliminary

Multimodal large language models (MLLMs) [24, 31] typically comprise three core components: *(i)* a vision encoder  $g(\cdot)$ , which encodes the input image  $\mathbf{X}_v$  into a visual embedding  $\mathbf{Z}_v = g(\mathbf{X}_v)$ ; *(ii)* a projector  $p(\cdot)$ , usually a multi-layer perceptron (MLP), which projects the image features to the word embedding space  $\mathbf{H}_v = p(\mathbf{Z}_v)$ ; and *(iii)* a large language model (LLM), which generates the response  $\mathbf{X}_a$  based on the visual embedding  $\mathbf{H}_v$  and the tokenized language instruction  $\mathbf{H}_q$ .

The training of MLLMs primarily involves pre-training and supervised fine-tuning (SFT). Taking the advanced

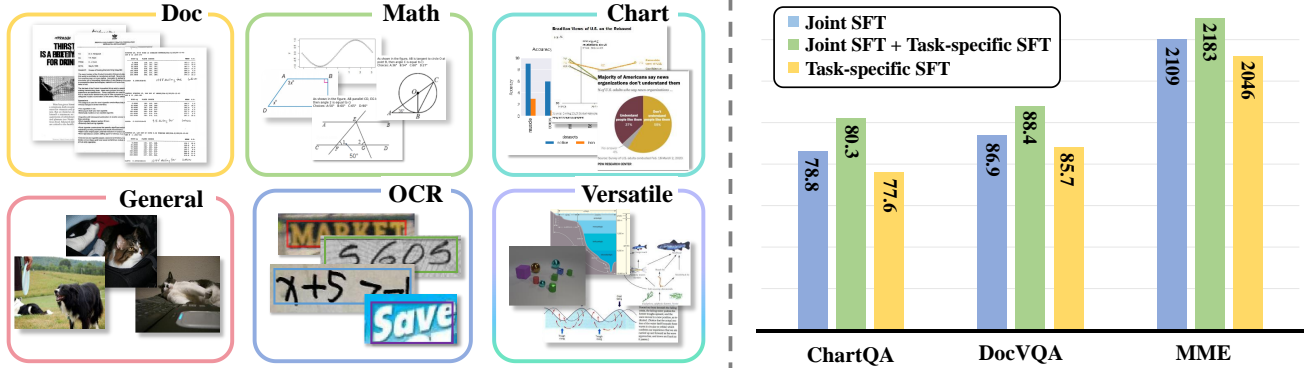


Figure 2. **(Left)** Multimodal data encompasses a variety of types, with images differing widely in content and structure. **(Right)** Building upon the joint SFT model, further domain-specific specialization can enhance performance on the respective task. This also demonstrates that joint optimization, due to balancing multiple objectives, limits the ability to achieve optimal performance in each individual domain. However, SFT on a single task alone does not yield satisfactory results, underscoring the necessity of joint learning for leveraging common knowledge across diverse data.

LLaVA-OV [24] as an example, its pre-training consists of two sub-stages: language-image alignment (*Stage-1*), which is trained on 558K data, and high-quality knowledge learning (*Stage-1.5*), where 4M data is used to inject additional knowledge. The SFT (*Stage-2*) in LLaVA-OV includes two modes: single-image (SI) training with 3.2M data and OneVision version using a mixture of video, single-image, and multi-image data. This paper only considers the single-image mode, and all subsequent mentions of LLaVA-OV refer to LLaVA-OV-7B (SI).

### 3.2. Motivation

**Domain Conflicts in Multi-task Learning.** MLLMs encounter diverse visual challenges in real-world scenarios, with each task exhibiting distinct image characteristics. As shown in Fig. 2 (Left), tasks vary significantly in terms of image content and focal details. This diversity requires the visual encoder to adapt to each type, yet it limits the ability to fully realize its potential across multiple domains. When the model attempts to learning multiple tasks concurrently, inter-task competition emerges, constraining the attainment of potential performance for each individual task. We experimentally verify the existence of domain conflicts on LLaVA-OV [24]. Specifically, starting from the joint SFT (*Stage-2*) checkpoint, we further fine-tune the model separately on the *Chart*, *Doc*, and *General* domains using corresponding visual instruction tuning data provided by LLaVA-OV [24] (more details are provided in Sec. 4.1). Here, we fine-tune all the parameters of the MLLM. As shown in Fig. 2 (Right), the task-specific fine-tuned model outperforms the original across all three domains, indicating that multi-task joint learning somewhat restricts optimal performance in each domain.

**Advantages of Joint Optimization.** While we highlight

the domain conflicts in multi-task learning, it does not suggest that joint optimization is inherently ineffective. On the contrary, joint SFT leverages a broader range of data, thereby fostering the acquisition of foundational capabilities and common knowledge. In Fig. 2 (Right), we report the results of non-joint training, *i.e.*, using the pre-trained LLaVA-OV (*Stage-1.5*) weights and fine-tuning on domain-specific data only. The results show that joint optimization significantly outperforms task-specific SFT, indicating its advantages in integrating domain knowledge, enhancing single-task performance, and fostering inter-task synergy.

**Discussion.** From the above experiments, we conclude that joint optimization enables the model to acquire robust and comprehensive foundational capabilities by exposing it to a broader spectrum of data across various scenarios. On the other hand, task-specific SFT allows the model to specialize and overcome the limitations imposed by domain conflicts. We attempt to leverage the strengths of both joint optimization and task-specific SFT, enabling the vision encoder to assimilate shared knowledge while simultaneously responding to the distinct demands of each domain.

### 3.3. Mixpert: Efficient Mixture-of-Vision-Experts

**Vision Expert Task Allocation.** We design Mixpert, as illustrated in Fig. 3, to leverage the benefits of multi-task joint learning while alleviating domain conflicts. Based on the MLLM that has undergone joint SFT, Mixpert incorporates an MoE [16, 17] structure within the vision encoder, employing multiple experts to address tasks across various domains and enabling flexible adaptation to the demands of diverse scenarios. Specifically, we partition the vision encoder (including the projector) into two parts: the shallow layers serve as a shared component  $g_s(\cdot)$  to extract common information, while the deeper layers and projector are

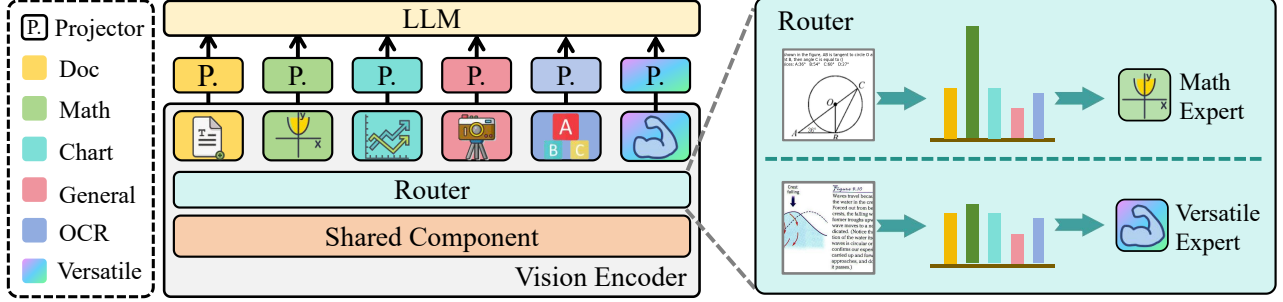


Figure 3. **Architecture of Mixpert.** We build a multi-expert component within the vision encoder, with each expert responsible for a specific task. This allows for specialization even when there are significant differences between domains, effectively mitigating potential conflicts. For images with ambiguous types or multiple characteristics, the router struggles to make a decisive prediction, thus directing them to the versatile expert for handling.

structured as an MoE, enabling specialized processing for different data through a divide-and-conquer manner.

In contrast to typical MoE models [13, 28], we do not rely on the model to learn how to partition the data and allocate inputs to the experts (the core challenge while optimizing MoE). Instead, we manually predefine the data types and assign each type to a dedicated expert. This design delineates the task allocation and specialization for each expert, thereby enhancing the interpretability of their respective roles. We categorize multimodal data into five primary types: chart analysis [18, 19, 39], document recognition [40, 50], mathematical reasoning [12, 20], OCR [43, 56], and general understanding [2, 47], assigning a corresponding expert  $e_i(\cdot)$  ( $i = 1, 2, 3, 4, 5$ ) to each data type.

**Disentangled Vision Expert Tuning.** We propose a simple yet effective disentangled vision expert tuning strategy to further enhance the specialization of each expert within its domain. We freeze all common modules, including the LLM and the shared component of the vision encoder, initialize the experts with the weights from joint SFT, and fine-tune them using domain-specific data. This approach maintains the robustness of the common modules while empowering each expert to achieve in-depth optimization within its specialized domain, thereby elevating the performance across various tasks. Another advantage of disentangled tuning is that it eliminates the need to consider the data distribution across domains, allowing each expert to learn directly from the data corresponding to its specific task.

**Versatile Expert.** In practical applications, MLLMs are tasked with addressing an infinite variety of scenarios, while the predefined data types are limited, preventing a precise and exhaustive categorization of all images. Furthermore, as illustrated in Fig. 2 (Left), some images encompass multiple domains, complicating a singular categorization based solely on visual content. To mitigate these issues, we introduce a versatile expert  $e_v(\cdot)$ , which reuses the weights from the original joint SFT, enabling flexible adaptation to vari-

ous tasks. This expert functions as a fallback mechanism, handling inputs that cannot be reliably classified.

**Dynamic Routing.** While our strategy of manually assigning expert roles allows for fine-tuning with only data corresponding to each type during training, the challenge arises when selecting the appropriate expert in the inference. The most straightforward approach would be to pre-assign a type to the input image, but this increases operational complexity and deviates from practical usage conventions. Therefore, we design a routing mechanism to automatically select the appropriate expert for the input image during inference. Specifically, the router  $r(\cdot)$  is based on an MLP to predict the probability that an image belongs to each data type and selects the expert with the highest score for processing. The image features extracted by the shared component  $g_s$  are used as input for the router. This process can be formalized as follows, where  $\sigma$  denotes Softmax:

$$\mathbf{H}_s = g_s(\mathbf{X}_v), \mathbf{H}_v = e_i(\mathbf{H}_s), i = \operatorname{argmax} \sigma(r(\mathbf{H}_s)).$$

It is worth noting that many modern MLLMs employ dynamic resolution strategies, where images are split into multiple sub-images. For simplicity, the routing prediction in our approach relies solely on the features of the entire image, with all tokens subjected to global average pooling.

The above describes a naïve routing strategy, but the highest-scoring expert may be unreliable for images that are difficult to categorize or have multiple characteristics. To address this, we route the image to the versatile expert designed to handle such challenges. Specifically, we compute the difference  $s_d = s_{(1)} - s_{(2)}$  between the highest and second-highest routing scores and compare it with a threshold  $\tau$ , which can be written as:

$$\mathbf{H}_v = \begin{cases} e_i(\mathbf{H}_s), & \text{if } s_d \geq \tau \\ e_v(\mathbf{H}_s), & \text{if } s_d < \tau \end{cases}$$

When  $s_d$  is small, it suggests that the image type is ambiguous and difficult to classify, in which case we allocate

Table 1. Details of the reorganized training datasets, all of which are sourced from the LLaVA-OV [24] collection.

<b>Chart (0.35M)</b>	<b>Math (0.33M)</b>	K12Printing	ShareGPT-4o
Chart2Tex	MAVIS MCollect	OCR-VQA	ShareGPT4V
ChartQA	MAVIS Data Engine	RenderedText	ST-VQA
DVQA	Geo170K QA	SynthDog-EN	TallyQA
FigureQA	Geometry3K	TextCaps	VisionFLAN
Infographic VQA	GeoMVerse	TextOCR	Visual7W
LRV Chart	GeoQA+	<b>General (0.94M)</b>	VisText
<b>Doc (0.30M)</b>	Geo170K Align	ALLaVA Inst	VizWiz
DocVQA	<b>OCR (0.28M)</b>	AOKVQA	VQAv2
RoBUT SQA	ChromeWriting	COCO Caption	WebSight
RoBUT WikiSQL	HME100K	LLaVA-158K	
RoBUTWTQ	IIIT5K	LLaVAR	
VisualMRC	IAM	OKVQA	

it to the versatile expert to ensure model robustness. More ablations are shown in Sec. 4.3.

## 4. Experiments

### 4.1. Datasets

**Training Datasets.** We follow the training datasets used in LLaVA-OV [24] and categorize them into five categories based on image characteristics: *Chart*, *Doc*, *Math*, *OCR*, and *General*. The details of our reorganized datasets are shown in Table 1. As mentioned in Sec. 3.3, different types of datasets will be used to train specialized experts.

**Evaluation Datasets.** Following previous works [24, 31], this work primarily focuses on single-image benchmarks. Therefore, we have selected several widely-used evaluation datasets within the field, tailored to different task types, including DocVQA test set [40], ChartQA test set [39], OCR-Bench test set [34], InfoVQA test set [41], MathVerse mini-vision set [58], MME test set [11], AI2D test set [21], MathVista testmini set [38], MMBench en-dev set [35]. The above datasets encompass lots of task types faced by multi-modal large language models (MLLMs). It is believed that a comprehensive evaluation on these datasets can well reflect the performance of MLLMs.

**Training and Evaluation Datasets for Router.** We randomly sample 200K entries from each of the five categorized datasets (shown in Table 1), constructing a dataset of 1M samples for training the router to assign the appropriate expert for each visual input. In addition, we sample 5K chart samples from the ChartQA [39] test and validation set, 5K document samples from the DocVQA [40] test set, 5K math samples from the MathVerse [58] and MathVista [38] test sets, 5K OCR samples from the IIITK5K [42] and HME100K [56] test sets, 5K general samples from the COCO Caption [3] test set, to form a validation dataset con-

sisting of 25K samples for evaluating its accuracy of router in assigning the appropriate expert for each visual input.

### 4.2. Implementation Details

**Training and Inference Pipeline.** In summary, our training pipeline comprises expert fine-tuning and router training. Initially, we fine-tuned the experts based on LLaVA-OV [24], with the categorized datasets shown in Table 1. During this phase, only the multi-expert layers are trainable, while the shared components remain frozen. Subsequently, we employ the routing dataset described in Sec. 4.1 to train the router. At the inference phase, we first extract features from the visual input via the shared visual encoder layers. Then, we feed these features into the router, which dynamically chooses the most suitable expert for each visual input based on the routing strategy. Furthermore, the selected experts will process the features extracted from the shared layers of the visual encoder and subsequently input these features into Large Language Models.

**Training Details.** While fine-tuning the added task-specific experts, each expert is trained for one epoch with a global batch size of 256. A cosine warm-up strategy is employed for adjusting the learning rate, and the maximum learning rate is set as  $1 \times 10^{-5}$  for the projection layer and  $2 \times 10^{-6}$  for the visual encoder. We train the router for one epoch with a batch size of 128, using a cosine annealing schedule, and set the maximum learning rate to  $2 \times 10^{-4}$ . AdamW [36] serves as the optimizer to train both experts and router, with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$ , and the weight decay of 0.01, respectively.

### 4.3. Ablation Study

**MoE Layer Scanning.** As described in Sec. 3.3, this work targets to design an efficient mixture-of-vision-experts architecture, which requires a trade-off between efficiency and performance improvements. To achieve this, we have conducted a comprehensive ablation study to determine which layers are most suitable for being structured as MoE layers. As shown in Table 2, allowing all layers to be restructured as MoE layers yields the best performance, *e.g.*, its accuracies on ChartQA and MathVista are improved by 1.5% and 1.7% compared to the baseline (LLaVA-OV), respectively.

However, this mechanism significantly increases the overall complexity. Therefore, it is opted to freeze several non-essential layers and adjust the remaining layers into MoE layers. From the Table 2, it is observed that as fewer layers are restructured as MoE layers, the performances gradually declines. Notably, it is found that the layers best suited for reconstruction are the projector and the deep layers of the Vision Transformer (ViT) [7]. For instance, compared to unfreezing both the projector and all layers of ViT, the mechanism that only modifying the pro-

Table 2. **Ablation study of incorporating different components into the MoE structure.** # layers indicates the number of vision encoder layers included starting from the deepest (last) layer, where 0 means without the vision encoder, and 27 represents all layers, *i.e.*, the entire vision encoder. Additional Params refers to the total additional parameters of five experts compared to the baseline (LLaVA-OV). Based on the characteristics of testing samples, we manually selected the specific expert for each evaluation benchmark.

MoE Layers			Additional Params	ChartQA	DocVQA	OCRBench	MME	MathVerse	MathVista
# layers	Projector	LLM		test	test	test	test	mini-vision	testmini
0			0B	78.8	86.9	701	2109	26.9	56.1
27	✓	✓	37.41B	80.3	88.4	769	2183	28.6	57.8
27	✓		2.14B	80.0	88.1	758	2166	28.2	57.6
12	✓		1.00B	79.9	87.9	753	2157	27.9	57.3
6	✓		0.54B	79.7	87.8	749	2151	27.7	57.1
2	✓		0.24B	79.5	87.6	735	2146	27.5	56.8
0	✓		0.09B	79.1	87.3	718	2128	27.2	56.5

Table 3. **Ablation studies on the classification accuracy of the router** trained with different scale datasets.

Data Size	Chart	Document	OCR	Math	General
500K	86.4	89.8	91.6	91.0	90.4
750K	87.9	90.6	93.9	93.2	92.5
1M	89.2	91.3	95.2	94.0	93.7

jector and the last two layers of ViT can heavily reduce the additional parameters (1.9B) while avoiding substantial performance degradation. Hence, for the trade-off between efficiency and performance, we simply chose to restructure only the projector and the last two layers of ViT as MoE layers, while keeping the rest of the network unchanged. Additionally, the above modification offers additional advantage of not requiring designing a complex router network. Instead, we can simply use two MLP layers that leverages the features extracted by the shallow layers of ViT to assign the suitable expert for each visual input.

**Performance of Router.** The performance of router directly affects the overall accuracy of our Mixpert. For instance, if the router incorrectly assigns a mathematics expert for document-type inputs, it could severely degrade the corresponding accuracy. To evaluate whether the designed router is capable of selecting the appropriate expert for visual inputs, we uniformly sampled data of different scales from our organized dataset (as described in Sec. 4.1) to train the router and evaluate its classification accuracy for different image types. As shown in Table 3, the expert assignment accuracy of the router consistently improves as the training dataset increases. Notably, when training with 1M samples, the router’s classification accuracy across various types of datasets is close to or exceeds 90%. It achieves 89.2%, 91.3%, 95.2%, 94.0%, and 93.7% on the *chart*, *document*, *OCR*, *math*, and *general*, respectively. Moreover, while training the router, only the two added MLP layers

Table 4. **Ablation studies of different routing strategies.** “Direct” refers to selecting the highest-scoring one among five task-specific experts without involving the versatile expert. “Score-threshold” and “Score-difference” route difficult-to-categorize images to the versatile expert based on predefined thresholds.

Routing	ChartQA	DocVQA	OCRBench	MME	MathVista
	test	test	test	test	testmini
Direct	79.1	87.3	722	2135	56.5
Score-threshold	79.3	87.4	728	2140	56.7
<b>Score-difference</b>	<b>79.4</b>	<b>87.5</b>	<b>732</b>	<b>2143</b>	<b>56.7</b>

are trainable and the shared component always keep frozen. Consequently, the overall training cost is relatively low.

**Evaluation on Routing Strategy.** As mentioned in Sec. 3.3, we adopt a score-difference routing strategy. However, there are also some naive routing strategies, such as the direct routing strategy and the score-threshold routing strategy. The direct routing strategy represents that directly selects the most suitable expert among the five experts according to the highest confidence. Additionally, the score-threshold routing strategy means that applies a threshold  $\lambda$  to filter the highest confidence among experts and those with highest confidence below the threshold are directly assigned to the versatile expert. To evaluate the effectiveness of these three routing strategies, we conduct a comprehensive ablation study, in which we use the router trained with 1M dataset and evaluate the above three routing strategies on ChartQA [39], DocVQA [40], OCRBench [34], MME [11] and MathVista [38] benchmark.

As presented in Table 4, the score-difference routing strategy achieves better performances across various tasks compared to the other two routing mechanisms. Specifically, compared to the direct routing, our score-difference strategy can achieve 0.3%, 0.2%, 10, 8, and 0.2% score improvements on ChartQA [39], DocVQA [40], OCRBench [34], MME [11], and MathVista [38], respectively.

Table 5. **Ablation studies of different threshold  $\tau$**  in score-difference routing strategy.

$\tau$	ChartQA test	DocVQA test	OCRBench test	MME test	MathVista testmini
0.1	79.2	87.3	725	2132	56.5
0.3	79.2	87.4	727	2133	56.5
0.5	79.3	87.5	732	2142	56.7
<b>0.6</b>	<b>79.4</b>	<b>87.5</b>	<b>732</b>	<b>2143</b>	<b>56.7</b>
0.7	79.4	87.4	731	2143	56.5

Notably, in the above experiments, we set the score-difference threshold  $\tau$  as 0.6. To further verify its influence, we further conduct ablations with different values of  $\tau$ . As shown in Table 5, while  $\tau$  is set as a small number (*e.g.*, 0.1), our score-difference strategy approximates direct routing and achieves similar results. As  $\tau$  increases beyond a certain threshold (*e.g.*, 0.5), the routing strategy becomes less sensitive to the value changes of  $\tau$ , and the results across various tasks remains nearly consistent. This observation further demonstrates that the confidence of the assigned expert for each visual input with clear data types is extremely high. As for ambiguous types of visual inputs, setting an appropriate value of  $\tau$  can effectively filter out and assign them to the versatile expert, which is the key reason why our score-difference routing strategy outperforms the direct routing mechanism.

**Additional Cost.** Once the overall architecture of Mixpert has been confirmed, the remaining question is how much additional complexity Mixpert introduces compared to the baseline (LLaVA-OV [24]). To address this, we calculate the total parameters, activated parameters, and computation cost for both LLaVA-OV and Mixpert in details. As shown in Table 6, Mixpert adds only an additional 237.1M parameters compared to LLaVA-OV. Since Mixpert only selects the most suitable expert for each visual input, its additional activated parameters and computation cost during inference phrase are primarily due to the router module, amounting to 1.3M additional parameters and 0.001G FLOPs, respectively. Compared to the overall parameters and computation overhead of LLaVA-OV, these increments are negligible.

**The Benefits from Fine-tuning.** Another issue arises from the fact that Mixpert requires additional fine-tuning for each expert. Consequently, it is pertinent to inquire whether directly fine-tune the last two vision layers and the projector would also yield additional improvements. To evaluate this, we further conduct fine-tuning on LLaVA-OV and other state-of-the-arts with all the collected expert data, respectively. As shown in Table 7, the performance gains of Mixpert primarily stems from its appropriate dynamic routing scheme rather than the additional fine-tuning process.

Table 6. **Comparisons of parameters and computation costs** between the baseline (LLaVA-OV-7B) and ours.

Method	Params	Activated Params	FLOPS
LLaVA-OV-7B	7482.1M	7482.1M	5451.157G
Mixpert (LLaVA-OV-7B)	7719.2M	7483.4M	5451.158G

Table 7. **Additional training.** \*: the results achieved by further fine-tuning the last two layers of the visual encoder and the projector.

Model	ChartQA test	DocVQA test	OCRBench test	MME test	MathVista testmini
LLaVA-OV-7B*	78.7	86.7	694	2090	56.3
Mixpert(LLaVA-OV-7B)	<b>79.4</b>	<b>87.5</b>	<b>732</b>	<b>2143</b>	<b>56.7</b>
InternVL2-8B*	83.0	91.4	789	2207	57.9
Mixpert(InternVL2-8B)	<b>84.0</b>	<b>92.3</b>	<b>806</b>	<b>2271</b>	<b>58.8</b>

#### 4.4. Comparisons with State-of-the-Arts

**Comparisons with Fully Open-Source Methods.** As a well-known open-source project in the field of MLLMs, LLaVA series [24, 31, 33] have fully released both their training data and well-trained model weights. Therefore, the proposed Mixpert is primarily built upon LLaVA-OV [24] to conduct comprehensive and fair comparisons on various multimodal benchmarks. As described in Sec. 4.1, the training of all newly added experts and the router in Mixpert is carried out using the same data corpus utilized by LLaVA-OV. Thus, the improvements of Mixpert over LLaVA-OV are primarily due to the methodology itself, rather than new training datasets.

As shown in Table 8, Mixpert demonstrates varying degrees of improvement over LLaVA-OV [24] across multiple multimodal benchmarks. Specifically, there is a 0.6% improvement compared to LLaVA-OV-7B [24] on both ChartQA [39] and DocVQA [40]. In the OCRBench [34] and MME [11], Mixpert achieves scores of 732 and 2143, surpasses LLaVA-OV-7B [24] with 31 and 34, respectively. In other benchmarks, such as MathVerse [58] and MathVista [38], Mixpert has also achieved clear improvements. Furthermore, LLaVA-OV(MI) is the version trained on multi-image datasets. Considering that each image or sub-image splitted from a single image may belong to distinct categories, we also conduct experiments with independent routing strategy for each sub-image. As shown in Table 8, indeed, compared to the global-image routing strategy, the sub-image routing strategy brings more improvements. Of course, adhering to the sub-image routing strategy also introduces additional costs, specifically increasing the activated parameters for each test image. This is the reason why Mixpert directly opts for global-image routing. Additionally, when compared to recent methods that incor-

Table 8. **Comparisons between Mixpert and other state-of-the-arts** across different types of commonly used benchmarks. \*: the evaluation using chain-of-thought prompting; †: the results tested by ourselves with official checkpoints; MI: the multi-image version of LLaVA-OV; §: the results achieved by the sub-image routing strategy rather than the global-image routing strategy.

Model	ChartQA test	DocVQA test	OCRBench test	MME test	AI2D test	InfoVQA test	MathVerse mini-vision	MathVista testmini	MMBench en-dev
<b>Open-source</b>									
Cambrian-34B [51]	75.6	75.5	-	-	79.7	-	-	53.2	81.4
Eagle-X5-13B [49]	71.0	-	573	1604	-	-	-	39.7	70.5
SPHINX-MoE [30]	55.0	68.4	-	1852	55.6	41.8	-	-	71.3
Mini-Gemini-35B [27]	-	-	-	2141	-	-	-	43.3	80.6
CuMo (Mistral-8x7B) [26]	-	-	-	1640	-	-	-	38.2	75.3
LLaVA-OV-7B [24]	78.8	86.9	701	2109	81.6	65.3	26.9	56.1	81.7
LLaVA-OV-7B(MI) [24]	80.0	87.5	621	1998	81.4	68.8	26.2	63.2	80.8
<b>Mixpert (LLaVA-OV-7B)</b>	<b>79.4</b>	<b>87.5</b>	<b>732</b>	<b>2143</b>	<b>82.0</b>	<b>65.9</b>	<b>27.4</b>	<b>56.7</b>	<b>82.2</b>
<b>Mixpert (LLaVA-OV-7B(MI))</b>	<b>81.1</b>	<b>88.3</b>	<b>659</b>	<b>2021</b>	<b>81.8</b>	<b>69.4</b>	<b>27.0</b>	<b>63.8</b>	<b>81.5</b>
<b>Mixpert (LLaVA-OV-7B(MI))<sup>§</sup></b>	<b>81.4</b>	<b>88.6</b>	<b>667</b>	<b>2030</b>	<b>82.0</b>	<b>69.8</b>	<b>27.1</b>	<b>64.0</b>	<b>81.9</b>
<b>Open-weights</b>									
MiniCPM-V2.6 [55]	-	90.8	852*	2348*	82.1	-	-	60.6	-
InternVL2-8B [4]	83.3	91.6	794	2210	83.8	74.8	27.5	58.3	81.7
Qwen2-VL-7B <sup>†</sup> [54]	82.9	94.4	863	2328	82.8	76.6	25.8	58.1	82.9
<b>Mixpert (InternVL2-8B)</b>	<b>84.0</b>	<b>92.3</b>	<b>806</b>	<b>2271</b>	<b>84.1</b>	<b>75.3</b>	<b>27.9</b>	<b>58.8</b>	<b>82.3</b>
<b>Mixpert (Qwen2-VL-7B)</b>	<b>83.4</b>	<b>94.8</b>	<b>873</b>	<b>2346</b>	<b>83.2</b>	<b>76.9</b>	<b>26.4</b>	<b>58.4</b>	<b>83.4</b>

porating multiple vision encoders or designing MoE architectures (e.g., EAGLE [49] and CuMo [26]), Mixpert still shows competitive results across various tasks.

**Comparisons with Open-Weights Methods.** Many state-of-the-art methods have released model weights, yet not training data. To further validate the effectiveness of Mixpert, we integrate it into two recent well-known MLLMs: InternVL2-8B [4] and Qwen2-VL-7B [54]. Notably, the visual encoder used in InternVL2-8B [4] is InternViT [5], whereas Qwen2-VL-7B [54] employs a self-designed ViT architecture with support for naive dynamic resolution of inputs. However, for both InternVL2-8B [4] and Qwen2-VL-7B [54], we only restructure the corresponding projector module and the last two layers of the visual encoder into MoE layers. We collect approximately 5.5M samples from publicly available datasets, which are simply categorized into three types: *chart*, *document*, and *general*. As a result, Mixpert includes four experts: versatile, chart, document, and general experts. Additionally, 3M images are randomly sampled from the above-collected datasets and used to train the router.

As indicated by the results in Table 8, even built upon state-of-the-art methods that are trained on unknown training data, Mixpert is still able to improve the performance across various benchmarks. Compared to InternVL2-8B [4], Mixpert demonstrates a significant improvement of 0.7%, 0.7%, and 61 on ChartQA [39], DocVQA [40], and

MME [11], respectively. Even evaluated on the currently most powerful open-weights model, Qwen2-VL-7B [54], Mixpert still achieves consistent improvements. The above results further demonstrate that Mixpert can integrate seamlessly into any MLLM, effectively mitigating the underlying multi-task learning conflicts, which further enhances the visual perception capabilities of MLLMs while handling different types of visual inputs.

## 5. Conclusion

This work firstly reveals the multi-task learning conflicts within a single visual encoder that are commonly faced by current multimodal large language models (MLLMs), and then points out that recent works aiming to enhance visual perception capabilities by directly integrating multiple visual encoders will inevitably add heavy computation overhead. To address this, this paper further proposes Mixpert, an efficient mixture-of-vision-experts architecture to assign the most suitable domain-specific expert for each visual input. Comprehensive experiments have substantiated the effectiveness of Mixpert, which can seamlessly integrate into any MLLM and significantly improve the corresponding performances on various scenarios with less additional cost. We believe that Mixpert is an initial attempt to address the multi-task learning conflicts while minimizing the additional complexity in MLLMs, and more efficient approaches are worth exploring in future work.

## References

- [1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 2
- [2] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023. 4
- [3] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5
- [4] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 2, 8
- [5] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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, 2024. 2, 8
- [6] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 1, 2
- [7] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5
- [8] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022. 2
- [9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1
- [10] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. 2
- [11] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 5, 6, 7, 8
- [12] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wan-jun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023. 4
- [13] Yunhao Gou, Zhili Liu, Kai Chen, Lanqing Hong, Hang Xu, Aoxue Li, Dit-Yan Yeung, James T Kwok, and Yu Zhang. Mixture of cluster-conditional lora experts for vision-language instruction tuning. *arXiv preprint arXiv:2312.12379*, 2023. 2, 4
- [14] Xumeng Han, Longhui Wei, Zhiyang Dou, Zipeng Wang, Chenhui Qiang, Xin He, Yingfei Sun, Zhenjun Han, and Qi Tian. Vimoe: An empirical study of designing vision mixture-of-experts. *arXiv preprint arXiv:2410.15732*, 2024. 2
- [15] Xin He, Longhui Wei, Lingxi Xie, and Qi Tian. Incorporating visual experts to resolve the information loss in multimodal large language models. *arXiv preprint arXiv:2401.03105*, 2024. 1, 2
- [16] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 2, 3
- [17] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2): 181–214, 1994. 2, 3
- [18] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2018. 4
- [19] Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486*, 2022. 4
- [20] Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. Geomverse: A systematic evaluation of large models for geometric reasoning. *arXiv preprint arXiv:2312.12241*, 2023. 4
- [21] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 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, 2016. 5
- [22] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 2
- [23] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 2
- [24] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2, 3, 5, 7, 8
- [25] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 2

- [26] Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts. *arXiv preprint arXiv:2405.05949*, 2024. 8
- [27] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 1, 2, 8
- [28] Bin Lin, Zhenyu Tang, Yang Ye, Jiayi Cui, Bin Zhu, Peng Jin, Jinfa Huang, Junwu Zhang, Yatian Pang, Munan Ning, et al. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 2, 4
- [29] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023. 1, 2
- [30] Dongyang Liu, Renrui Zhang, Longtian Qiu, Siyuan Huang, Weifeng Lin, Shitian Zhao, Shijie Geng, Ziyi Lin, Peng Jin, Kaipeng Zhang, et al. Sphinx-x: Scaling data and parameters for a family of multi-modal large language models. *arXiv preprint arXiv:2402.05935*, 2024. 2, 8
- [31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 2, 5, 7
- [32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.
- [33] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2, 7
- [34] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Chenglin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: On the hidden mystery of ocr in large multimodal models, 2024. 5, 6, 7
- [35] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pages 216–233. Springer, 2025. 5
- [36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5
- [37] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024. 1, 2
- [38] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 5, 6, 7
- [39] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 4, 5, 6, 7, 8
- [40] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 4, 5, 6, 7, 8
- [41] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 5
- [42] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC-British machine vision conference*. BMVA, 2012. 5
- [43] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019. 4
- [44] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1
- [45] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1
- [46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1
- [47] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer, 2022. 4
- [48] N Shazeer, A Mirhoseini, K Maziarz, A Davis, Q Le, G Hinton, and J Dean. The sparsely-gated mixture-of-experts layer. *Outrageously large neural networks*, 2017. 2
- [49] Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, et al. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv preprint arXiv:2408.15998*, 2024. 1, 8
- [50] Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13878–13888, 2021. 4
- [51] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. 1, 2, 8
- [52] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1
- [53] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov,

- Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. [1](#)
- [54] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. [2](#), [8](#)
- [55] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. [1](#), [2](#), [8](#)
- [56] Ye Yuan, Xiao Liu, Wondimu Dikubab, Hui Liu, Zhilong Ji, Zhongqin Wu, and Xiang Bai. Syntax-aware network for handwritten mathematical expression recognition. *arXiv preprint arXiv:2203.01601*, 2022. [4](#), [5](#)
- [57] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. [1](#)
- [58] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2025. [5](#), [7](#)
- [59] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022. [2](#)
- [60] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. [1](#), [2](#)