

MouSi: Poly-Visual-Expert Vision-Language Models

Xiaoran Fan*, Tao Ji*, Changhao Jiang*, Shuo Li*, Senjie Jin*,

Sirui Song, Junke Wang, Boyang Hong, Lu Chen,
Guodong Zheng, Ming Zhang, Caishuang Huang,
Rui Zheng, Zhiheng Xi, Yuhao Zhou, Shihan Dou, Junjie Ye, Hang Yan,

Tao Gui[†], Qi Zhang[‡], Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang

Fudan NLP Lab & Fudan Vision and Learning Lab

Abstract

Current large vision-language models (VLMs) often encounter challenges such as insufficient capabilities of a single visual component and excessively long visual tokens. These issues can limit the model’s effectiveness in accurately interpreting complex visual information and over-lengthy contextual information. Addressing these challenges is crucial for enhancing the performance and applicability of VLMs. This paper proposes the use of ensemble experts technique to synergizes the capabilities of individual visual encoders, including those skilled in image-text matching, OCR, image segmentation, etc. This technique introduces a fusion network to unify the processing of outputs from different visual experts, while bridging the gap between image encoders and pre-trained LLMs. In addition, we explore different positional encoding schemes to alleviate the waste of positional encoding caused by lengthy image feature sequences, effectively addressing the issue of position overflow and length limitations. For instance, in our implementation, this technique significantly reduces the positional occupancy in models like SAM, from a substantial 4096 to a more efficient and manageable 64 or even down to 1. Experimental results demonstrate that VLMs with multiple experts exhibit consistently superior performance over isolated visual encoders and mark a significant performance boost as more experts are integrated.

We have open-sourced the training code used in this report. All of these resources can be found on our project website¹.

1 Introduction

Current large vision-language models (VLMs) demonstrate significant potential in tasks requiring joint visual and linguistic perception, such as image captioning [1], visual question answering [2], visual grounding [3], and autonomous agents [4, 5]. VLMs harness large language models (LLMs) as cognitive foundation models to empower various vision-related tasks, while **one vision component**, such as CLIP [6], typically serves as auxiliary modules that provide additional visual perception [7]. However, the perception abilities of the individual vision models still lag behind, even in simple

* Equal contributions.

[†] Correspondence to: {tgui, qz}@fudan.edu.cn

¹ <https://github.com/FudanNLPLAB/MouSi>

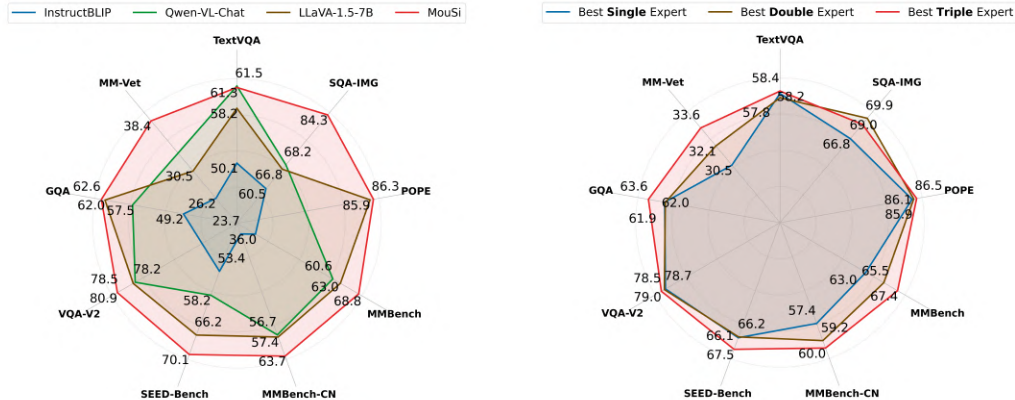


Figure 1: Left: Comparing InstructBLIP, Qwen-VL-Chat, and LLaVA-1.5-7B, poly-visual-expert **MouSi** achieves SoTA on a broad range of nine benchmarks. Right: Performances of the best models with different numbers of experts on nine benchmark datasets. Overall, triple experts are better than double experts, who in turn are better than a single expert.

tasks like counting. [8–10]. This gap highlights a significant limitation in these models’ capacity to process and understand visual information as effectively as they handle linguistic data. According to the operation of the vertebrate visual system, with each functional unit encoding different visual aspects in parallel, retinal ganglion cells transmit distinct features to the brain [11]. This biological mechanism suggests **a model structure where the varied visual information should be parallelly encoded by diverse perception channels.**

To this end, the community has verified that each model, with its unique approach to vision processing, contributes differently to understanding visual content [12]. CLIP, with its contrastive learning approach, excels in aligning images with textual descriptions, providing a robust semantic understanding [6]. DINOv2, through its self-supervised learning paradigm at both the image level and patch level, offers significant advances in robust and stabilized feature extraction without relying on labeled data [13]. LayoutLMv3’s specialization in document AI tasks demonstrates the power of visual text processing [14]. [15] empirically investigated different visual tokenizers pre-trained with dominant methods (i.e., DeiT [16], CLIP, MAE [17], Dino [18]), and observed that CLIP could capture more semantics, whereas the other models excelled at fine-grained perception. However, on the multimodal leaderboard organized by OpenCompass², the visual encoders of all open-source VLMs are based on the pre-trained CLIP encoder family. Many researchers have pointed out the shortcomings of the CLIP encoder, such as the inability to reliably capture even basic spatial factors of images [19], suffering from object hallucination [20], and so on. In light of the distinct capabilities and limitations of these diverse vision models, a key question emerges: **How can we incorporate the strengths of multiple visual experts so that they work in synergy to improve overall performance?**

Drawing inspiration from biology, we take on the poly-visual-expert perspective and design a novel model, similar to how the vertebrate visual system operates. Consequently, in the process of developing VLMs with poly-visual experts, three problems are in major concern: (1) whether the poly-visual experts are effective; (2) how to better integrate multiple experts; and (3) how to avoid exceeding the LLM’s maximum length with multiple visual experts?

In order to verify whether multiple visual experts are effective for VLMs, we construct a candidate pool consisting of six well-known experts, including CLIP, DINOv2, LayoutLMv3, Convnext [21], SAM, and MAE. Using LLaVA-1.5 as the base setup, we explored single-expert, double-expert combinations, and triple-expert combinations in eleven benchmarks. The results, as shown in Figure 1, indicate that as the number of visual experts increases, the VLMs acquire richer visual information (due to more visual channels), and the upper limit of the multimodal capability improves across the board.

² <https://rank.opencompass.org.cn/leaderboard-multimodal>

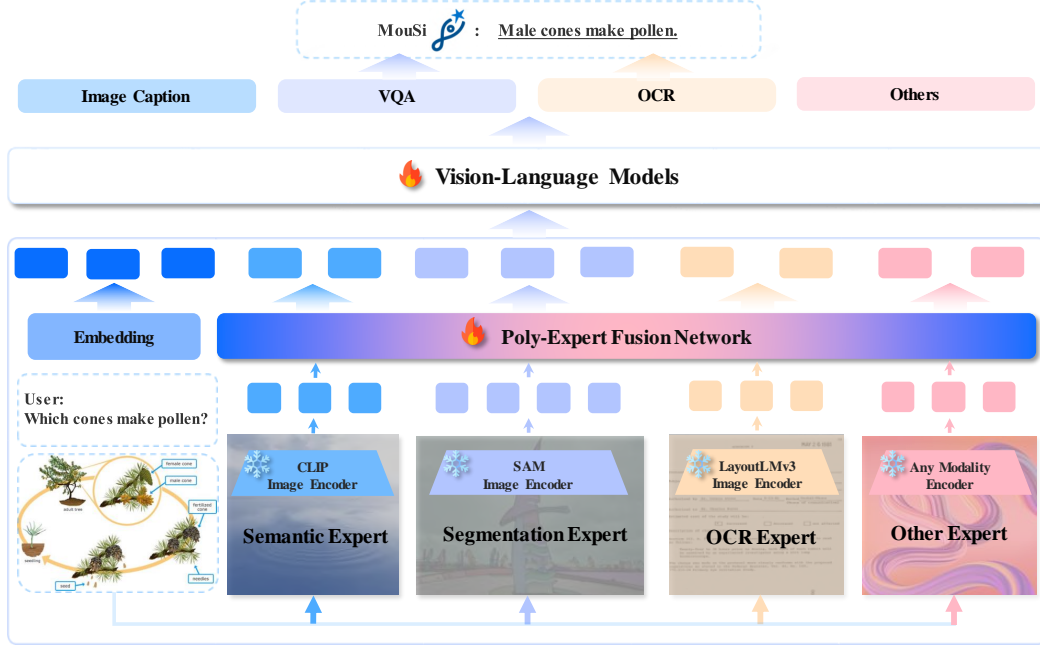


Figure 2: An overview of the MouSi model structure. The poly-vision-expert MouSi model supports the integration of visual experts with various types and capabilities.

In existing single visual channel VLMs, the methods for transmitting visual signals are either the MLP projection network [22, 23] or the Q-Former network [24, 25]. To accommodate multi-channel signal transmission from multiple experts, we modified both methods for poly-expert fusion networks separately. The proposed method also compresses the local visual information by multi-patch-one-token for better transmission efficiency and reduces the quadratic computational cost of subsequent processing of VLMs.

In position-aware VLMs, vision tokens consume a staggering amount of positional embeddings. Taking a single-turn multimodal dialogue in VQA as an example, with the MAE expert, the number of vision tokens (about 4096) is more than 500 times higher than the number of text tokens (about 8.7). Inspired by the fact that visual experts already have positional encodings, we believe it is redundant to again assign a VLM position embedding to each visual token individually. Therefore, we explore different positional encoding schemes to effectively address the issue of position encoding waste. The results show that the two schemes: sharing one position for all patches and 2D positional encoding (rows plus columns) are able to reduce the position consumption (in the case of CLIP, the PE used drops from 576 to 24 or even 1), while the performance is still comparable.

Our contributions can be summarized as follows:

- We introduce a poly-visual-expert VLM that synergistically combines the strengths of various vision encoders to improve the overall capabilities of VLMs.
- We tackle the challenge of vision token overflow in VLMs by proposing multi-patch-single-token projection and efficient positional encoding solutions.
- By experimenting with different combinations of experts, our results demonstrate enhanced performance (+1.53 with fair comparison) in multimodal tasks.

2 Architecture

2.1 The Overview

When a user uploads an image of wind pollination in a conical inflorescence and asks “Which cones make pollen?” the image is processed in sequence through the encodings of the CLIP expert, the SAM expert, and the LayoutLM expert, yielding three sets of visual representations. Subsequently, a poly-

Expert	Res.	Param.	d_hid	#Patch	Type	Pre-training	
						Tasks	Images
CLIP	336	300M	1024	576	ViT	Image-Text Matching	400M
DINOv2	224	1.1B	1536	256	ViT	DINO+iBOT+SwAV	142M
LayoutLMv3	224	368M	1024	196	ViT	Document OCR	11M
ConvNeXt	384	200M	768	1024	CNN	Image Classification	2B
SAM	1024	637M	1280	4096	ViT	Image Segmentation	11M
MAE	224	630M	1280	256	ViT	Patch-level Denoising	1.3M

Table 1: Comparison of six pre-trained visual experts. **Res.** indicates image resolution, **d_hid** indicates hidden dimension and **Param.** indicates the number of parameters.

expert fusion network compresses the multi-channel visual information and aligns it multimodally to the vision input tokens for MouSi. The user’s question is processed into text tokens by the LLMs’ Embedding layer. Finally, MouSi generates the correct answer “Male cones make pollen.” by employing its VQA capabilities to understand the vision-language question, and its OCR capabilities to recognize the answer text from the image.

In order to accomplish the above task, we propose MouSi, which consists of three fundamental components:

1. a multi-expert visual encoder, which combines the experts selected from a pool;
2. a poly-expert fusion network, which is implemented as a simple projection fusion method or a Q-Former fusion method [26];
3. a pre-trained open-source LLM (e.g., *Vicuna v1.5*).

Figure 2 shows an overview of the MouSi architecture. The core of a Vision-Language Model (VLM) is typically an LLM which is pre-trained on large-scale textual corpus. In order to perceive the visual signals, a vision encoder and vision-language connection layer are adopted to separately extract the visual features and align them to the semantic space of LLM.

The VLM takes as input a sequence comprised of interleaved text and image segments, denoted as $X = (\dots, T_1, I_1, T_2, I_2, \dots)$, where text fragments T are processed by the tokenizer and embedding layer of the LLM, and image segments I are fed to the vision encoder. To ensure the universality and generalizability of the vision encoder, it is common practice to freeze its pre-trained parameters. In this paper, we rethink the design of the visual encoder in VLMs and aim to improve its capability by ensembled experts.

2.2 Multi-Expert Vision Encoder

After extensive investigation, we choose six vision encoders skilled in different domains, including CLIP [6], DINOv2 [13], LayoutLMv3 [14], Convnext [21], SAM [27], and MAE [17]. They differ significantly from each other in terms of input resolution, hidden size, model type, model size, pre-training tasks, and training methods, as shown in Table 1.

CLIP learns the image-text alignment through contrastive learning. It is pre-trained on a large-scale dataset consisting of 400M noisy image-text pairs sourced from the internet. The vision encoder of CLIP is a Vision Transformer (ViT) with 300M parameters. The input resolution is fixed to 336×336 , and the feature dimension is 1024.³

DINOv2 trains a student network to mimic the behavior of a more powerful teacher network, without the need for any training labels. Two objective functions are utilized for self-supervised pretraining: an image-level object that constrains the CLS tokens from the student network and teacher network, and a patch-level object that is applied to the extracted representations of masked input. The Dinov2 vision encoder is a Vision Transformer (ViT) with 1.1B parameters. The input image is preprocessed to 224×224 resolution and the hidden dimension is 1536.⁴

³<https://huggingface.co/openai/clip-vit-large-patch14-336>

⁴<https://huggingface.co/facebook/dinov2-giant>

LayoutLMv3 pre-trains multimodal Transformers for Document AI with unified text and image masking. The simple unified architecture and training objectives make LayoutLMv3 a general-purpose model for both text-centric and image-centric Document AI tasks. The LayoutLMv3 vision encoder is a ViT architecture with 368M parameters. The input image is first preprocessed to the resolution of 224×224 and then encoded to 1024-dimension patch embeddings.⁵

Convnext is a purely convolutional network (ConvNet) that introduces a fully convolutional masked autoencoder framework (FCMAE) and a new global response normalization (GRN) layer to ConvNeXt. ConvNeXt underwent pretraining on the ImageNet-22K dataset, significantly enhancing the performance of the pure ConvNet across various recognition benchmarks. The ConvNeXt vision encoder we used has 200M parameters. The input resolution is 384×384 and the feature dimension is 768.⁶

SAM is trained on a large-scale segmentation dataset, comprising 11 million images and over 1 billion masks, and achieves impressive zero-shot generalization. It is designed to efficiently predict object masks from images with different types of prompts, e.g., text or point. SAM also adopts ViT as a vision encoder with 637M parameters. The input resolution and hidden dimension are both larger, i.e., 1024×1024 and 1280, respectively.⁷

MAE aims to reconstruct the original image given only partial observations (25% of the patches). The ViT-Huge encoder paired with MAE achieved a new record at the time on the ImageNet-1K dataset with an accuracy of 87.8% and generalized very well. The MAE vision encoder has 630M parameters, while input resolution and hidden dimension are 1024×1024 and 1280.⁸

Given a image I in the input sequence and a vision expert encoder $e_i(\cdot)$, we can obtain the representation vectors of n image patches:

$$\mathbf{v}_1^i, \mathbf{v}_2^i, \dots, \mathbf{v}_n^i = e_i(I). \quad (1)$$

Assuming we have three experts ($e_i(\cdot) \in \mathbb{R}^{n_i \times d_i}$, $e_j(\cdot) \in \mathbb{R}^{n_j \times d_j}$, $e_k(\cdot) \in \mathbb{R}^{n_k \times d_k}$), the final sequence of image representations V_I is a concatenation of the three output sequences.

$$\begin{aligned} V_I &= e_i(I) \oplus e_j(I) \oplus e_k(I) \\ &= [\mathbf{v}_1^i, \dots, \mathbf{v}_{n_i}^i, \mathbf{v}_1^j, \dots, \mathbf{v}_{n_j}^j, \mathbf{v}_1^k, \dots, \mathbf{v}_{n_k}^k] \end{aligned} \quad (2)$$

It is worth noting that each expert outputs a different number (n_i vs. n_j vs. n_k) and dimension (d_i vs. d_j vs. d_k) of representations, and we will handle these differences in the poly-expert fusion network. In addition, the order of the experts could also have an impact on the results, which we specifically evaluate in the ablation experiments (Section 3.2.2).

2.3 Poly-Expert Fusion Network

Since the dimension and number of output sequences are often different for different visual experts, a fusion network needs to be designed to unify the processing. Following LLaVA [7] and BLIP [28], we propose an MLP projection fusion network and a Q-Former fusion network, respectively.

MLP projection is a 2-layer ($d_{in} \rightarrow d_{hidden} \rightarrow d_{out}$) multilayer perceptron network. To simplify the processing and to share the knowledge among multiple experts, we set the hidden dimension (d_{hidden}) and the output dimension (d_{out}) equal to the dimension (d_{model}) of the LLM, and the second layer network ($MLP^{(2)} : d_{hidden} \rightarrow d_{out}$) parameters are shared among all experts. Given a specific expert $e_i(\cdot)$, the first layer network is defined as $MLP_i^{(1)} : d_i \rightarrow d_{hidden}$.

$$V_I = MLP^{(2)} \left(MLP_i^{(1)}(e_i(I)) \oplus MLP_j^{(1)}(e_j(I)) \oplus MLP_k^{(1)}(e_k(I)) \right) \quad (3)$$

In practice, multiple experts output a large number of vision tokens, which not only increases the computational cost and memory usage of the VLM but also tends to exceed the maximum length

⁵<https://huggingface.co/microsoft/layoutlmv3-large>

⁶https://huggingface.co/laion/CLIP-convnext_large_d_320.laion2B-s29B-b131K-ft-soup

⁷<https://huggingface.co/facebook/sam-vit-huge>

⁸<https://huggingface.co/facebook/vit-mae-huge>

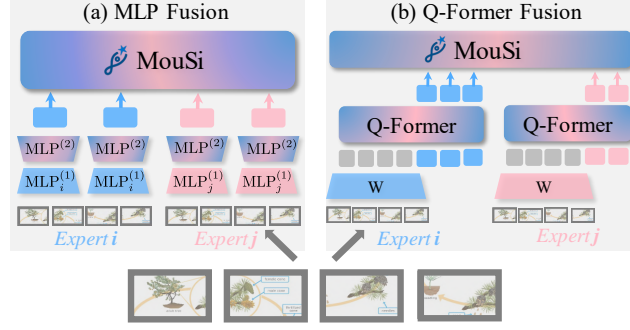


Figure 3: Examples of two types of multi-expert fusion networks. We show how the MLP method compresses visual information with “2-patches-1-token”, and how the Q-Former method compresses information with 3 trainable queries. The modules with color gradients represent the sharing of parameters among multiple experts to transfer knowledge.

limit during inference. Therefore, we propose **multi-patches-one-token** projection to proportionally reduce the number of tokens output by each expert. Since image signals have local or sparse properties, it is reasonable to use one token to represent neighboring patches. Take m -patch-one-token for example, we make the input dimension of the first layer of the network m times ($\text{MLP}^{(1)} : d_{in} \times m \rightarrow d_{hidden}$), and its hidden layer output vectors h_1^i, h_2^i, \dots are defined as follows:

$$h_1^i = \text{MLP}^{(1)} \left(\begin{bmatrix} v_1^i \\ v_2^i \\ \vdots \\ v_m^i \end{bmatrix} \right), \quad h_2^i = \text{MLP}^{(1)} \left(\begin{bmatrix} v_{m+1}^i \\ v_{m+2}^i \\ \vdots \\ v_{2m}^i \end{bmatrix} \right), \dots \quad (4)$$

where the $[\cdot]$ notation denotes concatenation over the vector dimension. The final number of vision tokens is reduced to $\frac{1}{m}$ of the original. In practice, m is typically set from 2 to 8, which reduces cost while usually not losing performance on downstream tasks. If m is set too large, the information of the image might be lost.

Q-Former network is a trainable Querying Transformer module and proposed to bridge the gap between a frozen image encoder and a pre-trained LLM. It extracts a fixed number of output features from the vision encoder, independent of input image resolution. We create a set number of learnable query embeddings as input to the Q-Former. The queries interact with each other through self-attention layers, and interact with frozen image features $e_i(I)$ through cross-attention layers. The output queries of the last layer are projected to the input layer of the LLM. We use the pre-trained parameters in BLIP-2 as initialization to accelerate convergence and, similar to the second layer MLP network, share the parameters among all experts. Since the dimension of query embeddings is equal to 768, we add an additional linear transformation ($W_i \in \mathbb{R}^{d_i \times 768}$) for each expert.

$$V_I = \text{Q-Former} (W_i (e_i(I)) \oplus W_j (e_j(I)) \oplus W_k (e_k(I))) \quad (5)$$

The ablation study in Section 3.2.1 shows that the MLP fusion network fuses better than the Q-Former despite having fewer parameters and not being pre-trained.

2.4 Different Positional Encoding Schemes

Although the m -patch-one-token operation or defining a small number of queries in the Q-Former has been able to reduce the proportion of vision tokens, the occupation of position embeddings by vision tokens should not be underestimated during inference. Inspired by the fact that visual experts already have positional encodings (e.g., 2D position encoding in ViT [29]), we believe it is redundant to again assign a VLM position embedding to each visual token individually. As shown in Figure 4, this report explores four positional encoding schemes for improving the assignment of position embeddings (PEs):

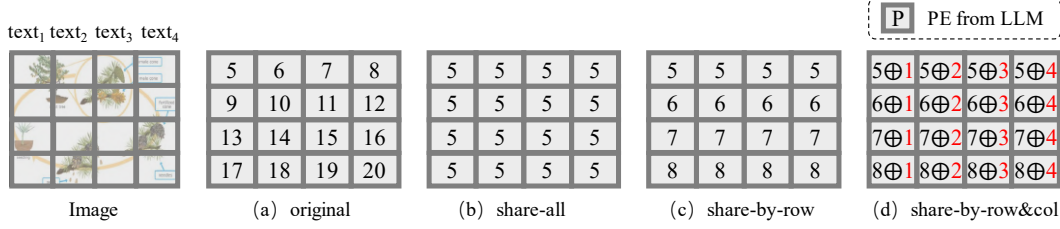


Figure 4: Diagram of the four positional encoding schemes. The \oplus operator indicates that the row position embedding and column position embedding are summed.

1. a separate position vector for each patch (*original*);
2. all vision tokens of an image share a PE (*share-all*);
3. one PE shared by the same row of vision tokens (*share-by-row*);
4. one PE shared by the same row of vision tokens, plus a set of learnable columns PEs (*share-by-row&col*).

Among the four methods, *share-all* can reduce the original $O(N^2)$ PE cost to $O(1)$, while the *share-by-row* and *share-by-row&col* can reduce the PE cost to $O(N)$. All of them can significantly alleviate the out-of-maximum-length problem, but the question is **how much do they affect the performance of VLM?** We report ablation results in Section 3.2.3.

3 Experiments

3.1 Main Results

The main focus of our experiments is to conduct explorations of single-expert, double-expert, and triple-expert ensembles. Following LLaVA-1.5 [22], our training pipeline consists of two phases. In phase 1, or the pre-training phase, we freeze the text-only LLM and the multi-expert encoder, and train the poly-visual fusion network from scratch to align the representation space of both. After training on a large-scale weakly-supervised (with noise) dataset, the text-only LLM is already capable of multimodal input and comprehension. In phase 2, or the fine-tuning phase, we unfreeze the LLM and further train it together with the poly-visual fusion network on diverse and high-quality supervised fine-tuning (SFT) datasets. The construct of the datasets and the training configuration for both stages are detailed as follows.

Datasets. During the pre-training phase, we utilized the LCS-558K dataset, which comprises $\sim 558K$ image-text pairs from the LAION-CC-SBU collection, annotated with BLIP-generated captions. During the fine-tuning phase, we mixed 10 diverse and high-quality SFT datasets containing VQA, OCR, region-level VQA, visual conversation, and language conversation data. To reduce training costs and enhance efficiency, we adopted the same preprocessing strategy as LLaVA-1.5, ultimately obtaining $\sim 665K$ SFT samples. All data splits are concatenated together and sampled with the same probability. We selected 9 of the 12 evaluation benchmarks for LLaVA-1.5 (excluding LLaVA-Bench that rely on unstable GPT4 responses, as well as VisWiz [30] and MME [31] for the website crashed), including VQA^{v2} [32]; GQA [33]; SQA¹: ScienceQA-IMG [34]; VQA^T: TextVQA [35]; POPE [20]; MMB & MMB^{CN}: MMBench & MMBench-Chinese *dev* results [36]; SEED¹: SEED-Bench-IMG [37]; MM-Vet [38]. Detailed statistical information can be found in Appendix A.

Hyperparameters. For main results, we keep all training hyperparameters roughly the same as the LLaVA series [7, 22]. We present a detailed description of the hyperparameters in Appendix B. For the MLP fusion network, we set m in m -patches-one-token from 1 to 16 to avoid exceeding the maximum length for training and inference. For the Q-Former fusion network, we set the number of queries per expert to match the number of outputs from the MLP fusion network. The parameters of the Q-Former fusion network are initialized using the pre-training parameters of BLIP-2 [26].

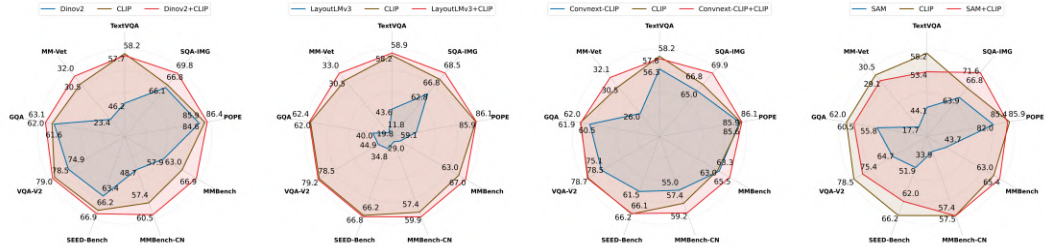
Model	Param.	VQA ^{v2}	GQA	SQA ^I	VQA ^T	POPE	MMB	MMB ^{CN}	SEED ^I	MM-Vet	Avg.
<i>Single Expert</i>											
CLIP	7.3B	78.5	62.0	66.8	58.2	85.9	63.0	57.4	66.2	30.5	63.2
DINOv2	8.1B	74.9	<u>61.7</u>	<u>66.1</u>	46.2	84.6	57.9	48.7	<u>63.4</u>	23.4	58.5
LayoutLMv3	7.4B	44.9	40.0	62.8	43.6	59.1	29.0	19.8	34.8	11.8	38.4
ConvNeXt	7.2B	<u>75.1</u>	60.5	65.0	<u>56.3</u>	<u>85.6</u>	<u>63.3</u>	<u>55.0</u>	61.5	<u>26.0</u>	<u>60.9</u>
SAM	7.6B	64.7	55.8	63.9	44.1	82.0	43.7	33.9	51.9	17.7	50.9
MAE	7.6B	62.0	53.2	63.3	44.5	79.7	41.6	33.0	49.4	16.5	49.2

Table 2: **Comparison of six vision experts on 9 benchmarks.** Param indicates the number of parameters.

3.1.1 Single Vision Expert

Table 2 compares the performance of all six VLMs with a single vision expert. The CLIP expert achieves the best performance in **all** 9 benchmarks, fully explaining why it has become the dominant choice of vision encoder for VLMs. Comparing the different attributes of the experts, CLIP ranked 5th in terms of the number of parameters, 3rd in terms of resolution, and 2nd above the size of the pre-training data, none of which had an absolute lead. Therefore, we guess that its main advantage lies in its image-text matching pre-training task, which has multimodal alignment capability in advance. Overall, the performance ranking of the six experts is roughly CLIP>ConvNeXt>DINOv2>SAM>MAE>LayoutLMv3. In addition, LayoutLMv3 is an undisputed expert in OCR and SAM in image segmentation but performs poorly as a single visual encoder in VLM. A natural question is *whether multi-expert fusion can activate their capabilities in their specialized fields?*

3.1.2 Double Vision Experts



Model	Param.	VQA ^{v2}	GQA	SQA ^I	VQA ^T	POPE	MMB	MMB ^{CN}	SEED ^I	MM-Vet	Avg.
<i>Double Experts</i>											
DINOv2+CLIP	8.4B	79.0	63.1	69.8	57.7	86.4	67.0	60.5	66.9	32.0	64.7
ΔDINOv2		4.1	1.5	3.7	11.5	1.8	9.1	11.8	3.5	8.6	
ΔCLIP		0.5	1.1	3.0	0.5	0.5	4.0	3.1	0.7	1.5	
LayoutLMv3+CLIP	7.7B	79.2	62.4	68.5	58.9	86.1	67.0	59.9	66.8	33.0	64.6
ΔLayoutLMv3		34.3	22.4	5.7	15.3	27.0	38.0	40.1	32.0	21.2	
ΔCLIP		0.7	0.4	1.7	0.7	0.2	4.0	2.5	0.6	2.5	
ConvNeXt+CLIP	7.5B	78.7	61.9	69.9	57.8	86.1	65.5	59.2	66.1	32.1	64.1
ΔConvNeXt		3.6	1.4	4.9	1.5	0.5	2.2	4.2	4.6	6.1	
ΔCLIP		0.2	0.1	3.1	0.4	0.2	2.5	1.8	0.1	1.6	
SAM+CLIP	7.9B	75.4	60.5	71.6	53.4	85.4	65.4	57.5	62.0	29.1	62.3
ΔSAM		10.7	4.7	7.7	9.3	3.4	21.7	23.6	10.1	11.4	
ΔCLIP		3.1	1.5	4.8	4.8	0.5	2.4	0.1	4.2	1.4	

Table 3: Performance comparison of different double-expert methods. The Δ-marked rows are compared to the single-expert method. Where **blue cells** indicate the **double-expert** model is better, and **red cells** indicate the **single-expert** model is better.

The current mainstream open-source VLMs have only one visual encoder, i.e., a single visual channel. However, multimodal tasks are diverse, and different tasks require different visual signals. In this subsection, we investigate whether dual-channel, i.e., double visual experts can outperform single experts on various tasks. We combine the strongest CLIP expert with other experts to construct a total of four double-expert combinations.

Table 3 shows the performance of the double-expert vision encoder on the nine benchmarks, and relative to each single expert belong them (a positive number indicates that the double expert outperforms the single expert). The results show that the “DINOv2+CLIP” experts, “LayoutLMv3+CLIP” experts, and “ConvNeXt+CLIP experts” three double-expert encoders outperform the arbitrary single encoder in almost all cases (23/27). The results indicate that two visual channels do outperform a single visual channel in terms of multimodal capabilities, demonstrating that multi-expert collaboration is feasible. For the “SAM+CLIP” combination, the results are surprising, with the dual expert outperforming the single expert in only 2/9 benchmarks, and lagging behind the single expert (specifically CLIP) in the remaining 7 benchmarks. The main reason might be that SAM encodes much more signals than CLIP (4096 patches vs. 576 patches), and fusion networks require a large information compression ratio. The most efficient CLIP channel is also compressed at this point, leading to performance decreases. There is a need to develop a more efficient visual information transfer network for experts with massive patches such as SAM.

Comparing the performance between double-expert methods, we found that the best double-expert is DINOv2+CLIP, rather than the ensemble of the best single expert and the second-best single expert, ConvNeXt+CLIP. It indicates that superior performance as a single expert does not necessarily imply optimality when ensembled. Since ConvNeXt and CLIP have considerable overlap in their training methods and training corpora, leading to the extraction of similar visual information, whereas the self-supervised DINOv2 and the weakly-supervised CLIP complement each other, resulting in a more effective ensemble. Furthermore, it is worth mentioning that LayoutLMv3, which performed the worst as a single expert, shows significant improvement when collaborating with CLIP, performing the best on four benchmarks and ranking overall just behind DINOv2+CLIP. Even SAM, whose information was compressed, achieved the highest performance on the ScienceQA-IMG benchmark. Therefore, we can conclude that when paired with the versatile visual expert CLIP, other experts can focus on capturing supplemental visual information to further enhance performance.

3.1.3 Triple Vision Experts

Model	Param.	VQA ^{v2}	GQA	SQA ^I	VQA ^T	POPE	MMB	MMB ^{CN}	SEED ^I	MM-Vet	Avg.
<i>Triple Experts</i>											
ConvNeXt+LayoutLMv3+CLIP	7.9B	78.5	63.3	70.2	58.0	87.3	66.8	58.9	66.0	32.2	64.6
ΔConvNeXt+CLIP		0.2	1.4	0.3	0.2	1.2	1.3	0.3	0.1	0.1	
ΔLayoutLMv3+CLIP		0.7	0.9	0.9	1.7	1.2	0.2	1.0	0.8	0.8	
ConvNeXt+DINOv2+CLIP	8.6B	78.6	63.2	69.2	57.8	86.5	66.6	58.9	67.1	32.9	64.5
ΔConvNeXt+CLIP		0.1	1.3	0.7	0.0	0.4	1.1	0.3	1.0	0.8	
ΔDINOv2+CLIP		0.4	0.1	0.6	0.1	0.1	0.4	1.6	0.2	0.9	
LayoutLMv3+DINOv2+CLIP	8.8B	79.1	63.6	69.0	58.4	86.5	67.4	60.0	67.5	33.6	65.0
ΔLayoutLMv3+CLIP		0.1	1.2	0.5	0.5	0.4	0.4	0.1	0.7	0.6	
ΔDINOv2+CLIP		0.1	0.5	0.8	0.7	0.1	0.4	0.5	0.6	1.6	

Table 4: Performance comparison of different triple-expert methods. The Δ-marked rows are compared to the double-expert method. Where **blue cells** indicate the **triple-expert** model is better, and **red cells** indicate the **double-expert** model is better.

Based on the double-expert encoder, we further construct the three-expert combinations. As shown in Table 4, the three-expert approach wins in 4/6 cases in comparison with the two-expert at the data size of LLaVA-1.5. The best-performing three-expert is LayoutLMv3+DINOv2+CLIP, followed by ConvNeXt+LayoutLMv3+CLIP, and finally ConvNeXt+DINOv2+CLIP. Among them, model LayoutLMv3+DINOv2+CLIP has the largest number of parameters, reaching 8.8 billion. We suspect that the main reason limiting the performance of the triple-expert methods is the insufficient amount of data. We train the MouSi on larger (1647K) augmented data and observe more significant performance gains in Section 3.4.

3.2 Ablation Study

3.2.1 Effect of Fusion Methods

Model	Param.	VQA ^{v2}	GQA	SQA ^I	VQA ^T	POPE	MMB	MMB ^{CN}	SEED ^I	MM-Vet
<i>Effect of Fusion Methods</i>										
DINOv2+CLIP+MLP	8.4B	79.0	63.1	69.8	57.7	86.4	67.0	60.5	66.9	32.0
DINOv2+CLIP+Q-Former	8.5B	60.4	50.9	66.7	45.1	45.2	52.7	44.8	51.8	20.5
ConvNeXt+CLIP+MLP	7.5B	78.7	61.9	69.9	57.8	86.1	65.5	59.2	66.1	32.1
ConvNeXt+CLIP+Q-Former	7.6B	65.8	52.6	68.7	45.6	77.0	59.7	49.8	53.2	22.1

Table 5: Performance comparison of different poly-expert fusion methods.

The MLP projection and Q-Former network are two mainstream methods for connecting vision and language. *Which of them can more effectively convey visual signals* is a key issue, especially in the context of multi-expert fusion. Table 5 presents the performance of using MLP and Q-Former respectively on three double-expert combinations, including “DINOv2 & CLIP” and “ConvNeXt & CLIP”. The results demonstrate that MLP significantly outperforms Q-Former in **all** cases, despite having fewer parameters and not utilizing pre-trained parameters like Q-Former, being instead directly initialized randomly. It suggests that we should prefer a straightforward connection in the LLaVA with poly-visual experts setup.

3.2.2 Effect of Expert Order

Model	Param.	VQA ^{v2}	GQA	SQA ^I	VQA ^T	POPE	MMB	MMB ^{CN}	SEED ^I	MM-Vet
<i>Effect of the Order of Experts</i>										
DINOv2→CLIP	8.4B	79.0	63.1	69.8	57.7	86.4	67.0	60.5	66.9	32.0
CLIP→DINOv2	8.4B	79.6	63.9	69.2	59.1	86.4	67.5	59.4	67.0	31.8
ConvNeXt→CLIP	7.5B	78.7	61.9	69.9	57.8	86.1	65.5	59.2	66.1	32.1
CLIP→ConvNeXt	7.5B	78.0	61.9	68.7	57.4	86.9	66.0	58.1	65.4	30.6

Table 6: Performance comparison of different expert orders. We exchange the order of experts in “DINOv2+CLIP”, and “ConvNext+CLIP”.

Due to the autoregressive and position-aware characteristics of LLMs, even if the visual experts are exactly the same, a different order alone could affect the final output. Table 6 presents the effect of swapping the order between double experts. The swap results of groups “DINOv2 & CLIP” and “ConvNeXt & CLIP” indicate that order can cause some fluctuations in performance, with gains (7 of 22) on some benchmarks and losses (15 of 22) on others. In general, placing CLIP later brings about better overall performance. Because CLIP is the most effective single expert and the expert placed later is closer to the generation, we speculate that the latter-positioned expert has a slightly greater effect on the response. This phenomenon is also consistent with the characteristics of binocular vision organisms, such as humans, where one eye is the dominant eye and the other is the non-dominant eye. The brain typically favors the input from the dominant eye when processing visual information [39].

3.2.3 Effect of Different Positional Encoding Schemes

Model	VQA ^{v2}	GQA	SQA ^I	VQA ^T	POPE	MMB	MMB ^{CN}	SEED ^I	MM-Vet	Avg.
<i>Different Positional Encoding Schemes</i>										
Origin	78.5	62.0	66.8	<u>58.2</u>	85.9	64.3	<u>58.3</u>	<u>66.2</u>	30.5	63.4
Share-all	<u>79.0</u>	<u>62.4</u>	68.4	58.4	<u>86.3</u>	67.4	58.2	65.7	<u>31.7</u>	64.2
Share-by-row	75.0	57.2	63.4	51.7	86.1	46.4	43.4	55.6	31.9	56.7
Share-by-row&col	79.0	62.6	<u>68.3</u>	58.1	86.3	<u>66.0</u>	58.8	66.2	30.6	<u>64.0</u>

Table 7: Comparison of four positional encoding schemes on 9 benchmarks.

Benchmark	Text Prompt	LayoutLMv3	DINOv2	CLIP
MMB	61.1%	0.14%	2.76%	11.1%
MMB ^{CN}	58.8%	0.16%	2.92%	10.7%

Table 8: Average attention probability (%) allocation of Mousi’s output on each visual expert. The model used is a “LayoutLMv3+DINOv2+CLIP” triple-expert visual encoder.

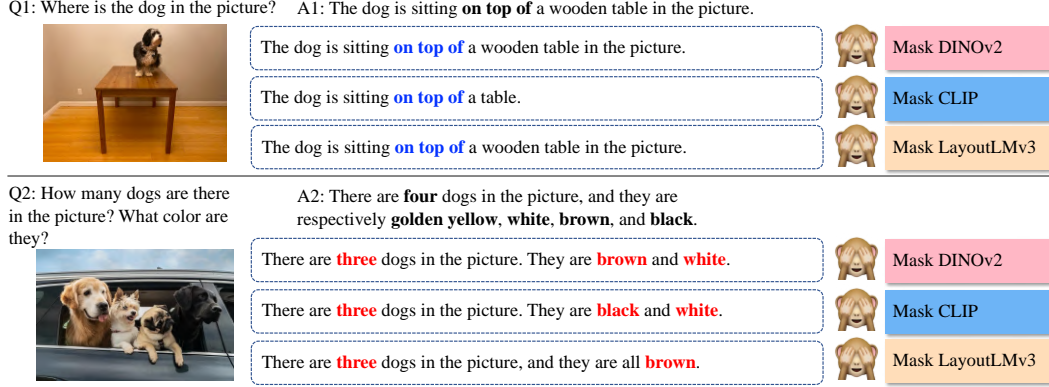


Figure 5: The perturbation experiments on the triple-expert LayoutLMv3+DINOv2+CLIP model, the specific perturbation is to mask all the output of the corresponding vision expert.

This subsection compares the four positional encoding schemes of VLMs introduced in Section 2.4. Table 7 shows the results of the four approaches, where share-all not only saves the most PE but also improves the average performance by 0.8 on top of CLIP. The 2D positional coding (share-by-row&col) also improves the average performance by 0.6. However, share-by-row impairs the performance of the model, probably because row sharing corrupts the position information of the visual coder itself. The experimental results validate our conjecture that it is redundant to re-assign LLM positional encoding to each vision token that already has positional information.

3.3 Analysis

Among multiple visual encoders, one question worthy of analysis is the contribution of different experts to the model’s output. Attention mechanisms are commonly used interpretive tools in Transformer networks. Here, we take a three-expert encoder as an example and analyze the average contribution of each expert across two multilingual benchmarks, MMB-English and MMB-Chinese. The contribution of one sample is the output token’s average attention to each expert’s representation. Averaging over the entire dataset yields the overall average contribution.

Table 8 shows the individual contributions of the text prompt, LayoutLMv3, DINOv2, and CLIP to the output. The results indicate that the contribution of the text prompt to the answer is significantly higher than that of the visual experts. This is as expected. Firstly, the text prompt defines the format of the VLM’s response, necessitating attention to the prompt during output, and secondly, the text has a higher information density than images, hence the average attention is usually higher for text. Comparing the three visual experts, we find that their contributions in descending order are CLIP, DINOv2, and LayoutLMv3. CLIP still demonstrates the characteristics of being the dominant eye or the primary visual channel. DINOv2’s contribution is approximately 20% of CLIP’s, while LayoutLM’s contribution is minimal, at only 1% of CLIP’s.

A natural question that follows is, given the existence of visual channels with very low contributions, is there a necessity for them to be part of the model? Figure 5 shows our perturbation experiments on the triple-expert LayoutLMv3+DINOv2+CLIP model. The output tokens of the corresponding expert are fully masked when generating answers, thus exploring the effect of the current expert on the output. In Case 1, the user asks MouSi a simple question: “Where is the dog in the picture?”. No matter which visual expert’s output signal is masked, the remaining two visual channels are sufficient to correctly answer the location question “on top of”. More details are provided when CLIP experts

are present, such as outputting “wooden table” instead of just “table”. In Case 2, the user asks MouSi “How many dogs are there in the picture? What colors are they?” The perturbation results show that only three experts working together can answer the question correctly. The absence of any one expert results in an incorrect answer, which demonstrates the difference in the information captured by the multiple visual channels of the poly-visual-expert VLM. Some multimodal tasks rely on the synergy of multiple channels, which a single channel (i.e., a single expert VLM) does not have.

3.4 Data Enhancement

Model	Param.	VQA ^{v2}	GQA	SQA ^I	VQA ^T	POPE	MMB	MMB ^{CN}	SEED ^I	MM-Vet	Avg.
<i>Baselines</i>											
InstructBLIP[40]	8.0B	–	49.2	60.5	50.1	–	36.0	23.7	53.4	26.2	
Qwen-VL-Chat[24]	9.6B	78.2	57.5	68.2	61.5	–	60.6	56.7	58.2	–	
BLIP-2[26]	14.1B	41.0	41.0	61.0	42.5	85.3	–	–	46.4	22.4	
Shikra[41]	7.3B	77.4	–	–	–	–	58.8	–	–	–	
PandaGPT[42]	13B	–	–	–	–	–	45.4	32.0	47.6	19.6	
mPLUG-Owl2[43]	8.2B	–	–	–	–	–	66.5	59.5	64.5	<u>35.7</u>	
Emu2-chat[44]	37B	–	–	–	–	–	62.4	44.2	68.9	31.0	
<i>Default Data</i>											
CLIP (LLaVA-1.5[22])	7.3B	78.5	62.0	66.8	58.2	85.9	64.3	58.3	66.2	30.5	63.1
ConvNeXt+LayoutLMv3+CLIP	7.9B	78.5	63.3	70.2	58.0	87.3	66.8	58.9	66.0	32.2	64.6
<i>Data Enhancement</i>											
CLIP	7.3B	<u>80.8</u>	62.7	<u>81.9</u>	60.7	85.5	69.2	<u>61.7</u>	<u>69.8</u>	35.6	<u>67.5</u>
LayoutLMv3+ConvNeXt+CLIP	7.9B	80.9	<u>62.6</u>	84.3	<u>61.3</u>	<u>86.3</u>	<u>68.8</u>	63.7	70.1	38.4	68.5

Table 9: The effect of data enhancement on nine benchmarks. **Param.** indicates the number of parameters.

After comprehensively exploring the architecture and effectiveness of the poly-visual-expert VLM, we further augmented the data from LLaVA-1.5 to explore the upper limits of the performance of the poly-visual-expert VLM.

Setting During the pre-training phase, we used 1.2 million pre-training data to replace the original 558K data in LLaVA-1.5. Where 100K data were generated by GPT4v, and the remaining data were produced by a supervised trained image captioner, which included the 558K images but with higher quality captions. During the SFT phase, we expanded the 665K SFT data to 1647K. Detailed statistical information can be found in Appendix A. For data enhancement results, we keep all training hyperparameters roughly the same as the main results. Besides the number of iterations varies with the increase of data size.

Table 9 reports the results for LLaVA-1.5 (i.e., single CLIP expert), LLaVA-1.5 after data enhancement, and MouSi (with triple-expert “LayoutLM+ConvNeXt+CLIP”) after data enhancement on nine benchmarks. The results show that with data enhancement, the poly-visual expert VLM can further improve performance (7/9) compared with single-expert VLM. The average performance improved by 1.0, yet the number of parameters increased by only 300M. Comparing the effects of data augmentation, we observe that the single-expert approach improved by 4.4, and the triple-expert method improved by 3.9. This confirms that the potential of poly-visual-expert VLMs has not yet been fully tapped and that more data can significantly enhance the capabilities of VLMs. Finally, compared to mainstream VLMs, MouSi performs the best in 8 out of 9 benchmarks while exhibiting the second-best performance in the remaining one, demonstrating strong multimodal assistant capabilities.

4 Case Study

Figure 6 shows the case study of MouSi on seven tasks, including Complex Image Captioning, Visual Text Generating, OCR Interpreting Reasoning with World Knowledge, Visual Math Problem Solving, Complex Counting, and Visual Grounding. MouSi is able to successfully follow a variety of multimodal instructions, allowing for flexible interaction with humans.

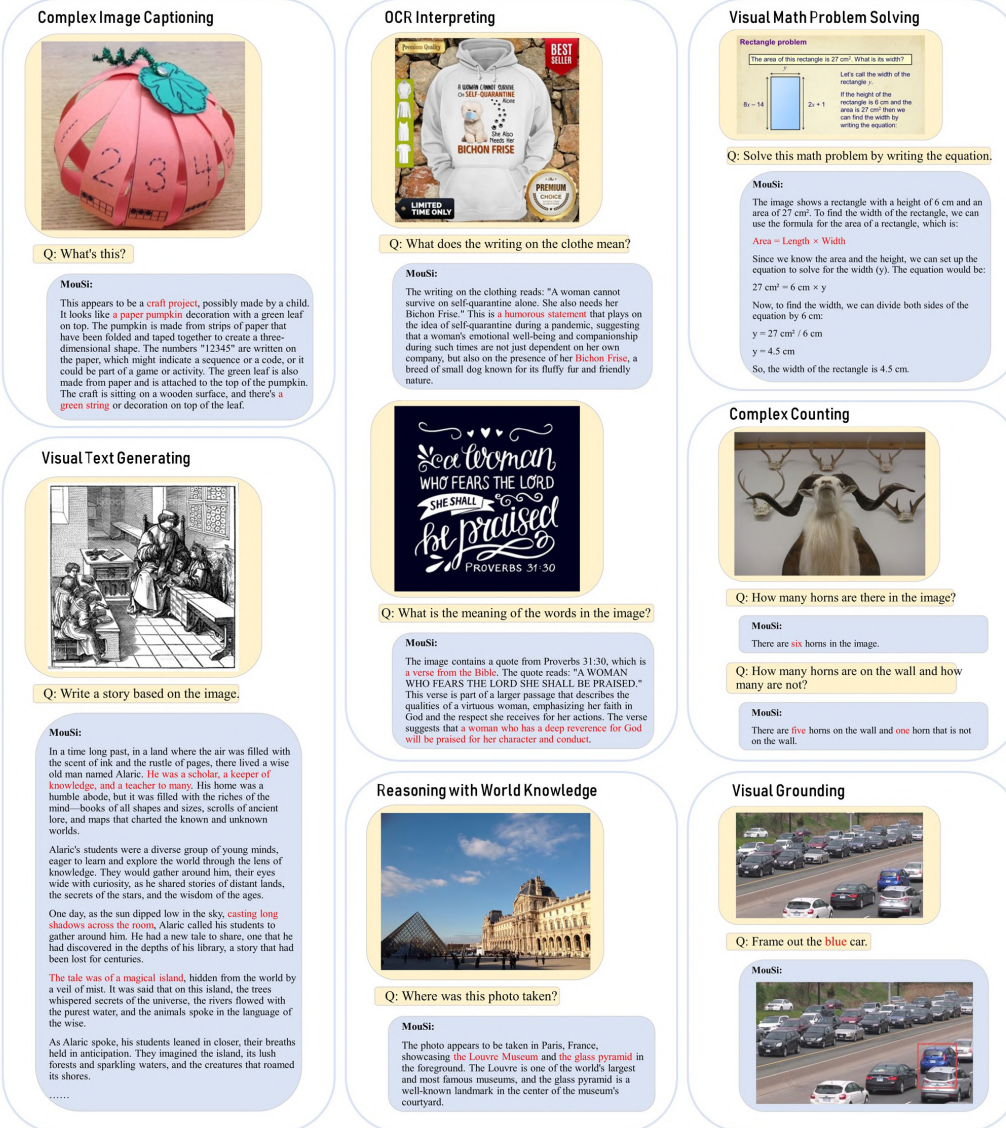


Figure 6: Qualitative examples generated by MouSi.

5 Related Work

Vision-Language Models (VLMs) represent the confluence of linguistic and visual processing, and they have shown promising results in various applications. Early models such as VisualGPT [45] provided foundational work in image captioning, while the BLIP series [28, 26] extended capabilities to include visual question answering. Flamingo [46] and Kosmos-1 [47] demonstrated effective multi-modal understanding within image-text frameworks. LLaMA adaptations like LLaVA [7] and MiniGPT-4 [48] utilize projection layers for connecting vision encoders and LLMs. CoGVLM [23] replicated close to double the parameters to build visual experts specializing in visual tokens, while similar to our exploration of positional encoding, they used share-by-one rather than the original approach. Qwen-VL and BLIP series [24, 25] use the Q-Former network to bridge text and image.

Visual Encoding Experts The success of vision language models pivots upon adept visual encoding; hence, a curated selection of vision encoders, each with its own domain expertise, is crucial for holistic visual understanding. The CLIP model by [6] employs contrastive learning to align images and text, effectively facilitating semantic image understanding. Dinov2 [13] from Meta

advances self-supervised learning through a student-teacher network paradigm, developing spatial understanding with a robust ViT framework. Microsoft’s LayoutLMv3 [14], on the other hand, presents a multimodal Transformer adept in document AI by bolstering word-patch alignment in a ViT model. Convnext [21] reintroduces the efficacy of ConvNets with its FCMAE framework and GRN layer, finetuned with ImageNet-22K data. The Segment Anything Model (SAM) by [27] showcases exceptional segmentation prowess, trained on a vast dataset to champion zero-shot generalization in its ViT infrastructure. The MAE [17] demonstrated remarkable denoising self-supervised capabilities, reconstructing images with a high degree of fidelity. Yet these encoders, notably CLIP, possess limitations as evidenced by [19] highlighted its struggles with spatial orientation and [20]’s findings on object hallucination. Moreover, [15] recognized a division of competencies, noting more semantics in fully/weakly supervised encoders like CLIP, while others excel in fine-grained perception.

Multi-Modal Large Language Models (MLLMs) have been evolving rapidly, with models like ImageBind-LLM [49] and PandaGPT [42] incorporating richer modality inputs, including audio and video. There is also a growing focus on region-level parsing [41], text-to-image generation [50], and 3D understanding [51]. These models show that MLLMs can achieve meaningful performance across a range of tasks. MouSi, as a poly-visual-expert VLM, is easily adapted to multi-modal-expert models, which will be our future work.

6 Conclusion

In this paper, we push the boundaries of vision-language models (VLMs) by proposing a novel polyvisual system that closely mirrors the complex and multi-dimensional nature of biological visual processing. Leveraging the unique attributes of diverse visual encoders, our system unifies their strengths to enrich the multimodal understanding of VLMs. Furthermore, we address the challenge of efficiently integrating visual information into language models by introducing techniques such as multi-patch-single-token projection and optimizing positional embeddings. This not only allows us to manage the overflow of vision tokens that typically burdens VLMs but also retains the models’ semantic and spatial reasoning capabilities. Through rigorous experiments across a suite of benchmarks, we demonstrate that our polyvisual approach significantly enhances the VLMs’ performance, outpacing existing models in accuracy and depth of understanding. These results support our hypothesis that a well-integrated assembly of expert encoders can lead to a substantial improvement in handling complex multimodal inputs.

References

- [1] Agrawal, H., K. Desai, Y. Wang, et al. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957. 2019.
- [2] Antol, S., A. Agrawal, J. Lu, et al. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433. 2015.
- [3] Yu, L., P. Poirson, S. Yang, et al. Modeling context in referring expressions. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016.
- [4] Durante, Z., Q. Huang, N. Wake, et al. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568*, 2024.
- [5] Xi, Z., W. Chen, X. Guo, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.
- [6] Radford, A., J. W. Kim, C. Hallacy, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [7] Liu, H., C. Li, Q. Wu, et al. Visual instruction tuning. In *NeurIPS*. 2023.
- [8] Yamada, Y., Y. Tang, I. Yildirim. When are lemons purple? the concept association bias of clip. *arXiv preprint arXiv:2212.12043*, 2022.

- [9] Thrush, T., R. Jiang, M. Bartolo, et al. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248. 2022.
- [10] Yuksekgonul, M., F. Bianchi, P. Kalluri, et al. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*. 2022.
- [11] Baden, T., P. Berens, K. Franke, et al. The functional diversity of retinal ganglion cells in the mouse. *Nature*, 529(7586):345–350, 2016.
- [12] Chen, F.-L., D.-Z. Zhang, M.-L. Han, et al. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023.
- [13] Oquab, M., T. Darcet, T. Moutakanni, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [14] Huang, Y., T. Lv, L. Cui, et al. Layoutlmv3: Pre-training for document ai with unified text and image masking, 2022.
- [15] Wang, G., Y. Ge, X. Ding, et al. What makes for good visual tokenizers for large language models? *arXiv preprint arXiv:2305.12223*, 2023.
- [16] Touvron, H., M. Cord, M. Douze, et al. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.
- [17] He, K., X. Chen, S. Xie, et al. Masked autoencoders are scalable vision learners, 2021.
- [18] Caron, M., H. Touvron, I. Misra, et al. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660. 2021.
- [19] Kamath, A., J. Hessel, K.-W. Chang. What’s “up” with vision-language models? investigating their struggle with spatial reasoning. *arXiv preprint arXiv:2310.19785*, 2023.
- [20] Li, Y., Y. Du, K. Zhou, et al. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.
- [21] Woo, S., S. Debnath, R. Hu, et al. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142. 2023.
- [22] Liu, H., C. Li, Y. Li, et al. Improved baselines with visual instruction tuning, 2023.
- [23] Wang, W., Q. Lv, W. Yu, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.
- [24] Bai, J., S. Bai, S. Yang, et al. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.
- [25] Dai, W., J. Li, D. Li, et al. Instructblip: Towards general-purpose vision-language models with instruction tuning. arxiv 2023. *arXiv preprint arXiv:2305.06500*.
- [26] Li, J., D. Li, S. Savarese, et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.
- [27] Kirillov, A., E. Mintun, N. Ravi, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [28] Li, J., D. Li, C. Xiong, et al. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

- [29] Wang, Z., J.-C. Liu. Translating math formula images to latex sequences using deep neural networks with sequence-level training, 2019.
- [30] Gurari, D., Q. Li, A. J. Stangl, et al. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617. 2018.
- [31] Fu, C., P. Chen, Y. Shen, et al. A comprehensive evaluation benchmark for multimodal large language models. *CoRR*, abs/2306.13394, 2023.
- [32] Goyal, Y., T. Khot, D. Summers-Stay, et al. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913. 2017.
- [33] Hudson, D. A., C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709. 2019.
- [34] Lu, P., S. Mishra, T. Xia, et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.
- [35] Singh, A., V. Natarajan, M. Shah, et al. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326. 2019.
- [36] Liu, Y., H. Duan, Y. Zhang, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.
- [37] Li, B., R. Wang, G. Wang, et al. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.
- [38] Yu, W., Z. Yang, L. Li, et al. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.
- [39] Miller, K. D., J. B. Keller, M. P. Stryker. Ocular dominance column development: analysis and simulation. *Science*, 245(4918):605–615, 1989.
- [40] Dai, W., J. Li, D. Li, et al. Instructblip: Towards general-purpose vision-language models with instruction tuning. arxiv 2023. *arXiv preprint arXiv:2305.06500*, 2023.
- [41] Chen, K., Z. Zhang, W. Zeng, et al. Shikra: Unleashing multimodal llm’s referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.
- [42] Su, Y., T. Lan, H. Li, et al. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*, 2023.
- [43] Ye, Q., H. Xu, J. Ye, et al. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration, 2023.
- [44] Sun, Q., Y. Cui, X. Zhang, et al. Generative multimodal models are in-context learners, 2023.
- [45] Chen, J., H. Guo, K. Yi, et al. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18030–18040. 2022.
- [46] Alayrac, J.-B., J. Donahue, P. Luc, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.
- [47] Huang, S., L. Dong, W. Wang, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.
- [48] Zhu, D., J. Chen, X. Shen, et al. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

- [49] Han, J., R. Zhang, W. Shao, et al. Imagebind-llm: Multi-modality instruction tuning. *arXiv preprint arXiv:2309.03905*, 2023.
- [50] Wen, S., G. Fang, R. Zhang, et al. Improving compositional text-to-image generation with large vision-language models. *arXiv preprint arXiv:2310.06311*, 2023.
- [51] Xu, R., X. Wang, T. Wang, et al. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023.
- [52] Chen, L., J. Li, X. Dong, et al. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023.
- [53] Wang, J., L. Meng, Z. Weng, et al. To see is to believe: Prompting gpt-4v for better visual instruction tuning. *arXiv preprint arXiv:2311.07574*, 2023.

A Datasets

During the pretrain phase, we employed the identical LCS-558K dataset as utilized in LLaVA-1.5, sourced from LAION-CC-SBU. For data-enhanced datasets, we incorporated the pre-trained dataset from ShareGPT4V [52], distinguished by its longer textual descriptions.

In the subsequent finetune phase, we utilized the same instruction-based fine-tuning data as LLaVA-1.5 for the default dataset, comprising approximately 665K samples. For datasets with enhanced data, we introduced supplementary data during the finetune stage, drawing from sources such as ShareGPT4V, LVIS-INSTRUCT4V [53], and CogVLM-SFT-311K-CN [23].

The specifics of our pretrain and finetune datasets are detailed in Table 10.

Default pretrain data	Size	Enhanced pretrain data	Size
LCS-558K	558K	ShareGPT4V	1200K
Default finetune data	Size	Enhanced finetune data	Size
LLaVA	158K	ShareGPT4V-cap100k	100K
ShareGPT	40K	ShareGPT4V-mix-665k	665K
VQAv2	83K	LVIS-INSTRUCT4V-220k	220K
GQA	72K	CogVLM-SFT-311K-CN	150K
OCRVQA	80K	VG	86K
A-OKVQA	50K	OCRVQA	80K
TextCaps	22K	GQA	72K
RefCOCO	30K	VQAv2	60K
VG	86K	docvqa	44K
OKVQA	9K	stvqa	30K
		fmqa	23K
		textvqa	21K
		coco-cn	20K
		ScienceQA	10K
		flickr8k-cn	8K
		chinese-food	1K
Total	665K	Total	1647K

Table 10: Default data and Enhanced data for the Pretrain and Finetune phases of our model.

B Hyperparameters


We use the same set of hyperparameters as the original LLaVA-1.5. The training hyperparameters for visual language alignment pre-training and visual instruction tuning are shown in Table 11.

Hyperparameter	Pretrain	Finetune
batch size	256	128
lr	1e-3	2e-5
lr schedule	cosine decay	
lr warmup ratio	0.03	
weight decay	0	
epoch	1	
optimizer	AdamW	
DeepSpeed stage	2	3

Table 11: Hyperparameters of our model’s pretrain and finetune.

C More Case Studies

2024/1/21 下午2:27:08



这是哪个城市拍摄的?
Which city was this shot in?

2024/1/21 下午2:27:29

这是在纽约市拍摄的。
This was shot in New York City.

2024/1/21 下午2:27:40

为什么说是在纽约拍摄的
Why was it shot in New York

2024/1/21 下午2:27:53

因为纽约市是一个著名的城市，其标志性建筑物 Empire State Building 在夜晚也是灯火辉煌的景点，因此可以确定这是在纽约市拍摄的。
Since New York City is a famous city and its iconic Empire state Building is also a brightly lit attraction at night, it is certain that this photo was taken in New York City.

2024/1/21 下午2:28:31

框出 Empire State Building (帝国大厦)
Frame the Empire State Building

2024/1/21 下午2:28:38




Figure 7: Qualitative Chinese examples generated by Mousi.

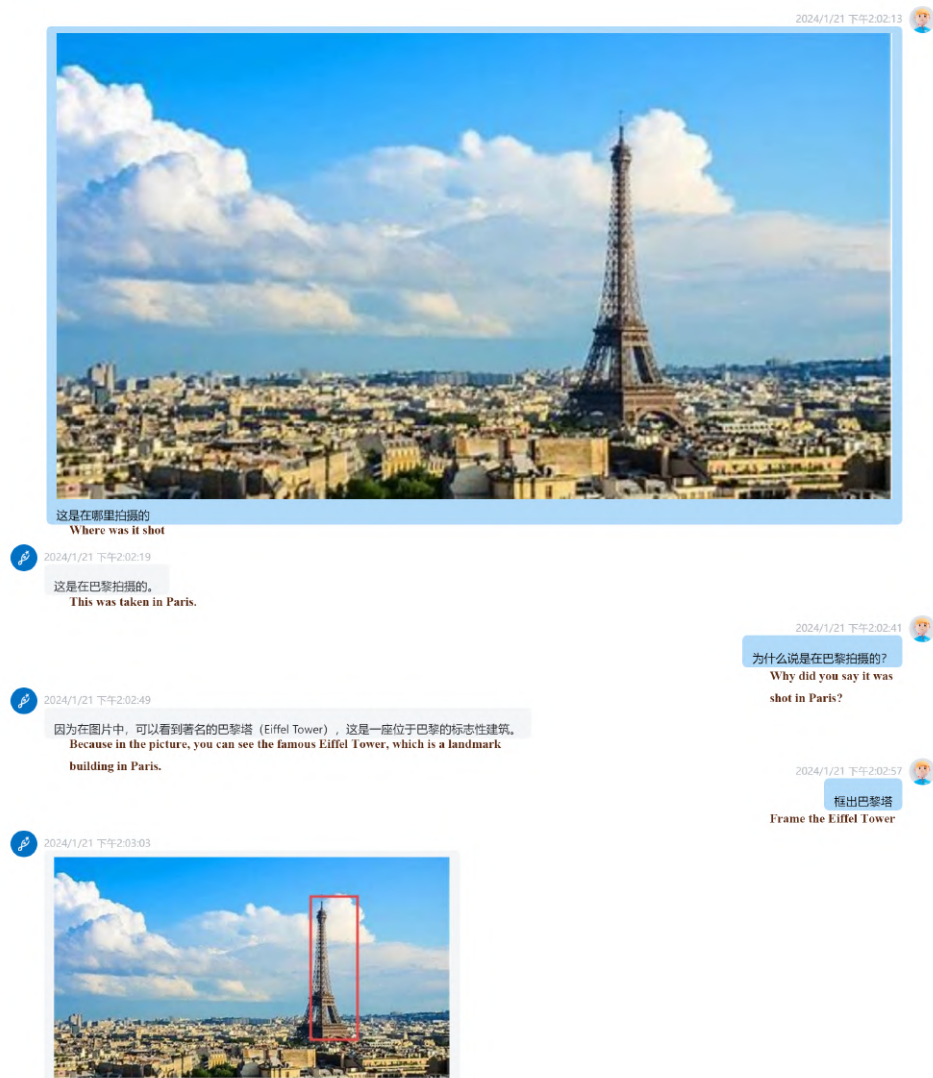


Figure 8: Qualitative Chinese examples generated by Mousi.

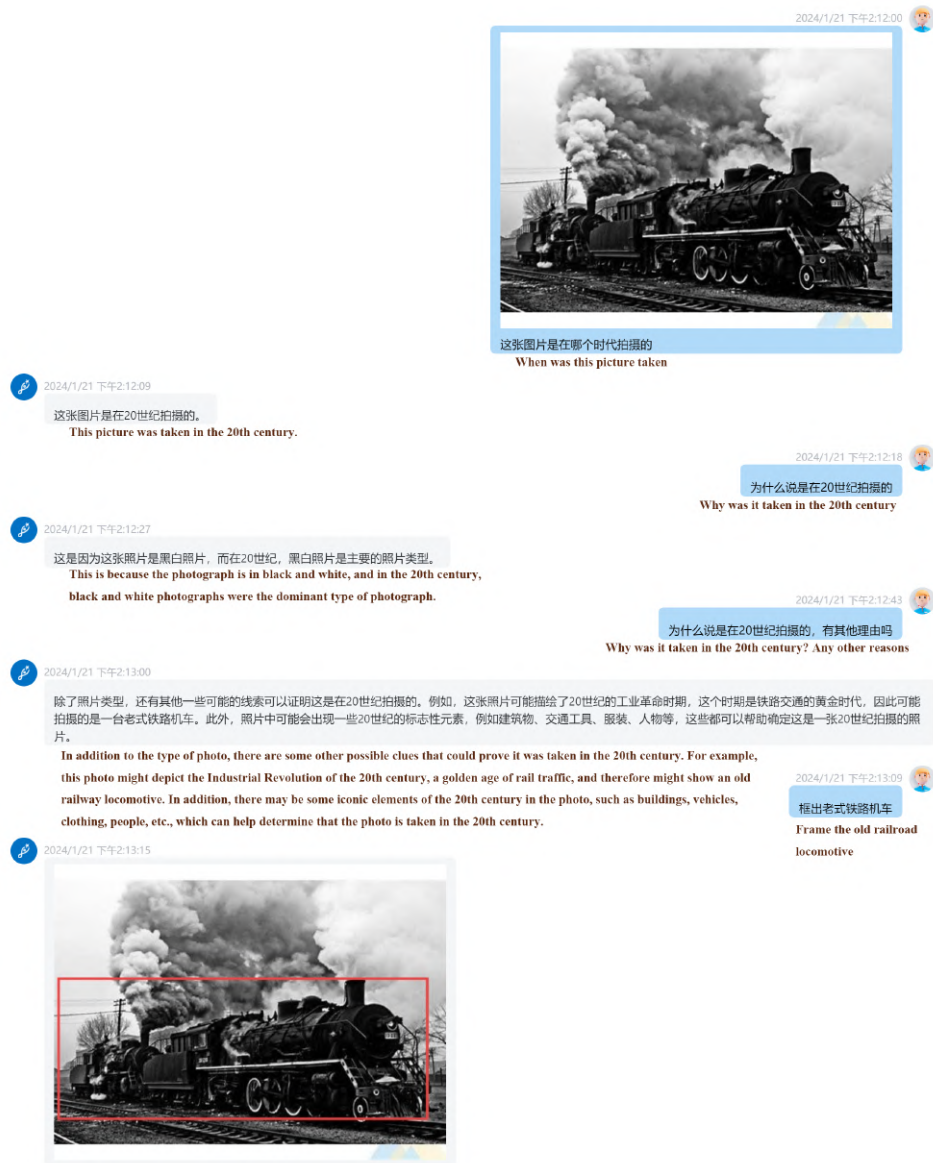


Figure 9: Qualitative Chinese examples generated by Mousi.



这幅画展示了什么历史时期

What period of history does the picture show



2024/1/21 下午2:51:39

这幅画展示了中世纪时期的战争场景。

The painting shows a battle scene from the Middle Ages.

2024/1/21 下午2:51:28



为什么说这是中世纪时期，从哪里看出

Why the Middle Ages? Where do you get that

2024/1/21 下午2:51:57



2024/1/21 下午2:52:11

从画作中可以看出，这是一个中世纪时期的战争场景，因为人们在画作中穿着中世纪时期的服装，使用中世纪时期的武器，并且战斗场景也符合中世纪时期的特点。

From the painting, it can be seen that this is a medieval battle scene, because the people in the painting are wearing medieval costumes, using medieval weapons, and the battle scene is also in line with the

2024/1/21 下午2:52:35



图中中世纪时期的武器有哪些

What are the medieval weapons in the picture



2024/1/21 下午2:52:45

在这幅画中，中世纪时期的武器包括长矛、弓箭、剑、铁甲等。

In this painting, the weapons of the medieval period include spears, bows and arrows, swords, and iron armor.

2024/1/21 下午2:52:58



框出长矛

Frame the spear



2024/1/21 下午2:53:05



Figure 10: Qualitative Chinese examples generated by Mousi.

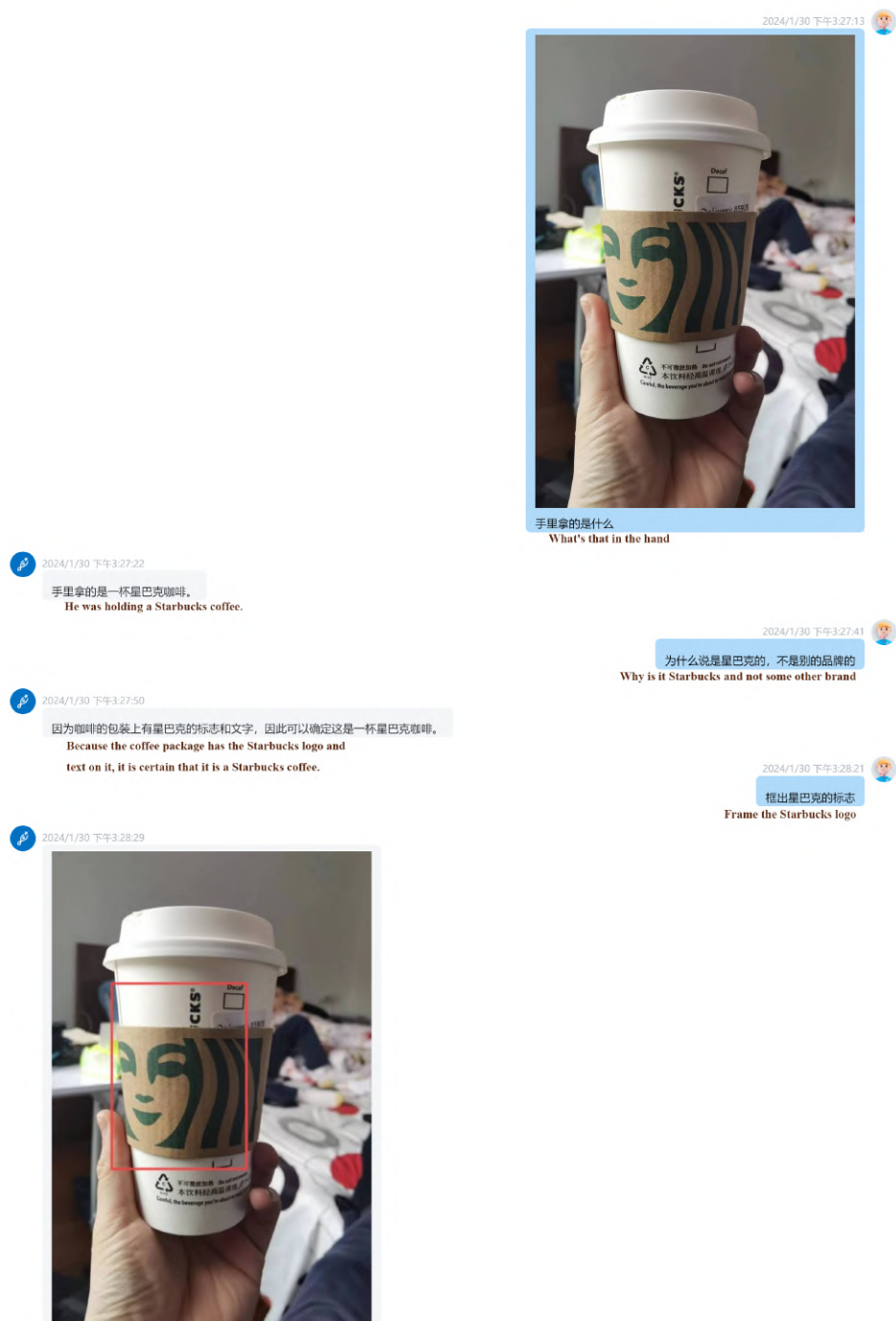


Figure 11: Qualitative Chinese examples generated by Mousi.



Figure 12: Qualitative Chinese examples generated by Mousi.