
MoME: Mixture of Multimodal Experts for Generalist Multimodal Large Language Models

Leyang Shen* Gongwei Chen* Rui Shao[†] Weili Guan Liqiang Nie[†]
School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen
{chengongwei, shaorui, guanweili, nieliqiang}@hit.edu.cn

Abstract

Multimodal large language models (MLLMs) have demonstrated impressive capabilities across various vision-language tasks. However, a generalist MLLM typically underperforms compared with a specialist MLLM on most VL tasks, which can be attributed to task interference. In this paper, we propose a mixture of multimodal experts (MoME) to mitigate task interference and obtain a generalist MLLM. Our MoME is composed of two key components, a mixture of vision experts (MoVE) and a mixture of language experts (MoLE). MoVE can adaptively modulate the features transformed from various vision encoders, and has a strong compatibility in transformation architecture. MoLE incorporates sparsely gated experts into LLMs to achieve painless improvements with roughly unchanged inference costs. In response to task interference, our MoME specializes in both vision and language modality to adapt to task discrepancies. Extensive experiments show that MoME significantly improves the performance of generalist MLLMs across various VL tasks.

1 Introduction

Recently, Multimodal Large Language Models (MLLMs) [39, 35, 48, 31, 67] have witnessed remarkable progress. With the help of Large Language Models (LLMs) [13, 63, 1] and Modality Encoders [50, 30, 17, 52, 70], MLLMs demonstrate excellent multimodal comprehensive abilities, especially in solving a wide range of vision-language (VL) tasks [3, 42, 37, 55, 57, 56], such as Image Caption, Visual Question Answering, Referring Expression Comprehension, and Optical Character Recognition. However, it is increasingly acknowledged that a generalist MLLM tends to have lower performance compared to a specialist MLLM on most VL tasks [12, 8], as depicted in Fig. 1 (a). This phenomenon can be attributed to task interference, a fundamental and crucial issue in multi-task learning [16, 41, 64].

There are some preliminary attempts [15, 6, 12, 5, 8, 58] to address this issue in instruction-tuning MLLMs. The most promising direction [12, 5, 8, 58] among these attempts is to use a mixture of experts (MoE) in MLLMs, aiming for each expert to specialize in several tasks. However, these works only investigate MoE in LLMs and primarily concentrate on textual differences between tasks, overlooking the equally important visual information. As shown in Fig. 1 (b-c), we analyze the distribution of various VL tasks across both vision and language modalities. It is evident that images from different groups of VL tasks exhibit distinct feature distributions, as do texts. Inspired by this observation, we argue that handling task interference needs to comprehensively exploit task differences in both vision and language modalities.

*Equal contribution

[†]Corresponding authors

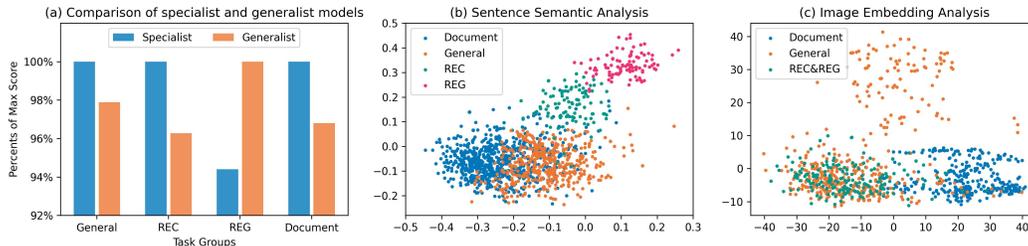


Figure 1: **VL data distribution visualization and model performance comparisons.** Experimental results in (a) show that a generalist model trained on a mixed dataset underperforms most specialist models trained on separate task groups. The feature distributions visualized in (b) and (c) show significant discrepancies across VL tasks in both images and instructions.

To mitigate task interference, we devise a Mixture of Multimodal Experts (MoME) and integrate it into MLLMs. Our MoME consists of a Mixture of Vision Experts (MoVE) for adaptively aggregating features from various vision encoders [52, 17, 50], and a Mixture of Language Experts (MoLE) for leveraging multiple sparsely-activated parameter-efficient adapters. To avoid feature mismatch in different vision encoders, we propose an adaptive deformable transformation (ADT) module in MoVE and use it to transfer features of vision encoders into a unified-length sequence of feature vectors. Our ADT module combines adaptive average pooling and deformable attention [71] to obtain compressed and self-enhanced visual features. After feature transformation, our MoVE uses an instance-level soft router to modulate and aggregate transformed visual features according to the instructions. Our MoLE introduces several parameter-efficient adapters [9] as experts and integrates them by using an instance-level sparsely-activated router. Due to the utilization of adapters, MoLE can be integrated into each feed-forward network layer of an LLM and only incurs a few computational costs with consistent performance gains.

To comprehensively evaluate the multimodal understanding ability of MoME, we collect an amount of VL tasks to form an instruction-tuning dataset and split them into four groups. Extensive experiments show that both MoVE and MoLE can consistently improve performance across all groups of tasks. Notably, MoVE can achieve an average performance gain of 12.87 points across all VL tasks, and improve by over 20 points on the "Document" group. Furthermore, we visualize the expert load distributions of MoVE and MoLE across various tasks. The experts in both MoVE and MoLE exhibit a relatively clear specialization in different groups of VL tasks. For example, the "Document" group of tasks has a strong preference for the "Pix2Struct" vision expert. The expert specialization is strong evidence to demonstrate that our MoME dynamically selects experts to adapt to task differences and mitigate task interference.

Our main contributions are summarized as follows:

- We propose MoME by simultaneously designing mixtures of experts tailored for both the vision encoder and the LLM, resulting in generalist MLLMs with the ability to combat task interference.
- Through statistical analysis, we demonstrate that our MoME specializes in both vision and language modality, effectively adapting to the varying requirements of different VL tasks.
- Extensive experiments show that our MoME possesses excellent multimodal understanding abilities and achieves superior performances across various groups of VL tasks.

2 Related Work

2.1 Vision Encoders in MLLMs

Vision encoders play important roles in the perception ability of recent MLLMs by encoding visual information into visual tokens, enabling LLMs to understand information on visual modalities. Most Multimodal Large Language Models (MLLM) use CLIP-ViT [52] as their vision encoder to provide the basic image-level perception of an image for LLMs, which is useful for tasks such as image caption and general VQA. However, Tong et al. [62] have found that CLIP-ViT struggles to encode some visual patterns, severely limiting perception and preventing MLLMs from becoming generalist.

To alleviate this, recent works [33, 62, 27] integrated different vision encoders [52, 17, 50, 70, 29] into a single MLLM, which enhanced the perception of MLLM. However, none have effectively mitigated the interference among different visual features, resulting in sub-optimal utilization of each encoder. Differently, we explore the task differences and propose MoVE to perform self-enhanced transformation and adaptive routing among features from different encoders, achieving consistently high performances across vast VL tasks.

2.2 Mixture of Experts in Large Models

Mixture of Experts (MoE) [24] is a type of structure with multiple expert networks working together, where each expert is responsible for a part of the knowledge space. By only activating specific experts adaptively during inference, MoE can reduce resource consumption and enhance reasoning speed, which is useful for LLM. Existing MoE LLM [18, 26, 14] usually replace the feed-forward network (FFN) with the MoE layer. Each MoE layer consists of a router and multiple expert networks and each token is assigned to several expert networks by the router. MoE LLMs tend to outperform dense models with the same inference activate parameters.

In addition to its effectiveness in foundation models, some works [12, 8, 58, 20, 32] have utilized MoE in the visual instruction tuning [36] phrase of MLLMs to mitigate task interference, aiming for each expert to specialize in several tasks. Some of them replicate the original FFN in LLMs, while others insert multiple low-rank adaptation [22] modules in parallel with the original FFN, converting LLM into multi-expert architecture. However, they primarily concentrate on LLM but overlook task interference within the visual perceiving process of MLLMs. In contrast, we comprehensively exploit task interference in both vision and language modalities and propose MoME to mitigate them with specialized vision and language experts.

3 Methods

To design a generalist MLLM with powerful multimodal understanding capabilities, we begin by analyzing task interference, a common challenge in current MLLMs (Section 3.1). Then, we propose our Mixture of Multimodal Experts and introduce its main components in Section 3.2.

3.1 Analysis of Task Interference

Task interference is a fundamental and crucial issue in multi-task learning. As current generalist MLLMs are trained with a number of tasks, they naturally suffer from this issue especially when the number of tasks increases. To investigate this issue in a scenario of MLLMs, we will analyze the feature distribution and the performance change of MLLMs on a tailored instruction-tuning dataset.

To demonstrate the external manifestations of task interference, we first construct a mixed instruction-tuning dataset with various VL tasks and split all tasks into four groups. The performance comparisons between MLLMs trained on the mixed dataset and each task group are illustrated in Fig. 1 (a). It is shown that a generalist model trained on the mixed dataset underperforms specialist models on three task groups. We conclude that the generalist model suffers from a notable task interference problem.

In the era of Large models, there are some attempts to handle task interference from various perspectives, such as instruction, architecture, and dataset configuration. Here, we focus on the mainstream direction, architecture design, and try to explore a robust architecture to combat task interference. In terms of architecture, most existing works prefer a mixture of experts in LLMs. The experts can be feed-forward networks or parameter-efficient modules. However, we argue that this paradigm of architectural design is sub-optimal as visual and textual information should be given equal importance.

In Fig. 1 (b-c), we investigate the feature distribution of various task groups on vision and language modalities, respectively. All samples of each task group are fed into vision and text encoders to produce modality features. These features are transformed by using PCA and then visualized to show the distribution. We observe that the feature distributions of different task groups exhibit significant differences in both the vision and language modalities. As mentioned above, the textual differences can be addressed by multiple experts in LLMs, but visual differences lack effective handling. In the following section, we will introduce our Mixture of Multimodal Experts, which simultaneously handles visual and textural differences between tasks to mitigate task interference.

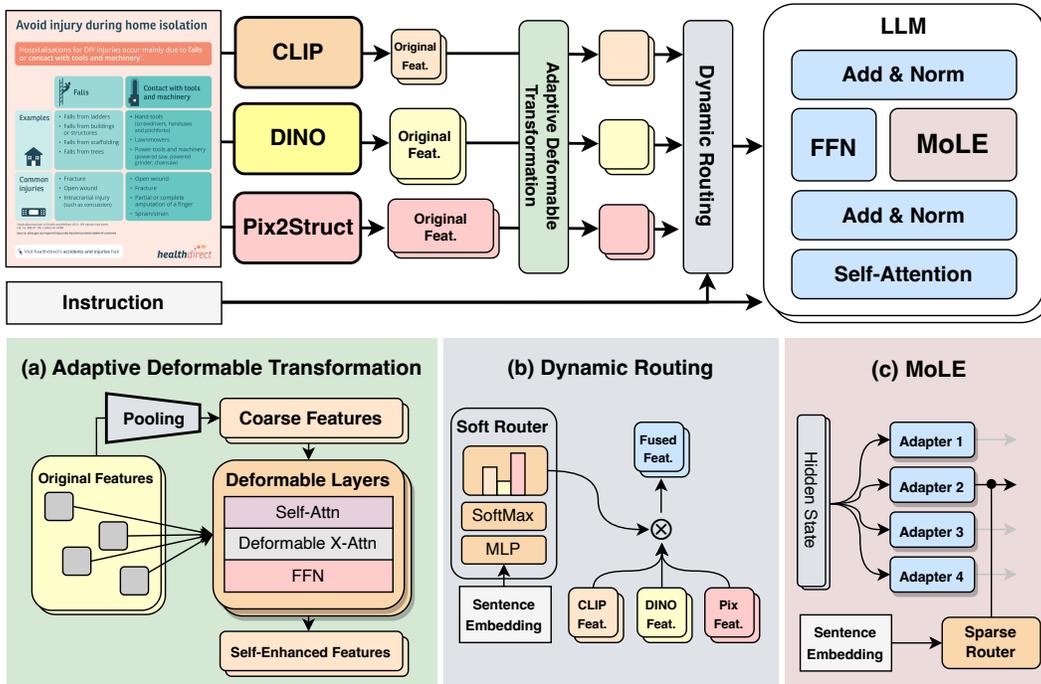


Figure 2: **The overall architecture of the proposed MoME.** The model obtains compressed and self-enhanced visual features from distinct vision encoders through adaptive deformable transformation (a) and aggregates them by dynamic routing (b). The MoLE blocks (c) are integrated into each FFN layer of LLM to improve multitasking capability with little cost.

3.2 Architecture

As illustrated in Figure 2, we present our novel MoME architecture that dynamically mixes vision and language experts. The proposed architecture aims to adaptively aggregate visual information (3.2.1) and select LLM pathways (3.2.2) based on the given instructions.

3.2.1 Mixture of Vision Experts

Before introducing our MoVE architecture, we will present a pilot study that inspires us to design MoVE. Specifically, we design three MLLMs (consists of vision encoder, projection, and LLM), which share the same architecture except vision encoders. These three MLLMs use three distinct vision encoders, CLIP, DINOv2, Pix2Struct, respectively. After training them using the same data and strategies, we found that MLLMs with different vision encoders excelled in specific tasks, as presented in Fig. 3. the MLLM with CLIP-ViT is good at general tasks and regional caption tasks. the MLLM with DINOv2 excels in REC, a visual grounding task. the MLLM with Pix2Struct is outstanding in text-intensive document understanding tasks. However, each model had weaknesses and none could achieve uniformly excellent performance across all tasks.

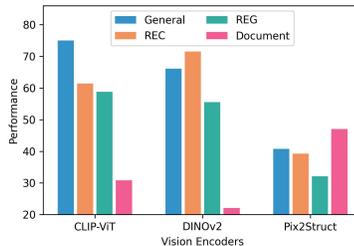


Figure 3: Comparison of MLLMs with different vision encoders.

Inspired by the above study, we propose MoVE to combine various off-the-shell vision encoders and effectively align and aggregate their visual features. The key components in MoVE are an adaptive deformable transformation module and an instruction-based soft router. The former aims to align the features from various vision encoders, and the latter seeks to modulate and aggregate transformed features based on the instructions.

Adaptive Deformable Transformation. Given the diversity in architecture and training methods of different vision encoders, the issue of mismatched visual representations in terms of sequence length and feature space becomes significant. While current researches [62, 27, 33] often focus on models like CLIP-ViT [52] and DINO [50], which share similar data processing pipeline, the mismatch problems are less important and frequently overlooked. However, the aspect ratios of Pix2Struct [30] feature shapes vary depending on the input image. Simply combining them through padding and addition will lead to the misalignment among visual tokens and the damage of visual information. To tackle this challenge, we innovatively design an adaptive deformable transformation (ADT) module that effectively transforms features from diverse vision encoders f into a unified-length feature sequence \hat{f} , ensuring more coherent and informative visual representations for subsequent aggregations.

As illustrated in Fig. 2 (a), the ADT module consists of a 2D adaptive pooling layer and M deformable attention layers (\mathbb{D}). It attempts to automatically select the corresponding information from original features f to refine the coarse features obtained by 2D adaptive pooling $\mathcal{D}(f)$,

$$\hat{f} = \mathbb{D}^M(\mathcal{D}(f), f) \quad (1)$$

Inspired by Deformable DERT [71], we choose deformable cross-attention for its 2D sampling mechanism, which is ideal for interactions among visual feature maps of varying shapes. Meanwhile, it converges fast and has computational and memory efficiency. Specifically, each deformable layer consists of a self-attention layer, a deformable cross-attention, and a feed-forward layer. The crucial select operation occurs in the deformable cross-attention layer, which takes the output of the upper self-attention as query $q \in \mathbb{R}^{L \times C}$, samples the original feature map $f \in \mathbb{R}^{H \times W \times C}$, and outputs result $\mathcal{O} \in \mathbb{R}^{L \times C}$. In this module, the first step is to generate attention weights $w \in \mathbb{R}^{L \times N_h \times N_p}$ and L sets of 2D sampling points, denoted as p , from the input queries q . For each set, there are $N_h \times N_p$ points, where N_h signifies the number of attention heads and N_p represents the number of points sampled by each head. The sample points and attention weights generation process is as follows,

$$p_{ijk} = (\mathcal{P}_p(q_i)_{jk} + R_i), i \in \{1, \dots, L\}, j \in \{1, \dots, N_h\}, k \in \{1, \dots, N_p\} \quad (2)$$

$$w = \mathcal{P}_w(q) \quad (3)$$

where \mathcal{P} denotes the linear projector and $R \in \mathbb{R}^{L \times 2}$ is a learnable vector called reference point. Then, the corresponding information is sampled by attention heads from the value feature maps $\mathcal{P}(f)_j$ projected and split on the last dimension for each head. The sampling mechanism of each attention head is as follows,

$$o_{ij} = \sum_{k=1}^{N_p} w_{ijk} \cdot \text{Sample}(\mathcal{P}_v(f)_j, p_{ijk}), i \in \{1, \dots, L\}, j \in \{1, \dots, N_h\} \quad (4)$$

The results of multiple attention heads are concatenated and projected to obtain the output feature \mathcal{O} , which is the input of subsequent feed-forward layer.

$$\mathcal{O} = \mathcal{P}_o(o) \quad (5)$$

Instance-level Soft Router. Since images from different groups of VL tasks exhibit distinct feature distributions, there is no one-fits-all strategy to aggregate them. To address this, we propose to generate a customized fusion ratio for each sample based on its instruction.

Specifically, we devise an instance-level soft router G_s , as depicted in Fig 2 (b). The router generates corresponding ratios for the visual representations from different vision encoders, followed by a weighted sum of these visual representation features \hat{f} , which can be formulated as,

$$G_s(I) = \text{SoftMax}(W_2(\sigma(W_1 I + b_1)) + b_2) \quad (6)$$

$$\mathcal{F} = \sum_{i=1}^N G_s(I)_i \times \hat{f}_i \quad (7)$$

where N is the number of vision experts and σ denotes GeLU [21]. The router is a multilayer perceptron (MLP) followed by a SoftMax layer to ensure that the sum of the weights equals 1. The instruction is first passed through Sentence-BERT [53] to extract sentence embedding I , which is then fed into the router.

Table 1: **Comparison of various MoVE strategies.** "Gen." and "Doc." respectively denote average performances on General and Document. "Avg." means the average of the four group scores. The best performances are marked as bold.

#	Encoder	Strategy		Performance				
		Transformation	Aggregation	Gen.	REC	REG	Doc.	Avg.
1	CLIP	AvgPool	-	75.04	61.42	58.79	30.84	56.52
2	DINO	AvgPool	-	66.09	71.52	55.58	22.10	53.82
3	Pix	AvgPool	-	40.75	39.26	32.11	47.05	39.79
4	MoVE	AvgPool	Addition	70.36	74.89	57.55	32.83	58.91
5	MoVE	ADT	Addition	74.35	76.93	61.01	39.23	62.88
6	MoVE	ADT	Router	79.05	81.92	63.82	52.77	69.39

3.2.2 Mixture of Language Experts

We introduce MoE architecture in LLM, aiming for each expert to specialize in several tasks. However, conventional MoE methods typically incorporate multiple parallel FFN layers in one block, significantly increasing training costs and memory consumption. To meet the multi-task learning needs in the instruction tuning stage, we incorporate several parameter-efficient adapters [9] parallel to FFN. Each adapter enhances the original FFN with task-specific understanding capabilities, thus effectively enhancing the multitasking abilities with a few computation costs.

We insert an MoE block parallel to each FFN layer in LLM. As depicted in Fig. 2 (c), The MoE block consists of several low-rank adapters and an instance-level sparsely-activated router G_h . The adapter is designed as a bottleneck structure for computational efficiency, featuring a down-projection layer \mathcal{P}_{down} , a ReLU layer σ , and an up-projection layer \mathcal{P}_{up} . Moreover, a learnable scalar \mathbf{s} is multiplied in the output to weigh the importance adaptively. The whole low-rank adaptation process is as follows,

$$y = \mathbf{s} \cdot \mathcal{P}_{up}(\sigma(\mathcal{P}_{down}(x))) \quad (8)$$

The router is an MLP network followed by a top-1 gate function to ensure the output is a one-hot vector $G(I) \in \{0, 1\}^K$. The router generates the selection based on the sentence embeddings I used in MoVE. Each sample is routed to the corresponding adapter to calculate the adapted value o , which can be further added to the output of the original FFN. The whole process of the MoLE block is as follows,

$$o = \sum_{k=1}^K G_h(I)_k \times y_k \quad (9)$$

where K denotes the number of experts.

4 Experimental Results and Analysis

We collect 24 datasets and categorize them into four groups for instruction-tuning and evaluation, the details of which can be found in Appendix B.

4.1 Analysis on MoVE

We conduct experiments on MLLMs with different vision encoders under the same training strategy to verify the effectiveness of the two key components in MoVE: ADT and router. Experimental results are summarized in Table 1. We take the multitasking performances of MLLMs with a single vision encoder as our baselines. The adaptive average pooling is applied to the visual representation from DINO and Pix2Struct, ensuring that the lengths of visual tokens fed into LLM are consistent. Experiments #1-3 show that MLLMs using CLIP, DINO, and Pix2Struct as vision encoders exhibit distinct strengths in General, REC, and Document tasks, respectively. Moreover, in the REG task, which requires both captioning and visual grounding abilities, the performance of MLLMs with CLIP and DINO significantly surpasses that of those using Pix2Struct. We can conclude that a single vision encoder cannot meet the visual perception needs of all tasks.

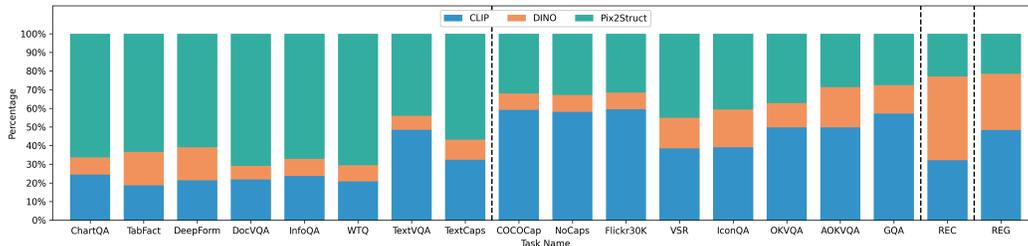


Figure 4: **Distribution of vision experts routing results.** In each bar, the lengths of different colors represent the frequency with which each expert is selected.

To make different vision encoders work together in perception, we first aggregate different visual representations by addition (#4). The average performance does improve compared with models with single vision encoders (#1-3). However, such a straightforward method does not bring much improvement, and even some sub-items have declined. This is due to the mismatch and interference among different visual features, which severely compromise their respective visual information. To make visual features aligned well and reduce information loss, we first replace the pooling with the proposed ADT network. As demonstrated in Experiment #5, ADT consistently enhances performance across four task groups, yielding an average improvement of 4 points. Based on the transformed visual features, we further replace the naive addition mixture process with the router to modulate and aggregate them adaptively according to instructions (#6). This achieves an impressive performance that significantly outperforms the addition method and the methods with a single vision encoder across all sub-tasks. Additional experiments on the deformable mechanism can be found in Appendix C. These experimental results prove that ADT and router are crucial and effective components of MoVE to mitigate interference and optimally utilize the capabilities of each visual expert.

Visualization of Routing Results. To provide a deeper understanding of the MoVE’s adaptive routing process, we visualize the routing outcomes across various tasks. Since the feature scale varies across different vision encoders, we cannot directly consider the output of the router as expert importance. Instead, we integrate the output features from vision encoders, taking the magnitude of the weighted feature vector as the importance metric. Because the final aggregated result will lean towards the side with the larger magnitude.

As displayed in Fig. 4, for tasks that involve text recognition and graph understanding, such as ChartQA and DocVQA, the features from the Pix2Struct encoder are dominant. In contrast, the model utilizes more CLIP features in image-level VQA tasks like COCOCaption [11], NoCaps [2], and Flickr30K [69]. Notably, for TextCaps [59], a task that requires two kinds of visual information, the router shows a preference for balancing the CLIP and Pix2Struct branches. For tasks that focus on visual grounding ability like REC [42], and REG [37], the model uses more DINO features to perceive region-level visual information. These observations indicate that MoVE can adaptively modulate the features transformed from various vision encoders.

4.2 Analysis on MoLE

We conduct ablation experiments to explore the best practice of MoLE, which are summarized in Table 2. We take the model with a single adapter in each FFN (#1) as baseline, which suffers severe task interference. Then, we replace the plain adapter with MoLE. As summarized in Table 2 #2-4, we test three kinds of MoLE routers with different inputs: token hidden states (MoLE-T), sentence embedding (MoLE-I), and a mixture of both (MoLE-IT). The token hidden states and sentence embedding are concatenated on the last dimension as the input for the MoLE-IT router. The experimental results indicate that all MoLE variations outperform the plain adapter, with the sentence-embedding-based router achieving the highest average performance.

We also explore two strategies for expert load balance in MoLE, which are tabulated in Table 2 #5-6. MoLE-I+GS introduces variability to the router by adding Gumbel-distributed noise to the logits [25], aiming to prevent certain experts from never being selected. MoLE-I+LB uses auxiliary loss [18] to

Table 2: **Comparison of different MoLE configurations** "-T", "-I", and "-IT" respectively represent MoLE with routers based on token, instance, and both. "GS" and "LB" represent the implementation of Gumble Softmax [25] and Load Balance [18] based on the MoLE-I, respectively.

#	VE	LE	Gen.	REC	REG	Doc.	Avg.
1	CLIP	Adapter	74.50	63.80	59.24	31.73	57.32
2	CLIP	MoLE-IT	75.61	65.63	59.17	32.46	58.22
3	CLIP	MoLE-T	75.35	66.09	58.09	32.12	57.91
4	CLIP	MoLE-I	75.62	66.95	60.90	32.27	58.94
5	CLIP	MoLE-I + GS	74.87	63.97	59.00	31.69	57.38
6	CLIP	MoLE-I + LB	75.42	64.74	59.48	32.05	57.92
7	MoVE	Adapter	79.05	81.92	63.82	52.77	69.39
8	MoVE	MoLE-I	79.65	81.58	64.83	53.69	69.94

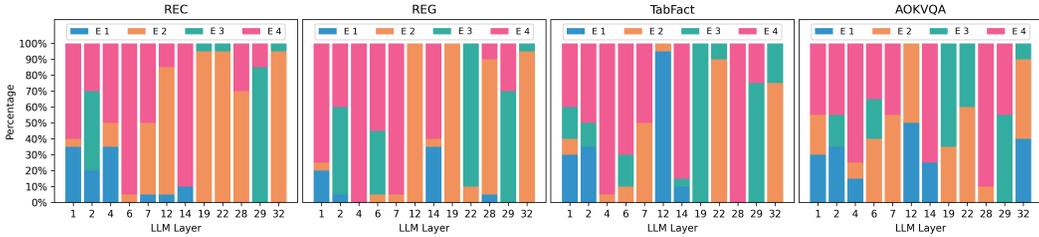


Figure 5: **Distribution of language experts routing results.** The figures depict the expert load conditions of four selected datasets. In each bar, the lengths of different colors represent the frequency with which each expert is selected.

impose load balancing. However, we find that these methods are not suitable for our MoLE as they perform worse than the plain MoLE-I configuration.

Through comprehensive comparative experiments, we choose MoLE-I as the default configuration for MoME. By training based on the MoVE model, MoME further enhances the multitasking capability of MLLM, as shown in Experiments #7 and #8.

Visualization of routing results. To provide a deeper understanding of our MoLE module, we sample several instances from each dataset and calculate their average routing outcomes. In Fig. 5, we present the expert load conditions of four selected datasets, each representing a kind of VL task. At the beginning, the router assigns equal probabilities to each expert as it is randomly initialized. However, after training, the routing distributions of MoE blocks vary significantly across tasks, as shown in Fig. 5. This means the language experts in our MoLE module gradually specialize in distinct task domains during training. In the inference process, the router adaptively chooses the optimal expert to achieve strong multitasking capabilities. Meanwhile, we can observe strong resemblances in the expert routing results of similar tasks, which are further analyzed in Appendix D.

4.3 Comparison with state-of-the-art MLLMs

We summarize the evaluation results of MoME and other MLLMs with similar resource consumption on popular VL tasks in Table 3. The results show that our MoME method achieves promising outcomes on most datasets compared with other generalist and MoE MLLMs, especially on TextCaps, Flicker30K, and IconQA.

4.4 Qualitative Analysis

We present several visualized examples of our MoME model from distinct dataset groups along with their MoVE and MoLE routing results in Fig. 6. In the REC case, DINOv2 accounts for nearly 50% among vision experts, providing fine-grained visual information. So the model can recognize the blue

Table 3: Comparison with state-of-the-art MLLMs with similar resource consumption. MoME achieves superior performances on most datasets and is capable of a broader range of VL tasks.

Model	Doc VQA	Chart QA	Text Caps	Text VQA	Flickr 30K	Icon QA	VSR	GQA	Ref COCO
Shikra-7B [7]	-	-	-	-	-	-	-	-	80.2
Ferret-7B [68]	-	-	-	-	-	-	-	-	82.5
IBLIP [15]	-	-	-	50.7	82.4	43.1	54.3	49.2	-
LLaVA-v1.5 [35]	-	-	-	58.2	-	-	-	62.0	-
LION [5]	-	-	108.8	-	87.4	54.89	73.8	51.6	-
DocPedia [19]	47.1	46.9	-	60.2	-	-	-	-	-
MoE-LLaVA [32]	-	-	-	50.2	-	-	-	61.1	-
MixLoRA [58]	-	-	-	40.0	-	-	51.2	-	-
MoCLE [20]	-	-	-	57.1	81.9	46.3	64.7	49.3	-
LLaVA-MoLE [8]	30.0	42.4	-	-	-	-	-	-	-
MoME	50.8	57.2	130.8	53.2	94.6	61.4	61.9	59.7	83.2

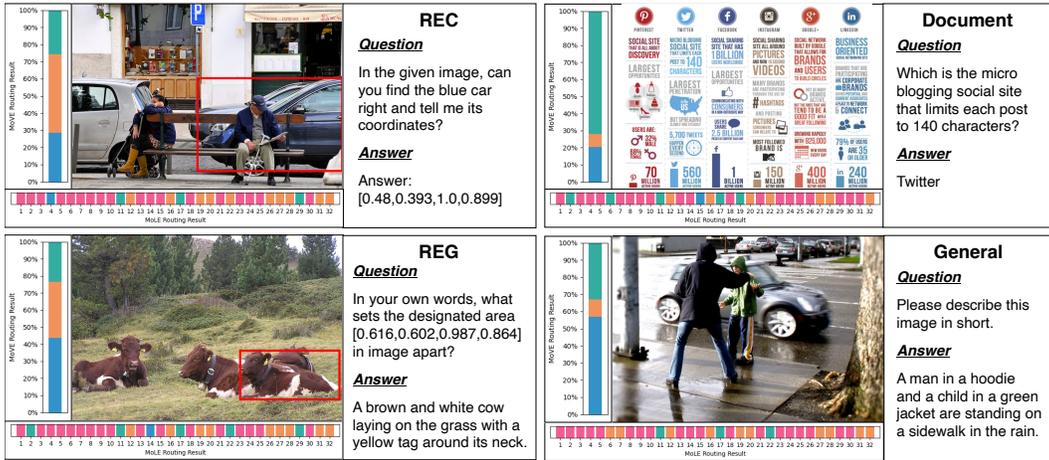


Figure 6: Visualization of samples along with their routing result distributions. The MoVE distributions on the left represent Pix2Struct, DINOv2, and CLIP-ViT from top to bottom. MoLE is on the bottom, with different colors indicating different experts.

car in the background and provide its precise bounding box. The Pix2Struct branch accounts for over 70% in the Document case for structured text understanding. The REG case utilizes information from both the CLIP-ViT and DINOv2 to locate objects and generate captions. In contrast, the conventional caption task in the General group only requires an image-level perception, so the CLIP-ViT is dominant. Remarkably, we can observe significant differences among the MoLE routing results. These examples show how MoME selects vision and language experts to adapt to various tasks.

5 Conclusion

This work investigates task interference when training a generalist MLLM across various VL tasks. To mitigate it, we propose MoME, which specializes in both vision and language modality to adapt to task differences. Extensive experiments validate the efficiency of MoME as a generalist MLLM.

However, due to the limitations of computing resources, We have not yet expanded our approach to more data and more modalities for experiments. Nonetheless, we believe that the proposed MoME is versatile and can be applied to constructing generalist models in a wider range of multimodal domains. We hope MoME will inspire new research in general-purpose multimodal AI and its applications.

6 Acknowledgement

This study is supported by National Natural Science Foundation of China (Grant No. 62306090, No. 62476071, No. 62236003), Natural Science Foundation of Guangdong Province of China (Grant No. 2024A1515010147), and Shenzhen College Stability Support Plan (Grant No. GXWD20220817144428005).

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019.
- [3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.
- [4] Ali Furkan Biten, Ruben Tito, Andres Mafra, Lluís Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019.
- [5] Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. Lion: Empowering multimodal large language model with dual-level visual knowledge. In *CVPR*, 2024.
- [6] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.
- [7] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm’s referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.
- [8] Shaoxiang Chen, Zequn Jie, and Lin Ma. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. *arXiv preprint arXiv:2401.16160*, 2024.
- [9] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adapt-former: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.
- [10] Wenhui Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyu Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*, 2019.
- [11] 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.
- [12] Zeren Chen, Ziqin Wang, Zhen Wang, Huayang Liu, Zhenfei Yin, Si Liu, Lu Sheng, Wanli Ouyang, Yu Qiao, and Jing Shao. Octavius: Mitigating task interference in mllms via lora-moe. In *ICLR*, 2024.
- [13] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.
- [14] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.
- [15] 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.
- [16] Chuntao Ding, Zhichao Lu, Shangguang Wang, Ran Cheng, and Vishnu Naresh Boddeti. Mitigating task interference in multi-task learning via explicit task routing with non-learnable primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7756–7765, 2023.

- [17] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, pages 19358–19369, 2023.
- [18] 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.
- [19] Hao Feng, Qi Liu, Hao Liu, Wengang Zhou, Houqiang Li, and Can Huang. Docpedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. *arXiv preprint arXiv:2311.11810*, 2023.
- [20] 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.
- [21] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [23] Drew A Hudson and Christopher 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.
- [24] 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.
- [25] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- [26] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.
- [27] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Xiaopeng Zhang, Jin Li, Hongkai Xiong, and Qi Tian. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023.
- [28] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.
- [29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [30] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023.
- [31] Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Dongmei Jiang, and Liqiang Nie. Optimus-1: Hybrid multimodal memory empowered agents excel in long-horizon tasks. In *NeurIPS*, 2024.
- [32] Bin Lin, Zhenyu Tang, Yang Ye, Jiayi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024.
- [33] 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.
- [34] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023.
- [35] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

- [36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.
- [37] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4856–4864, 2017.
- [38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [39] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. Deepseek-vl: Towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.
- [40] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.
- [41] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1851–1860, 2019.
- [42] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.
- [43] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.
- [44] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.
- [45] 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.
- [46] 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.
- [47] 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.
- [48] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruvi Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.
- [49] Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.
- [50] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [51] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*, 2015.
- [52] 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.
- [53] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.
- [54] 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.

- [55] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10023–10031, 2019.
- [56] Rui Shao, Tianxing Wu, and Ziwei Liu. Detecting and grounding multi-modal media manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2023.
- [57] Rui Shao, Tianxing Wu, Jianlong Wu, Liqiang Nie, and Ziwei Liu. Detecting and grounding multi-modal media manipulation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [58] Ying Shen, Zhiyang Xu, Qifan Wang, Yu Cheng, Wenpeng Yin, and Lifu Huang. Multimodal instruction tuning with conditional mixture of lora. *arXiv preprint arXiv:2402.15896*, 2024.
- [59] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020.
- [60] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.
- [61] Stacey Svetlichnaya. Deepform: Understand structured documents at scale, 2019.
- [62] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.
- [63] 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.
- [64] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3614–3633, 2021.
- [65] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.
- [66] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv preprint arXiv:2310.05126*, 2023.
- [67] Qilang Ye, Zitong Yu, Rui Shao, Xinyu Xie, Philip Torr, and Xiaochun Cao. Cat: Enhancing multimodal large language model to answer questions in dynamic audio-visual scenarios. In *ECCV*, 2024.
- [68] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023.
- [69] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.
- [70] 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.
- [71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A Implementation Details

A.1 Architecture Details

The MoME includes three off-the-shelf vision encoders: CLIP-ViT, DINOv2, and Pix2Struct. For CLIP-ViT, we use ViT-G/14 from EVA [17] without the last layer. The DINOv2 [50] is the official pre-trained version of ViT-L/14 without registers. For Pix2Struct, we use the vision branch of pre-trained Pix2Struct-base model [30]. The ADT consists of a 2D adaptive average pooling layer and a six-layer deformable attention network. The hidden size of ADT is the same as its corresponding vision encoder. We use 8 attention heads and each head samples 4 points in the deformable cross-attention process. In MoLE, the hidden dimension of each adapter is set to 64. We use Vicuna-v1.5(7B) [13] as our pre-trained LLM.

A.2 Training Details

Our training process comprises two stages. In stage 1, we train the model with MoVE and a single adapter in LLM for 30k steps with a batch size of 64. The learning rate is warmed up linearly from 0 to 5e-4 across 1000 steps and then reduces to a minimum of 0 using cosine decay. The AdamW [38] optimizer is employed with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.05. In stage 2, we load the checkpoint of stage 1 and replicate the weights of adapters to initialize MoLE, while keeping everything else unchanged.

We use a single node with A800 80GB \times 8 for all experiments, the entire training is done in one day including stage 1 and stage 2.

B Details of Multitasking Benchmark

We collected 24 datasets and categorized them into four groups for instruction-tuning and evaluation, as shown in Fig. 7. For most vision-language (VL) tasks, we used the datasets in both the training and evaluation phases. However, we only use NoCaps for evaluation because it only has an evaluation set, and we exclude the VSR training data due to its simplicity. During the training process, we mix the datasets within the same group into one large dataset, so the probability of each sub-dataset being sampled equals their size as a proportion of the total. However, we ensure the sample ratio of each group dataset is the same. For evaluation, we compute the overall score for each category by averaging its subitem evaluation results. We follow InstructBLIP [15], Shikra [7], and UReader [66] for our evaluation metrics, which are tabulated in Table 5. Notably, the model only takes images as visual information without introducing OCR tokens like InstructBLIP.

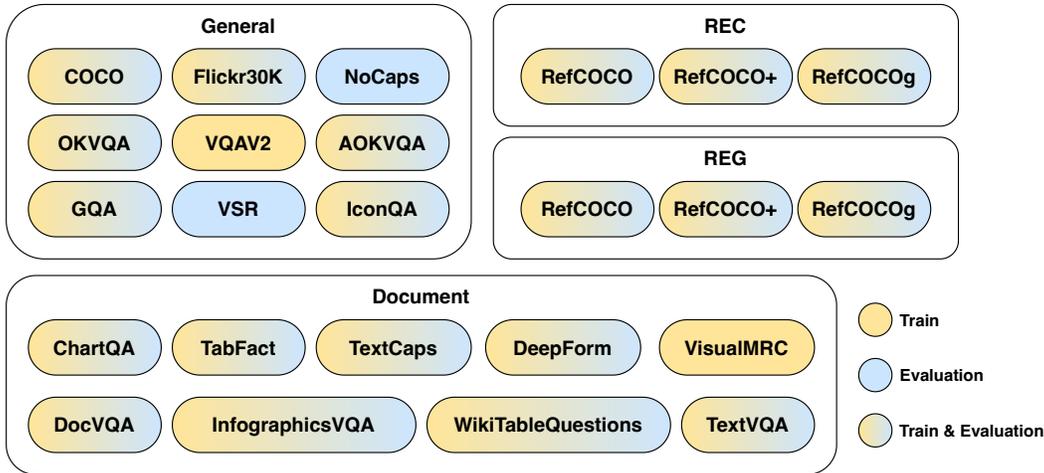


Figure 7: Multitask learning and evaluation datasets and their corresponding categories.

Table 4: **Ablation studies of deformable mechanism.** "Gen." and "Doc." respectively denote average performances on General and Document. "Avg." means the average of the four group scores. The best performances are marked as bold.

#	Deformable Mechanism	Aggregation	Gen.	REC	REG	Doc.	Avg.
1	×	Router	75.27	78.17	61.64	42.46	64.39
2	✓	Router	79.05	81.92	63.82	52.77	69.39

C Additional Ablation Experiments

To evaluate the effectiveness of the deformable mechanism of ADT, we conduct an ablation experiment by replacing the deformable cross-attention in MoME with standard cross-attention. As shown in Table 4, the model without deformable mechanism (#1) presents much worse performances than the original MoME (#2) consistently. We attribute this decline to the strong inductive bias of deformable attention in processing 2D feature maps.

D Additional Visualization and Analysis

We present the routing distributions across different datasets of all MoLE routers in Fig. 8. In general, the routing results differ significantly among different task groups, while the routing preferences are similar within the same task group. From the perspective of router preferences, the datasets can be classified as text-rich (ChartQA - TextCaps), caption (COCOCap - Flickr30K), VQA (IconQA - GQA), REC, and REG. It can prove that the MoLE captures the modularity of the tasks and mitigates task interferences through differential expert routing.

E Societal Impacts

MoME utilizes pre-trained large language models (LLMs), which inherently carry limitations from LLMs. These limitations include the potential for generating inaccurate information or biased outputs. To address these issues, we enhance the model’s visual perception ability with MoVA and conduct vision-language instruction tuning on high-quality datasets. Despite these improvements, we advise caution and recommend thorough safety and fairness assessments before deploying MoME models in any downstream applications.

Table 5: Summary of the evaluation datasets.

Task	Dataset	Split	Metric
General	COCOCap [11]	karpathy-test	CIDEr [65](↑)
	Flickr30K [69]	karpathy-test	CIDEr [65](↑)
	NoCaps [2]	val	CIDEr [65](↑)
	OKVQA [44]	val	Accuracy(↑)
	AOKVQA [54]	val	Accuracy(↑)
	GQA [23]	test-dev	Accuracy(↑)
	Visual Spatial Reasoning (VSR) [34]	val	Accuracy(↑)
IconQA [40]	test	Accuracy(↑)	
REC	RefCOCO [28]	val & testA & testB	Accuracy(↑)
	RefCOCO+ [28]	val & testA & testB	Accuracy(↑)
	RefCOCog [43]	val & test	Accuracy(↑)
REG	RefCOCO [28]	val & testA & testB	CIDEr [65](↑)
	RefCOCO+ [28]	val & testA & testB	CIDEr [65](↑)
	RefCOCog [43]	val & test	CIDEr [65](↑)
Document	ChartQA [45]	test	Relax Accuracy [49](↑)
	TabFact [10]	test	Accuracy(↑)
	DeepForm [61]	test	F1 Score(↑)
	DocVQA [47]	test	ANLS [4](↑)
	InfographicsVQA [46]	test	ANLS [4](↑)
	WikiTableQuestions (WTQ) [51]	test	Accuracy(↑)
	TextCaps [59]	val	CIDEr [65](↑)
TextVQA [60]	val	Accuracy(↑)	

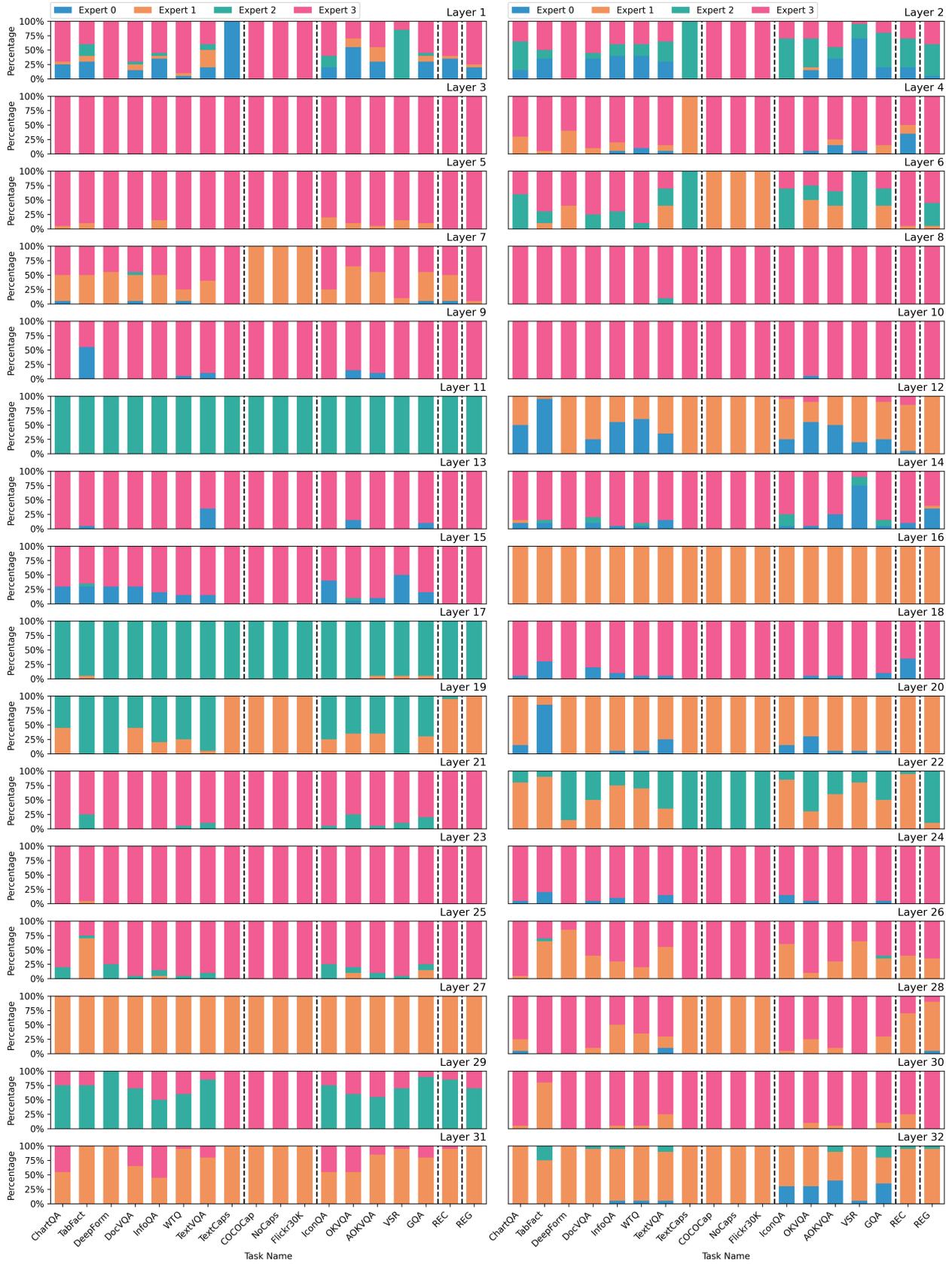


Figure 8: Distribution of all language experts routing results across all tasks.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The limitations of our work are discussed in the Conclusion section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The formulas are numbered and cross-referenced correctly in this paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully and clearly describe the model architecture in the main body, and provide a detailed training process and model configuration in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We use publicly available datasets for training and evaluation, which enables easy reproducibility. Upon acceptance, we will release the code and data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide detailed training and test configuration details in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: MLLM researches typically do not test and report error bars of experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide sufficient information on the computer resources in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss societal impacts in the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We describe safeguards about the pre-trained language model in the Appendix.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes] .

Justification: The creators and original owners of the assets used in the paper are properly credited and explicitly mentioned with respect.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We will release our code with detailed documentation in the future.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: we do not conduct any crowdsourcing experiments so this item is not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: we do not conduct any experiments with human subjects so this item is not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.