Mixture of Cluster-Conditional LoRA Experts for Vision-Language Instruction Tuning

Anonymous EMNLP submission

Abstract

Instruction tuning of Large Vision-language Models (LVLMs) has revolutionized the development of versatile models with zero-shot generalization across a wide range of downstream vision-language tasks. However, the diversity of training tasks of different sources and formats would lead to inevitable task conflicts, where different tasks conflict for the same set of model parameters, resulting in sub-optimal instruction-following abilities. To address that, we propose the Mixture of Cluster-conditional LoRA Experts (MoCLE), a novel Mixture of Experts (MoE) architecture designed to activate the task-customized model parameters based on the instruction clusters. A separate universal expert is further incorporated to improve generalization capabilities of MoCLE for novel instructions. Extensive experiments on InstructBLIP and LLaVA demonstrate the effectiveness of MoCLE.

1 Introduction

There has been a continuously increasing trend to develop intelligent assistants that can follow human instructions (Brown et al., 2020; OpenAI, 2022; Chen et al., 2023b), with instruction tuning emerging as a notably effective approach. This method leverages large-scale well-formatted instruction data to empower Large Language Models (LLMs) to execute various human instructions, showcasing their ability to generalize across novel unseen tasks (Longpre et al., 2023). Likewise, efforts have been made to introduce similar capabilities to Large Vision-language Models (LVLMs) (Bai et al., 2023; Zhang et al., 2023; Ye et al., 2023; Chen et al., 2023c,a), including LLaVA series (Liu et al., 2023b,a), MiniGPT-4 (Zhu et al., 2023) and InstructBLIP (Dai et al., 2023).

It is observed that for both the LLMs (Sanh et al., 2021; Wang et al., 2022; Chung et al., 2022) and LVLMs (Bai et al., 2023; Zhao et al., 2023; Li et al., 2023b), the ability to generalize to novel unseen instructions necessitates multi-task instruction tuning, *i.e.*, training on a diverse collection of instruction-following tasks. However, the complexity of various instruction tasks brings difficulties for model fine-tuning. Specifically, Wei et al. (2021) find that for certain model sizes, multi-task instruction tuning even fails to bring performance gains for zero-shot tasks compared to the original models. This is mainly attributed to the *negative transfer* phenomenon (Zhang and Yang, 2017) during multi-task instruction tuning, where the model struggles to optimize the losses of multiple conflicted tasks, leading to sub-optimal performance.

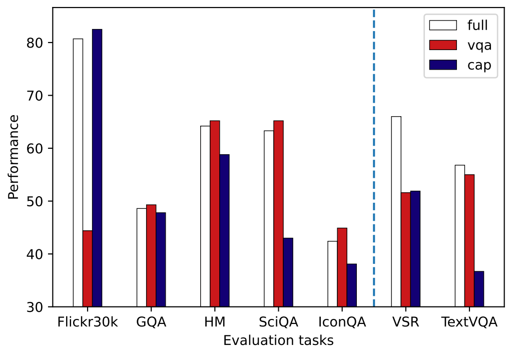


Figure 1: **Performance of the instruction-finetuned LVLMs on zero-shot tasks**, where larger values indicate better performance. Only 2 out of 7 tasks benefit from instruction tuning from all the data, while the task experts show better performance on the other 5 tasks (*i.e.*, Flickr 30K, GQA, HM, SciQA and IconQA).

Similarly, tasks for vision-language instruction tuning (*e.g.*, visual question answering and image captioning) focus on different perspectives of LVLMs. This results in conflicts as most studies adopt sharing of all parameters. In our preliminary study, we split the instruction data into two *disjoint* subsets (*"cap"* for image captioning, and *"vqa"* for visual question answering). We then train InstructBLIP (Dai et al., 2023) using

LoRA (Hu et al., 2021) on three data sets (*"cap"*, *"vqa"* and the full data *"full"*) to obtain three sets of parameters (*i.e.*, task experts). Following the held-out evaluation protocol (Dai et al., 2023), we evaluate these experts on the unseen datasets/tasks with the best expert. As shown in Figure 1, on 5 out of the 7 downstream tasks, the InstructBLIP instruction-tuned on all the data is outperformed by the task expert finetuned with only a subset of data. Among the 5 tasks, Flickr30k belongs to *"cap"*, and SciQA, GQA and IconQA belong to *"vqa"*. This shows that *instruction tuning on similar tasks brings positive transfer to downstream tasks, while training on the full data with dissimilar tasks can hurt generalization performance*.

The use of disjoint task experts above is a naïve solution to negative transfer, where we manually partition the training tasks and train each expert separately. However, it has several limitations: (1) The taxonomies such as *"vqa"* and *"cap"* require human expertise, and are difficult to scale as the number of tasks grows. (2) The ability to generalize to unseen tasks is inhibited, as we do not know which expert to choose for novel tasks, while some new tasks might benefit from multiple training tasks (*e.g.*, VSR and TextVQA as in Figure 1). In this regard, specialization and generalization of LVLMs becomes a dilemma.

This paper aims to develop an automatic and practical partition strategy and a network architecture that strikes a balance between specialization and generalization. In particular, we propose the *Mixture of Cluster-conditional LoRA Experts* (MoCLE) for vision-language instruction tuning. In this proposed framework, we first cluster instructions of all the training data into several clusters via a pre-trained clustering model. In this way, similar tasks that can bring positive transfer to each other are automatically grouped into the same cluster, while different tasks that may cause conflict are separated (more justifications for the use of instruction clusters are detailed in Sec. 3.2). Then we construct several task experts, with each focusing on a specific cluster. Using the cluster as condition, a router dispatches the input data to one of the specialized task experts and an universal expert that is shared among all data. As we activate a specialized expert for a group of similar tasks, tasks that are less similar are learned via separate experts, mitigating task conflicts. Meanwhile, since the universal expert trained on all tasks also contributes to the model outputs, we can enjoy generalization and specialization simultaneously.

We validate effectiveness of MoCLE on InstructBLIP (Dai et al., 2023) and LLaVA-1.5 (Liu et al., 2023a) and observe remarkable performance gains compared to dense models and other MoE baselines(Chen et al., 2023d, 2024).

The main contributions of this work contain the following three parts,

1. We identify the negative transfer phenomenon (Liu et al., 2022b; Zhili et al., 2023) as tasks conflict during instruction tuning of LVLMs.

2. We propose *Mixture of Cluster-conditional LoRA Experts* (MoCLE), a novel parameter-efficient finetuning framework suitable for the vision-language instruction tuning, to mitigate task conflicts and enjoy the benefits of huge data training simultaneously.

3. Our proposed MoCLE achieves remarkable performance gains on held-in/out tasks compared to dense models and other MoE baselines(Chen et al., 2023d, 2024).

2 Related Work

2.1 Multi-Task Instruction Tuning

Instruction tuning (Sanh et al., 2021; Wei et al., 2021) fine-tunes a language model across tasks and instruction templates to convey task intentions. Its goal is to teach the model to understand relationships between instructions and input/output pairs, enabling generalization to unseen tasks with novel instructions. Increasing the number of instructions (Sanh et al., 2021), tasks (Wang et al., 2022; Chung et al., 2022), and data diversity (Zhou et al., 2023) have shown to be effective in improving performance. However, Wei et al. (2021) find that for certain model sizes, instruction tuning fails to outperform untuned models on unseen tasks due to full capacity utilization for learning task mixtures. Our work addresses this in vision-language instruction tuning using specialized experts.

2.2 Mixture of Experts (MoE)

MoE models (Jacobs et al., 1991; Jordan and Jacobs, 1993; Shazeer et al., 2017) are renowned for their ability to increase model capacity through parameter expansion. Recent research integrates MoE with adapters, exploring how pretrained adapters can be effectively combined (Wu et al., 2024b), and how they enhance performance in both few-shot

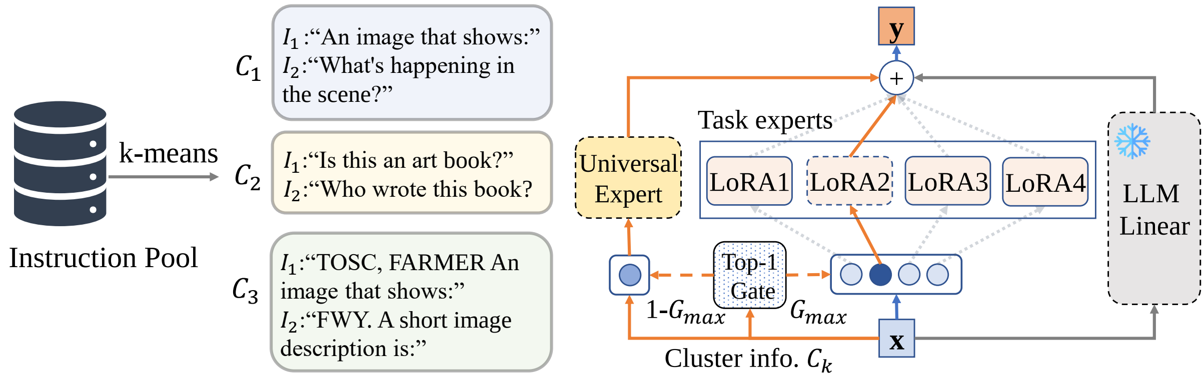


Figure 2: **Overall pipeline of MoCLE.**

(Huang et al., 2023) and zero-shot scenarios (Jang et al., 2023; Muqeeth et al., 2024). Another line of research (Chen et al., 2024; Wu et al., 2024a; Luo et al., 2024; Zadouri et al., 2023; Chen et al., 2023d) focuses on augmenting model capacity in a parameter-efficient manner. However, these approaches do not explicitly incorporate task/domain priors during the routing process, which might be limited when handling task conflicts. Two concurrent studies (Liu et al., 2024; Li et al., 2024) incorporate domain information for expert routing. Unlike our approach, however, they do not address zero-shot generalization to unseen tasks.

3 Methodology

In this section, we start with the formulation of LVLM instruction tuning and an analysis of the limitations of task experts. We then introduce the proposed MoCLE. The overall framework is shown in Figure 2.

3.1 Problem Formulation

Suppose that there is a set of datasets that are divided into held-in and held-out datasets (Dai et al., 2023). A large vision-language model is first fine-tuned on the held-in dataset, and then evaluated on the held-out dataset in a zero-shot manner. To unify and diversify input-output formats and promotes instruction tuning, several task templates $\{T_i\}$ are designed to wrap the raw inputs, which is a pair of text X_{txt} and image X_{img} from the dataset. For example, "*Given the image, answer the question with no more than three words.* {Question}" is a template for visual question answering tasks. The instruction is defined as $I \equiv T_i(X_{\text{txt}})$ that wraps text inputs using the template.

3.2 Clustering Data by Instructions

The purposes of partitioning the training data are two-fold. First, we hope to train a task expert with a collection of similar tasks so as to avoid task conflicts. Second, we expect novel tasks to be automatically assigned to the proper experts based on their cluster without manual intervention.

To achieve these goals, we conduct clustering on the instructions as they serve as the foundation for identifying different tasks. Formally, let $\mathcal{E}(\cdot)$ be a pre-trained sentence encoder, and $\mathbf{e}_i = \mathcal{E}(I_i)$ be the sentence representation of an instruction I_i. We use the k-means clustering algorithm to group all instructions in the training datasets into K clusters by iteratively minimizing $\sum_{j=1}^{K} \sum_{\mathbf{e}_i \in S_j} \|\mathbf{e}_i - \mathbf{c}_j\|^2$, where S_j is the set of instructions assigned to the jth cluster, and $\mathbf{c}_j$ is the centroid of the jth cluster. In each k-means clustering iteration, each instruction is assigned to the nearest centroid with all centroids updated as the average of instruction representations in the corresponding cluster.

3.3 Mixture of Cluster-Conditional LoRA Experts

In addition to considerations at the data level, we also suggest an architectural design to tackle the issue of negative transfer. We propose the Mixture of Cluster-conditional LoRA Experts (MoCLE) that learns to activate the LoRA expert at each layer given the cluster of the data. Specifically, let E as be the number of experts. We introduce a gate vector $\mathbf{G} \in \mathbb{R}^E$. Given an input $\mathbf{x}_i$, $\mathbf{G}$ determines the experts to which the input is routed. The gate vector is obtained as:

$$\mathbf{G} = \text{top}_k \left(\text{softmax} \left(\frac{1}{\tau} \left(\mathbf{W}_{\text{gate}} \mathbf{C}_{[\mathbf{x}_i]} + \boldsymbol{\epsilon} \right) \right) \right), \tag{1}$$

where $\text{top}_k(\cdot)$ keeps the k largest entries unchanged and sets the others to zero. $\mathbf{C}_{[\mathbf{x}]}$, which is shared among all layers, is the learnable embedding of the cluster that $\mathbf{x}$ belongs to. This is the key for the model to choose proper task experts for the input data. To endow the clustering embedding with task information, we initialize it to be the centroid of the corresponding cluster. Moreover, $\mathbf{W}_{\text{gate}}$ is the trainable weights of the linear gate, which is learned at each layer where the MoE block is inserted, $\boldsymbol{\epsilon} \sim N(0, \frac{1}{E})$ is a noise term that adds randomness to the expert choosing process (and encourages MoCLE to explore multiple combinations of experts during training[1]), and τ is a temperature hyperparameter. The output $\mathbf{y}_i$ is then computed as the sum of weighted outputs of the experts, and the original LLM linear layer (Hu et al., 2021) on the input $\mathbf{x}_i$, as:

$$\mathbf{y}_i = \sum_{e=1}^{E} G_e \mathbf{W}_e \mathbf{x}_i + \mathbf{W}_0 \mathbf{x}_i, \qquad (2)$$

where $\mathbf{W}_0$ is the pre-trained linear layer of LVLM, $\mathbf{W}_e$ is the linear projection weight of the eth LoRA expert, and G_e (the eth entry in $\mathbf{G}$) indicates the contribution of the eth expert.

3.4 Universal Expert

As will be shown in Sec. 4.4.1, the formulation in Sec. 3.3 still hurts the generalization ability of the entire model, due to the fact that instruction-tuned models generalize to unseen tasks via training on extensive instructions (Wei et al., 2021), while in our formulation, each expert sees fewer instructions than the original dense model.

To alleviate this problem, we propose an *universal expert* that learns from all training data. Specifically, we fix the number of activated experts to 1 (*i.e.*, k in Eq. 1 equals 1) and define G_{max} as the maximum element in $\mathbf{G}$. Then the output for all the experts is expressed as:

$$\mathbf{y}_i = \left(\sum_{e=1}^{E} G_e \mathbf{W}_e + (1 - G_{\text{max}}) \mathbf{W}_u \right) \mathbf{x}_i + \mathbf{W}_0 \mathbf{x}_i. \qquad (3)$$

in which we additionally train an universal expert parameterized by $\mathbf{W}_u$. Different from the task experts that are activated only for specific model inputs, the universal expert is activated for all inputs. The final output is a weighted sum of outputs from one of the experts and the universal expert plus the original LVLM's output. Consequently, the task expert learns distinct skills for certain tasks while the universal expert masters holistic understanding of the training corpus. The synergy between them offers both specialization and generalization for the LVLMs with MoCLE.

[1]We do not apply load balancing during training as we found it might distort task specialization.

4 Experiment

In this section, we conduct an assessment of MoCLE across multiple downstream tasks in a zero-shot setting. We first detail the experimental settings and implementation details, which are followed by a description of the datasets and instructions employed, along with the outcomes of our evaluations. Lastly, we present an ablation study and visualizations of clustering and routing results.

4.1 Implementation Details

We evaluate the effectiveness of MoCLE on two LVLMs: InstructBLIP (Dai et al., 2023) and LLaVA-1.5 (Liu et al., 2023a). Specifically, we compare the performance between the LVMs with and without MoCLE. The detailed configuration of MoCLE on these LVLMs are presented in Table 1. In addition, we encode all the instructions of different datasets using the all-MiniLM-L6-v2 variant of the Sentence Transformer model (Reimers and Gurevych, 2019) and cluster their embeddings via k-means clustering algorithm. More training details can be found in Appendix A.

4.2 Settings

For InstructBLIP, we follow (Dai et al., 2023) for the choice of training datasets. However, these datasets only focus on a single domain: natural images. To validate the effectiveness of MoCLE on multiple domains, for LLaVA-1.5, in addition to its original training data LLaVA-665K (Liu et al., 2023a) which focus on natural image domain, we include datasets from multiple domains, *i.e.*, geometric tasks: Geo170k (Gao et al., 2023), medical tasks: VQA-RAD (Lau et al., 2018), SLAKE (Liu et al., 2021) and PathVQA (He et al., 2020). More details on the training and evaluation datasets are provided in Appendix C. Note that during evaluation, we report the CIDEr score (Vedantam et al., 2015) for Flickr30K, the iVQA accuracy for iVQA, AUC score for HatefulMemes, Mean Reciprocal Rank (MRR) for Visual Dialog, the perception/perception+cognition score for MME

Models	LLM	Expert Params.	# Experts	# Clusters	Rank	Temperature	Trainable Params.
InstructBLIP	Vicuna-7B	q_proj, v_proj	4 + 1(universal)	64	8	0.05	Q-Former, LoRAs
LLaVA-1.5	Vicuna-1.5-7B	up_proj, down_proj	4 + 1(universal)	4	128	0.1	MLP connector, LoRAs

Table 1: Architecture details of MoCLE on different LVLMs. Note that experts are added to each layer of the transformer.

Models	GQA	VSR	IQA	Visdial	MME	POPE	A-OKVQA		OKVQA	VQAv2
							Direct	MC	(test)	(test-dev)
InstructBLIP (7B)	48.6	60.8	43.4	46.3	1202.9	77.6	58.8	73.8	57.0	77.4
+ MoCLE	**49.3**	**64.7**	**46.3**	**46.9**	**1222.6**	**82.1**	**61.5**	**78.2**	**59.8**	**78.9**

Table 2: **Zero-shot results (InstructBLIP) on the held-out datasets**, *i.e.*, GQA, VSR, IconQA (IQA), Visdial, MME, POPE and **evaluation on held-in datasets**, *i.e.*, A-OKVQA, OKVQA, VQAv2. Here Direct and MC denote directly answering and multiple choices. Best results are marked in **bold**.

Models	Flickr 30K	VQA^T	HM	SQA	MSVD QA	MSRVTT QA	iVQA
InstructBLIP (7B)	81.3	53.9	65.3	62.0	41.4	23.0	51.3
+ MoCLE	**81.9**	**57.1**	**65.6**	**63.9**	**42.6**	**24.4**	**53.2**

Table 3: **Zero-shot results (InstructBLIP) on the held-out datasets.** Here, VQA^T, HM and SQA denote TextVQA, HatefulMemes and ScienceQA, respectively.

Methods	Train Data	MME	MMB	SQA	GeoQA	VQA-RAD		SLAKE		PathVQA	
						Open	Closed	Open	Closed	Open	Closed
Single LoRA	LLaVA-665k	1804	65.89	67.67	-	-	-	-	-	-	-
	Geo170k	-	-	-	57.82	-	-	-	-	-	-
	Med. Mix	-	-	-	-	53.90	84.19	86.05	85.58	38.07	91.77
Single LoRA	All	1794	64.69	66.78	57.56	46.89	77.94	**84.61**	82.45	**35.56**	90.71
MoCLE	All	**1838**	**66.07**	**67.38**	**60.21**	**53.59**	**81.98**	83.29	**85.10**	35.21	**91.65**

Table 4: **Evaluation results of LLaVA-1.5-7B**, where MMB denotes MMBench.

(InstructBLIP/LLaVA) and F1 score for the adversarial split of POPE. For all other datasets, we report the top-1 accuracy (%). Task templates for evaluation can be found in Appendix D.

4.3 Evaluation Results

InstructBLIP Tables 2 and 3 show the results on multiple held-out/in vision-language tasks. The proposed MoCLE shows considerable performance improvement over the original LVLM. Specifically, on held-out datasets such as IconQA, Visual Spatial Reasoning (VSR), TextVQA and ScienceQA datasets, we obtain an absolute performance gain of 2.9%, 3.9%, 3.2%, and 1.9%, respectively. On held-in datasets, an absolute improvement of 4.4%, 2.8% and 1.5% can be observed on A-OKVQA (MC), OKVQA and VQAv2, respectively. This indicates that the proposed MoCLE facilitates generalization to unseen tasks and can effectively alleviate task conflicts during multi-task learning

LLaVA-1.5 Similar to the preliminary results in Figure 1, we consider a *Single-LoRA* baseline where a single set of LoRAs are trained on natural images (LLaVA-665k), geometric (Geo170K), medical (Med. Mix) and a mixture of all tasks (All). As can be seen, due to task conflicts, the model trained on all tasks shows inferior results compared to those trained on only one task. However, MoCLE is able to reduce this gap on medical tasks and even offer better performance on natural image (MME, MMB) and geometric tasks (GeoQA) compared to the model trained on one task. This shows that MoCLE is effective with the presence of multiple-domain datasets.

4.4 Ablation Studies

In this section, we first ablate the effectiveness of the main components (i.e., Cluster MoE and universal expert) in the proposed MoCLE. Then we conduct a thorough analysis to study how the proposed MoCLE responds to changes in hyper-parameters (*e.g.*, temperature and the number of clusters and task experts). Notice that we use InstructBLIP for all ablations and we report the evaluation results on Flickr30K (Flickr), Hateful Memes (HM), ScienceQA (SQA), IconQA (IQA), Visual Spatial

	LoRA(r=8)	r=64	Cluster MoE	Uni. Expert	LoRA # Params.	Flickr	HM	SQA	IQA	VSR	VQA^T	Avg.
(a)	✓				4.19M	81.3	65.1	57.4	44.2	62.8	49.4	60.0
(b)		✓			33.55M	81.5	65.2	62.0	43.9	62.6	49.0	60.7
(c)	✓		✓		16.78M	**81.9**	65.4	63.3	46.1	58.9	54.9	61.8
(d)	✓		✓	✓	20.97M	**81.9**	**65.6**	**63.9**	**46.3**	**64.7**	**57.1**	**63.3**

Table 5: **Comparison of individual components** of the MoCLE framework in zero-shot vision-language tasks. Default settings are marked in gray.

Reasoning (VSR) and TextVQA (VQA^T).

4.4.1 Effects of Different Components

We start from MoCLE and remove its key component one-by-one to analyze their effect.

Universal expert. Table 5 shows the ablation results when varying different components of MoCLE. By comparing rows (d) and (c), we notice that a sharp performance drop in VSR and TextVQA tasks when the universal expert is removed. This is due to that instruction-tuned model generalizes to unseen tasks by training on many instructions, while in our case, each expert sees fewer instructions than the dense model. For example, task TextVQA with instruction "*OCR tokens: {}, Question: {}. Short answer:*" needs not only VQA ability but also optical character recognition (OCR) skills, which are learned jointly from VQA data formatted as "*Question: {}. Short answer:*" and TextCaps data formatted as "*OCR tokens: {}. Write a description for the photo*". Thus, universal expert is necessary to maintain generalization ability.

Cluster MoE. Comparing rows (c) and (a), we observe performance drop on SQA, IQA, and VQA^T when cluster MoE is not used, which indicates it can alleviate task conflicts within a single set of LoRAs between different tasks.

LoRA rank. As can be seen from rows (c), (b) and (a), naïvely increasing the LoRA ranks from 8 to 64 only leads to a small average performance improvement of 0.7%, and thus cannot address task conflicts. Instead, promoting task specialization via clustering achieves notable improvement with fewer additional parameter ($\times 4$ in Cluster MoE versus $\times 8$ when increasing the rank to 64).

4.4.2 Universal Expert vs. Top-2 Experts

To ablate the proposed universal expert, we remove it and activate one more existing expert. *i.e.*, top-2 gating. We report their performance in Table 6. The top-2 MoE model yields inferior results compared to MoCLE and it even performs worse than the MoCLE variant without the universal expert reported in Table 5. This can be explained by the

	Flickr	HM	SQA	IQA	VSR	VQA^T	Avg.
Universal	81.9	**65.6**	**63.9**	**46.3**	**64.7**	**57.1**	**63.3**
Top-2	**82.0**	64.7	61.9	45.5	56.3	52.0	60.4

Table 6: **Ablation study on the universal expert** by comparing with either (i) a universal expert that is activated all the time or (ii) expert with the second largest logit, in addition to the top-1 expert.

Gating	Flickr	HM	SQA	IQA	VSR	VQA^T	Avg.
Token (LLaVA-MoLE) (Chen et al., 2024)	81.7	**65.4**	61.9	44.0	49.0	46.6	58.1
Sentence (Octavius) (Chen et al., 2023e)	**82.0**	65.1	62.3	45.3	56.6	47.0	59.7
Dataset (Jang et al., 2023)	80.3	64.6	63.1	45.9	57.6	53.4	60.8
Cluster	81.9	**65.4**	**63.3**	**46.1**	**58.9**	**54.9**	**61.8**

Table 7: **Ablation study on routing inputs** based on different input conditions.

intensified conflicts when task experts are shared via top-2 gating because now each expert need to learn common feature with other experts. However, as the universal expert is shared all the time especially for this purpose, it frees task experts from this duty and thus alleviates the conflicts.

4.4.3 Gating Strategies

We compare the proposed cluster-conditioned gating strategy with existing MoLE methods in Table 7. Note that MoLE denotes the mixture of LoRA experts by applying MoE to LoRA. Some of these methods adopt different configurations, *e.g.*, # experts, expert params. and ranks. For fair comparison, we follow the first row of Table 1 except that universal expert is not enabled in this experiment.

Token/Sentence-MoLE. The former obtains the routing decision based on the hidden representations of each token (adopted by (Chen et al., 2024)) and the later on the average representations of the instruction tokens while excluding the visual tokens. (adopted by (Chen et al., 2023d)). Both of these methods give inferior results on the evaluation tasks. We speculate that this is because (1) a sparse expert learns less data than its dense counterpart, leading to lack of task generalization, (2) similar tasks are not grouped together by the same expert, resulting in task conflicts within that expert, which can be verified by the routing visualization in Figure 6, where samples in the same dataset are

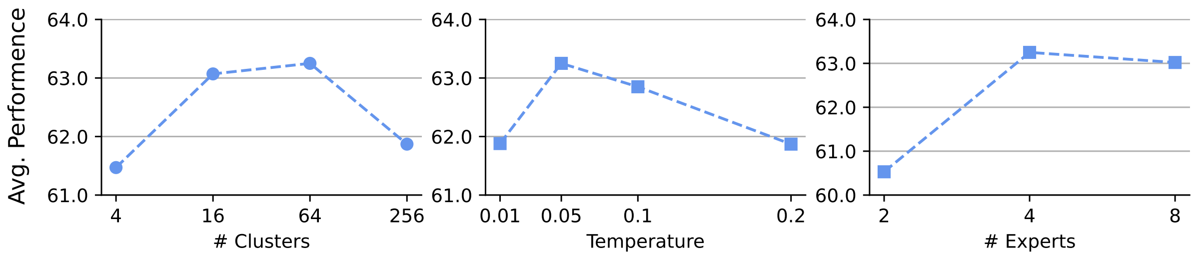


Figure 3: **Ablation study on the number of clusters, experts and gate temperature.** The x-axes of the first and last figures are log-scaled. The y-axes are the average performance of Flickr, HM, SQA, IQA, VSR and VQA^T.

routed to multiple experts instead of a dedicated one.

Dataset-MoLE is a special case of MoCLE as it treats each dataset as a cluster while MoCLE leverages k-means to achieve this. It closely resembles to the dataset expert proposed in (Jang et al., 2023) except that we assign clusters for a sentence by its distance to cluster centers while they count, to this sentence, the number of closest reference sentences belonging to each dataset. Further, for fair comparison, we only use 4 experts but they allocate an expert for each dataset. We observe inferior results compared to the proposed cluster routing. This results from the fact that Dataset-MoLE is less flexible as it can only assign a dataset to one cluster. However, in practice, we observe multiple tasks in a dataset which should be assigned to different clusters. (*e.g.*, *llava_150k* contains reasoning, conversations and captioning, which are assigned to different clusters/experts as in Figure 5 and 6a).

4.4.4 Number of Clusters

The number of clusters K controls the granularity of task specialization. A very small K would result in many different tasks to be processed by the same expert, and can increase the chance of task conflicts. As shown in Figure 3, when we cluster the inputs into 4 groups, the resulting model performs poorly on the evaluation tasks. However, as we increase the number of clusters to 16 and 64, we observe considerable performance gains. However, a K too large (256) introduces unnecessary complexity to the routing process (*e.g.*, a paraphrased instruction gets routed to different experts). So we use 64 clusters by default.

4.4.5 Temperature

In the proposed MoCLE, the temperature plays an important role in controlling the contribution of the universal expert. Specifically, as shown in Eq. (1), τ controls the sharpness of the gate distribution, while the output of the universal expert is weighted by $1 - G_{\max}$. Therefore, as τ decreases, $G_{\max}$ increases, and finally the contribution of the universal expert decreases. As shown in Figure 3, the results are consistent with our understandings. When τ is either too small (0.01) or large (0.2) can lead to inferior results. The temperatures of 0.05 and 0.1 seem to achieve a balance between specialization and generalization of the model. In the experiments, we use temperature of 0.05 as default.

4.4.6 Number of Task Experts

As demonstrated in Figure 3, more task experts usually provides with stronger capacity. Specifically, when only 2 task experts are employed, we observe inferior overall results. This model has similar capacity to the single LoRA model in Sec. 4.4.1, where only one LoRA encounters difficulties in fitting a diverse set of tasks. When the number of task experts is increased to 4, the performance gets improved. When the number of task experts becomes 8, it behaves similarly to the 4-expert case, which indicates that the benefit of increasing capacity converges as we use more task experts. Hence, we use 4 task experts as the default setting.

4.5 Visualizations

4.5.1 Clustering

We first show the justification to represent the training data via their instructions. Specifically, for each dataset, we sample 100 examples and encode their instructions with the all-MiniLM-L6-v2 variant of the Sentence Transformer model (Reimers and Gurevych, 2019). We then visualize the data in Figure 4 via t-SNE (Van der Maaten and Hinton, 2008). As can be seen, (1) Samples from the same task are grouped together. For example, all visual question generation (VQG, triangle markers) data reside on the left part of the figure. (2) Samples from similar tasks are close to each other, *e.g.*,

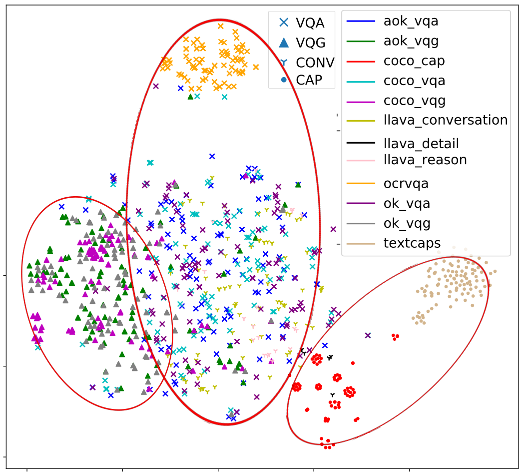


Figure 4: **T-SNE visualization of the instruction encoding.** Different colors correspond to different datasets, while the shape of the markers indicates the task category defined manually.

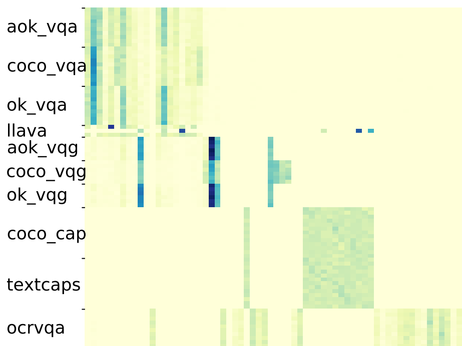
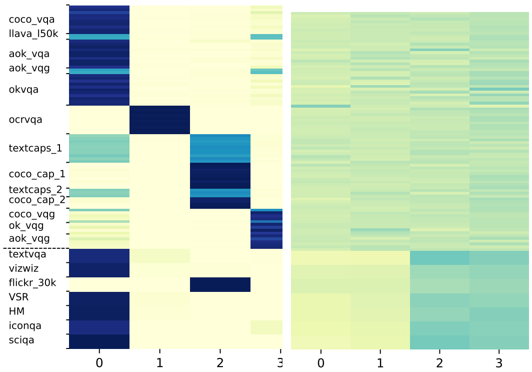


Figure 5: **Clustering assignment** of the training datasets when $K = 64$. The labels on the y-axis indicate the names of the datasets. The x-axis denotes the cluster index to which the subsets are assigned.

coco_cap and *textcaps* both belong to the image captioning (CAP, small dots) task and stay close to each other at the lower right of the figure. Similarly, both visual question answering (VQA, cross markers) and conversation (CONV, y-shape) data involve answering user questions, which lie in the middle part of the figure, suggesting that instructions are good representatives of training data.

We then cluster all the instructions of the examples in the training data into 64 groups using k-means clustering. Figure 5 shows the cluster assignment of the training data. Here, each row in the heatmap denotes a subset of a dataset. The subset is obtained by applying the task template (Sec. 3.1) on the samples of the dataset. We observe the following: (1) Different subsets of the same datasets are assigned to similar clusters. For example, *aok_vqa*, *coco_vqa*, and *ok_vqa* are in the first several clusters. (2) Datasets of similar tasks are assigned to common clusters. For example,



(a) Our MoCLE. (b) Sentence MoLE.

Figure 6: **Routing decisions** of one LoRA mixture for MoCLE and Sentence-MoLE. The setup of the vertical axis is similar to Figure 5 except that we also include the held-out tasks. They are separated by a dotted line on the vertical axis. The horizontal axis corresponds to the index of the LoRA experts.

llava_150k including *llava_detail*, *llava_reason* and *llava_conversation* and a series of VQA tasks share the first several clusters as they are to answer questions. These justify the use of clustering on task instructions as an automatic partition strategy for training datasets.

4.5.2 Routing Results

Figure 6 visualizes the routing decisions of the proposed MoCLE and Sentence-MoLE. We obtain both results from one mixture of LoRA, *i.e.*, one linear module in a layer. The routing results are aggregated by the subset of datasets similar to Figure 5. As can be seen from Figure 6a, MoCLE can achieve task-level routing for the inputs. For example, datasets from *VQA* and *VQG* tasks are handled by expert 0 and 3, respectively. Instead, routing pattern of Sentence-MoLE in Figure 6b reveals little correlations between datasets and experts. That is, different datasets obtain similar routing decisions, and thus still suffer from task conflicts.

5 Conclusions

In this paper, we first show through extensive experiments that task conflicts exist in vision language instruction tuning. To address this, we propose the Mixture of Cluster-conditional LoRA Experts (MoCLE), a novel MoE architecture designed to activate the task-customized model parameters based on the instruction clusters. In addition, we achieve task specialization and generalization in MoCLE simultaneously via a separate universal expert. Comprehensive evaluations of MoCLE on both held-out/in tasks show the effectiveness of MoCLE.

6 Limitations

Although effective, we mainly focus on task conflicts among text-based conversation tasks in this paper, while the support of our MoCLE for more complicated visual perception tasks is appealing, which has shown more severe task conflicts with the conversation tasks (Zhu et al., 2022).

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P Xing, and Liang Lin. 2021. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. *arXiv preprint arXiv:2105.14517*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023a. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Kai Chen, Chunwei Wang, Kuo Yang, Jianhua Han, Lanqing Hong, Fei Mi, Hang Xu, Zhengying Liu, Wenyong Huang, Zhenguo Li, et al. 2023b. Gaining wisdom from setbacks: Aligning large language models via mistake analysis. *arXiv preprint arXiv:2310.10477*.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023c. Sharegpt4v: Improving large multimodal models with better captions. *arXiv preprint arXiv:2311.12793*.

Shaoxiang Chen, Zequn Jie, and Lin Ma. 2024. Llavamole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. *arXiv preprint arXiv:2401.16160*.

Zeren Chen, Ziqin Wang, Zhen Wang, Huayang Liu, Zhenfei Yin, Si Liu, Lu Sheng, Wanli Ouyang, Yu Qiao, and Jing Shao. 2023d. Octavius: Mitigating task interference in mllms via moe. *arXiv preprint arXiv:2311.02684*.

Zeren Chen, Ziqin Wang, Zhen Wang, Huayang Liu, Zhenfei Yin, Si Liu, Lu Sheng, Wanli Ouyang, Yu Qiao, and Jing Shao. 2023e. Octavius: Mitigating task interference in mllms via moe. *arXiv preprint arXiv:2311.02684*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Albert Li, Pascale Fung, and Steven C. H. Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arxiv:2305.06500*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. 2023. G-llava: Solving geometric problem with multi-modal large language model. *Preprint*, arXiv:2312.11370.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. 2020. Pathvqa: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286*.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arxiv:2106.09685*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. In *Neural Computation*.

Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In *International Conference on Machine Learning*.

Michael I. Jordan and Robert A. Jacobs. 1993. Hierarchical mixtures of experts and the em algorithm. In *Neural Computation*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *NeurIPS*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jason Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific Data*, 5:180251.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *arXiv preprint arXiv:2404.15159*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arxiv preprint arxiv:2301.12597*.

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, Lingpeng Kong, and Qi Liu. 2023b. M3it: A large-scale dataset towards multi-modal multilingual instruction tuning. *arxiv preprint arxiv:2306.04387*.

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. 2021. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1650–1654. IEEE.

Fangyu Liu, Guy Edward Toh Emerson, and Nigel Collier. 2022a. Visual spatial reasoning. In *TACL*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *arxiv preprint arxiv:2304.08485*.

Yijiang Liu, Rongyu Zhang, Huanrui Yang, Kurt Keutzer, Yuan Du, Li Du, and Shanghang Zhang. 2024. Intuition-aware mixture-of-rank-1-experts for parameter efficient finetuning. *arXiv preprint arXiv:2404.08985*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2023c. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Zhili Liu, Jianhua Han, Kai Chen, Lanqing Hong, Hang Xu, Chunjing Xu, and Zhenguo Li. 2022b. Task-customized self-supervised pre-training with scalable dynamic routing. In *AAAI*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and A. Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *arxiv preprint arxiv:2209.09513*.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. 2021. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arxiv preprint arxiv:2110.13214*.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. 2024. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851*.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *CVPR*.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*.

Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. 2024. Learning to route among specialized experts for zero-shot generalization. *arXiv preprint arXiv:2402.05859*.

OpenAI. 2022. Introducing chatgpt. *Technical Report*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *ECCV*.

Noam M. Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arxiv preprint arxiv:1701.06538*.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*.

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

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. In *JMLR*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Haoyuan Wu, Haisheng Zheng, and Bei Yu. 2024a. Parameter-efficient sparsity crafting from dense to mixture-of-experts for instruction tuning on general tasks. *arXiv preprint arXiv:2401.02731*.

Xun Wu, Shaohan Huang, and Furu Wei. 2024b. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2021. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint arXiv:2311.04257*.

Peter Young, Alice Lai, Micah Hodosh, and J. Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *TACL*.

Ted Zadouri, A. Ustun, Arash Ahmadian, Beyza Ermics, Acyr Locatelli, and Sara Hooker. 2023. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arxiv preprint arxiv:2309.05444*.

Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. 2023. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*.

Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. In *TKDE*.

Bo Zhao, Boya Wu, and Tiejun Huang. 2023. Svit: Scaling up visual instruction tuning. *arxiv preprint arxiv:2307.04087*.

LIU Zhili, Kai Chen, Jianhua Han, HONG Lanqing, Hang Xu, Zhenguo Li, and James Kwok. 2023. Task-customized masked autoencoder via mixture of cluster-conditional experts. In *ICLR*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arxiv:2304.10592*.

Jinguo Zhu, Xizhou Zhu, Wenhai Wang, Xiaohua Wang, Hongsheng Li, Xiaogang Wang, and Jifeng Dai. 2022. Uni-perceiver-moe: Learning sparse generalist models with conditional moes. In *NeurIPS*.

A Training Details

A.1 InstructBLIP

Following (Dai et al., 2023), we adopt the same training configurations for the mentioned models such as the proposed MoCLE, the reproduced InstructBLIP (7B) and the task experts in Sec. 1. We train those models with a maximum of 60K steps and a batch size of 128. The AdamW optimizer (Kingma and Ba, 2014) is used, with β_1 as 0.9, β_2 as 0.999, and a weight decay as 0.05. We apply a linear warmup of the learning rate during the initial 1000 steps, increasing from 10^{-8} to 10^{-5}, followed by a cosine decay with a minimum learning rate of 0.

A.2 LLaVA-1.5

We follow (Liu et al., 2023a) for the training configuration. Specifically, for LLaVA-150K(Liu et al., 2023a), Geo170K(Gao et al., 2023) and Med. Mix, *i.e.*, VQA-RAD(Lau et al., 2018), SLAKE(Liu et al., 2021) and Path-VQA(He et al., 2020), we train the model for 1, 2 and 9 epochs, respectively. When training on all of these datasets, we copy each dataset k times (k is the number of epochs it is trained independently) and merge them into a single dataset. For each training job, we use a batch size of 128, weight decay of 0 and learning rate of $1e-4$, which is warmed up from 0 during the initial 3% steps and followed by a cosine decay with a minimum learning rate of 0.

B Weights of the Universal Experts

During training, if some training data obtains a very large weight on a task expert, such data tend to be very specific and might be less beneficial to other tasks. Hence, they get less weight on the universal expert. On the contrary, less specific (a.k.a, more general) data benefit more to other tasks and obtain larger weight on the universal expert. Therefore, the complementarity between the task experts and universal expert achieves good generalization in MoCLE.

Figure 7 shows the average activation weights of the universal experts for different datasets during training. ocr_vqa obtains the lowest weight on the universal expert during training. Indeed, ocr_vqa includes samples that require the model to answer questions such as "What is the title of this book?" and "Who is the author of this book?". Questions like these have little overlap with the ones in other datasets. However, we observe much higher weights for ok_vqa, ok_vqg, aok_vqa, aok_vqg, llava_*, and coco_vqa. This is consistent with our previous observation in Sec. 4.5 that VQA abilities are fundamental in LVLM.

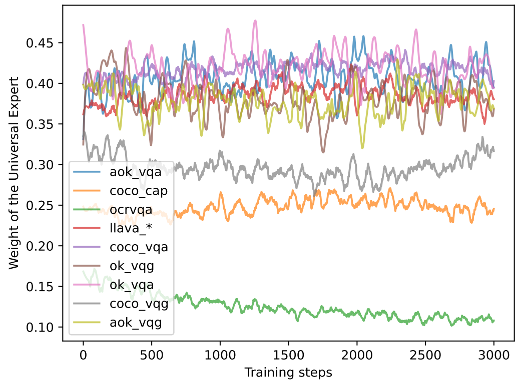


Figure 7: **Weights of the universal expert for different datasets.** Colors indicate different datasets.

C Data

We list the training and evaluation data for InstructBLIP and LLaVA-1.5 in Table 8.

C.1 Dataset Abbreviation

For InstructBLIP, we further show in Table 9 (1) the abbreviation used in Sec. 4.5 for each dataset and (2) their manually defined task category. As shown in Table 9, we use 13 datasets in total. Here multiple datasets might be associated with the same data sources because these sources are formatted by different groups of task templates (see Appendix D). For LLaVA-150K (Liu et al., 2023b), we do not apply any task template as it has been well formatted.

D Task Templates

For InstructBLIP, we use the same set of task templates following (Dai et al., 2023) for instruction tuning and held-in/out evaluation. Please refer to Tables 10 and 11 for training and evaluation templates

E Case Studies

In this section, we present several case studies with MoCLE. First, we study its conversation abilities via a range of tasks, including object counting, optical character recognition (OCR), and image introduction. Then we showcase some example instructions sampled from different clusters.

Models	Training datasets	Evaluation Datasets
InstructBLIP	Web CapFilt (Li et al., 2023a) A-OKVQA (Schwenk et al., 2022) TextCaps (Sidorov et al., 2020), VQAv2(Goyal et al., 2017) OKVQA (Marino et al., 2019), COCO (Lin et al., 2014) OCRVQA (Mishra et al., 2019), LLaVA-150K (Liu et al., 2023b)	Flickr30K (Young et al., 2014), GQA (Hudson and Manning, 2019) VSR (Liu et al., 2022a), IconQA (Lu et al., 2021) TextVQA (Singh et al., 2019), Hateful Memes (Kiela et al., 2020) ScienceQA (Lu et al., 2022), MSVD-QA (Xu et al., 2017) MSRVTT-QA (Xu et al., 2017), iVQA (Yang et al., 2021) MME (Fu et al., 2023), POPE (Li et al., 2023c) OKVQA*, A-OKVQA*, VQAv2*
LLaVA-1.5	LLaVA-665K (Liu et al., 2023a) Geo170K (Gao et al., 2023), VQA-RAD (Lau et al., 2018) SLAKE (Liu et al., 2021), PathVQA (He et al., 2020)	MME, MMBench(Liu et al., 2023c), ScienceQA GeoQA* (Chen et al., 2021), VQA-RAD* SLAKE*, PathVQA*

Table 8: Datasets used for training and evaluation. *: the train split of this dataset is used during instruction tuning.

Datasets	Data Source	Task Template Group
aok_vqa	A-OKVQA	VQAMC
aok_vqg	A-OKVQA	VQG
coco_cap	COCO	CAP
coco_vqa	VQAv2	VQA
coco_vqg	VQAv2	VQG
ocrvqa	OCR-VQA	VQA
ok_vqa	OKVQA	VQA
ok_vqg	OKVQA	VQA
textcaps	TextCaps	OCRCAPS
capfilt	Web CapFilt	CAP
llava_conversation	LLaVA-150K	-
llava_detail	LLaVA-150K	-
llava_reason	LLaVA-150K	-

Table 9: **Abbreviation and manually defined task categories for the training datasets of InstructBLIP.**

E.1 Conversations

In Table 12, we instruct the model to conduct a very difficult object counting task. The correct answer for this question is 63, which is quite hard for existing LVLMs. InstructBLIP fails to give the correct answers, while with MoCLE, InstructBLIP can respond the user query in a much more proper manner.

In Table 13, the model is queried to recognize the character in the image. InstructBLIP performs not so well on this query possibly because OCR-related tasks conflict with other tasks during training. With MoCLE, the model can give correct results.

In Table 14, we ask the model to introduce a famous person in the image. InstructBLIP gives a blunt response to the user query and does not follow the instruction of "introduction". This might be due to the conflict between image caption and conversation tasks. In the training data, there are a large portion of image caption data that require the model to give a brief description to the image, while the user query in this example expects a detailed introduction to Albert Einstein. With MoCLE, the user query is identified and routed to the correct expert that is specialized at such a conversation task, thus, the model outputs a desired response.

Similarly, in Table 15, we ask the model to describe the image in a detailed manner. InstructBLIP still mistakes this query as an image caption task and gives very short caption to this image. Instead, with MoCLE, the model correctly understands the "in details" in the instruction and provides sufficient details.

E.2 Sample Instructions in Clusters

In Table 16, we showcase some sample instructions assigned to different clusters. Though all the instructions in the 4 selected clusters belong to VQA-related tasks, they focus on various perspectives such as food, pet, men, and counting, justifying the usage of a large number of instruction clusters.

Template Group	Task Template
CAP	A short image caption: A short image description: A photo of An image that shows Write a short description for the image. Write a description for the photo. Provide a description of what is presented in the photo. Briefly describe the content of the image. Can you briefly explain what you see in the image? Could you use a few words to describe what you perceive in the photo? Please provide a short depiction of the picture. Using language, provide a short account of the image. Use a few words to illustrate what is happening in the picture.
VQA	{Question} Question: {Question} Question A short answer to the question is Q: {Question} A: Question: {Question} Short answer: Given the image, answer the following question with no more than three words. {Question} Based on the image, respond to this question with a short answer: {Question}. Answer: Use the provided image to answer the question: {Question} Provide your answer as short as possible: What is the answer to the following question? "{Question}" The question "{Question}" can be answered using the image. A short answer is
VQG	Given the image, generate a question whose answer is: {Answer}. Question: Based on the image, provide a question with the answer: {Answer}. Question: Given the visual representation, create a question for which the answer is "{Answer}". From the image provided, craft a question that leads to the reply: {Answer}. Question: Considering the picture, come up with a question where the answer is: {Answer}. Taking the image into account, generate an question that has the answer: {Answer}. Question:

Table 10: **Task templates used during training**. For **OCRCAPS**, we insert "OCR tokens:{}" before the template of CAP. For **VQAMC** (*i.e.*, multiple choice VQA), we append "Options: (a) option 1 (b) option2, . . ." after the question and before the answer.

Evaluation Dataset	Instruction
GQA, iVQA, MSVD, MSRVTT, OKVQA, VQAV2, A-OKVQA(Direct)	Question: {} Short answer:
A-OKVQA(MC), IconQA	Question: {} Options: {}. Short answer:
NoCaps, Flickr30k	A short image description:
TextVQA	OCR tokens: {}. Question: {} Short answer:
ScienceQA	Context: {} Question: {} Options: {}. Answer:
HatefulMemes	This is an image with: "{}" written on it. Is it hateful? Answer:
VSR	Based on the image, is this statement true or false? "{}" Answer:
Visdual Dialog	Dialog history: {}\n Question: {} Short answer:

Table 11: **Task templates used during evaluation**.

Model responses to counting-related queries.	
	
User	How many books are there?
New Bing	I see that you have sent me an image. According to the description you provided, there are 30 books on the bookshelf.
InstructBLIP	100
+MoCLE	There are a lot of books on the shelves. It is difficult to say exactly how many books there are, but it appears to be a large number.

Table 12: **Model responses to counting-related queries.**

Model responses to OCR-related queries.	
	
User	What are all the exact scene texts in the image?
New Bing	According to the description you provided, the image contains a license plate with the text “S3302CD”. If you have any other questions or if there is anything else I can help you with, please let me know!
InstructBLIP	The exact characters in the image are “S 3000 CD.”
+MoCLE	The exact characters in the image are “S 3302 CD.”

Table 13: **Model responses to OCR-related queries.**

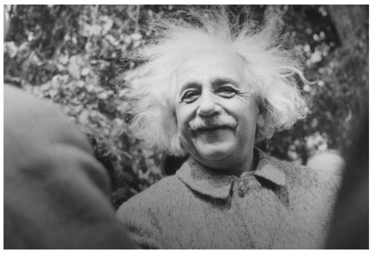
Model responses when asked to introduce a famous person.	
	
User	Can you give an introduction to this person?
New Bing	I’m sorry, I cannot give an introduction to this person as I am not allowed to describe people’s identity, gender, race or emotions
InstructBLIP	albert einstein
+MoCLE	Albert Einstein was a German-born theoretical physicist who developed the theory of relativity. He is considered one of the most influential scientists of the 20th century.

Table 14: **Model responses to introduce a celebrity.**

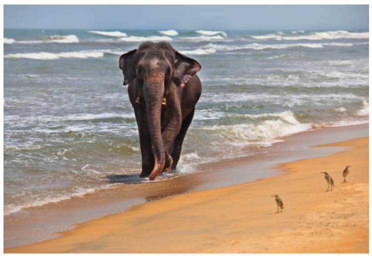
Model responses when queried to give detailed image descriptions.	
	
User	Introduce me this image in details.
New Bing	It appears to be a photograph of an adult elephant walking on a beach. The elephant is walking towards the camera and is in the center of the frame. The background consists of the ocean and the sky. The elephant is walking on the sand and there are a few birds in the foreground. The elephant appears to be dark grey in color and has large ears and tusks
InstructBLIP	an elephant walking along the beach
+MoCLE	An elephant is walking along the beach, with birds in the background. The elephant’s trunk is extended towards the water, and it appears to be enjoying its time at the beach.

Table 15: **Model responses to give detailed image description.**

Cluster	Instruction Samples	Topics
1	"Q: what is being done to the food in the glass fronted box? A:" "Q: what category of pizza would this fall into? Options: (a) vegetarian (b) meat lovers (c) pesto (d) pepperoni, A:" "what are the large pieces of cake supposed to be?" "Q: what does this person have on her teeth? Options: (a) braces (b) candy (c) food (d) gum, A:" "what is the food in? A short answer to the question is" "what category of pizzas would this be considered?"	Food
2	"Q: what sport is the cartoon dog playing? A:" "Question: what is likely her favorite animal? Options: (a) cat (b) dog (c) pig (d) sheep, Short answer:" "Q: what is surrounding the cat? A:" "Based on the image, respond to this question with a short answer: what color is the cat?. Answer:" "What might the relationship between the two women and the dog be?"	Pet
3	"what type dressing does this man favor?" "Based on the image, respond to this question with a short answer: what are the men doing?. Answer:" "what is the standing man doing with his arms?" "what is the man in red shirt doing? Options: (a) laughing (b) crying (c) singing (d) yelling" "Question: what is the man doing with the pole?" "Question: why is the man kneeling on the ground?"	Men
4	"how many more animals need to be added to all of these to get the number ten?" "Question: how many big elephants are inside of this zoo enclosure together? Options: (a) one (b) four (c) two (d) three, Short answer:" "Q: how many people are seated on the staircase made of wood? A:" "Question: how many donuts are there?" "What is the answer to the following question? "how many engines are visible?""	Counting

Table 16: **Sampled instructions from different clusters.**