

# EvoMoE: Expert Evolution in Mixture of Experts for Multimodal Large Language Models

Linglin Jing<sup>1</sup>\*, Yuting Gao<sup>2</sup>, Zhigang Wang<sup>1</sup>, Wang Lan<sup>2</sup>,  
Yiwen Tang<sup>1</sup>, Wenhai Wang<sup>1</sup>, Kaipeng Zhang<sup>1</sup>, Qingpei Guo<sup>2†</sup>

<sup>1</sup>Shanghai AI Laboratory,

<sup>2</sup>Ant Group

{l.jing@laboro.ac.uk}

## Abstract

Recent advancements have shown that the Mixture of Experts (MoE) approach significantly enhances the capacity of large language models (LLMs) and improves performance on downstream tasks. Building on these promising results, multi-modal large language models (MLLMs) have increasingly adopted MoE techniques. However, existing multi-modal MoE tuning methods typically face two key challenges: *expert uniformity* and *router rigidity*. Expert uniformity occurs because MoE experts are often initialized by simply replicating the FFN parameters from LLMs, leading to homogenized experts and weakening the intended diversification of the MoE architecture. Meanwhile, router rigidity stems from the prevalent use of static linear routers for expert selection, which fail to distinguish between visual and textual tokens, resulting in similar expert distributions for image and text. To address these limitations, we propose EvoMoE, an innovative MoE tuning framework. EvoMoE introduces a meticulously designed expert initialization strategy that progressively evolves multiple robust experts from a single trainable expert, a process termed expert evolution that specifically targets severe expert homogenization. Furthermore, we introduce the Dynamic Token-aware Router (DTR), a novel routing mechanism that allocates input tokens to appropriate experts based on their modality and intrinsic token values. This dynamic routing is facilitated by hypernetworks, which dynamically generate routing weights tailored for each individual token. Extensive experiments demonstrate that EvoMoE significantly outperforms other sparse MLLMs across a variety of multi-modal benchmarks, including MME, MMBench, TextVQA, and POPE. Our results highlight the effectiveness of EvoMoE in enhancing the performance of MLLMs by addressing the critical issues of expert uniformity and router rigidity.

## 1 Introduction

Multi-modal Large Language Models (MLLMs), such as GPT-4 [1] and Llama 3 [14], have achieved significant success in addressing open-world tasks, thanks to their scaled-up architectures. However, scaling up models often increases computational demands and is limited by device capacity. To address these challenges, sparsely activated mixture-of-expert (MoE) models have gained popularity in large language models (LLMs) [51, 29, 47, 36, 53, 52], reducing computational costs and enhancing efficiency. MoE models typically include multiple experts and a routing network that selects the optimal expert for each input token. This design minimizes interference among diverse input tokens, enabling each expert to specialize more effectively in specific tasks. For example, DeepSeek V3 [26]

\*This work was completed during his internship at Ant Group.

†Corresponding author

employ MoE language models with 671B parameters, activating 37B parameters, and have achieved notable results.

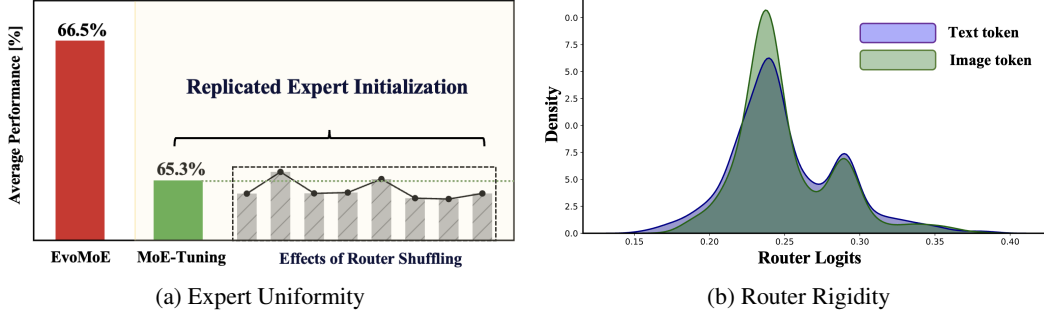


Figure 1: Two key challenges in MoE-tuning. (a) **Expert Uniformity**: Randomly shuffling the router during inference results in negligible performance degradation, suggesting uniformity among experts derived from replicated initialization. (b) **Router Rigidity**: Kernel density estimation (KDE) of the logits for image and text tokens reveals that the linear router generates input-insensitive selections, leading to static distributions with significant overlap in the density of logits for image and text token.

The success of MoE in LLMs has spurred significant interest in its application to MLLMs [20, 25, 26, 37, 44, 49, 54]. A recent prominent approach is MoE-LLaVA [25], which introduces MoE-tuning, a multi-stage process that converts dense MLLM models into sparse MoE structures during instruction tuning, specifically tailored for multi-modal tasks. However, in MoE-tuning, experts are typically initialized by replication, leading to the first critical challenge: *expert uniformity*. This results in expert homogenization during multi-stage tuning, thereby impeding specialization and undermining the efficacy of the MoE framework. Figure 1a illustrates an experiment on expert uniformity, in which we repeatedly shuffled the logits across router layers during evaluation and found no significant drop in average performance. This indicates that experts, which were initially replicated from a common source, tend to become homogeneous after training rather than developing specialized functions. This finding contradicts the fundamental principle of the Mixture-of-Experts (MoE) approach, which aims to enhance task-specific performance through the use of diverse experts. Additionally, MoE-tuning commonly employs a simple linear layer for expert assignment, mimicking the approach used in LLMs. This leads to the second challenge: *router rigidity*. The shared linear router struggles to differentiate between visual and text tokens in MLLMs, resulting in uniform predictions. This limits the model’s adaptability and effectiveness in multi-modal tasks. Figure 1b illustrates this rigidity using Kernel Density Estimation (KDE) plot, revealing significant overlap in the logit distributions of image and text tokens. This overlap indicates that the router becomes inflexible during training, producing uniform output distributions regardless of the input type, thus restricting the model’s adaptability in multi-modal tasks.

To tackle the aforementioned challenges, we introduce **EvoMoE**, a sparse MoE framework tailored for MLLM. Our method builds on MoE-tuning framework, gradually transforming dense models into MoE structures. To address the challenge of *expert uniformity* and enhance the diversity of expert initialization, we introduce a novel approach called Expert Evolution, which generates diverse MoE experts by iteratively adapting expert parameters through a dynamic evolution value. This evolution value integrates prior expertise with gradient-based updates, enabling continuous refinement and evolution. Consequently, the technique evolves multiple robust MoE experts from a single trainable expert. To address *router rigidity* and enhance the connection between the router and input modalities, we propose the Dynamic Token-aware Router (DTR). This router dynamically allocates input tokens to specific experts. Specifically, we employ a hypernetwork to generate unique parameters for each router, tailored to the unique value of each token.

Our core contributions are summarized as follows:

- We introduce EvoMoE, an innovative MoE-tuning framework specifically designed for MLLMs, which effectively addresses two critical challenges: expert uniformity and router rigidity.
- To tackle expert uniformity, we propose a novel method termed expert evolution, which flexibly generates a diverse set of MoE experts. Furthermore, to mitigate router rigidity, we introduce DTR, which assigns input tokens to specific experts based on their modality and intrinsic value.

- Extensive experiments on language models of various sizes demonstrate that EvoMoE achieves better performance with fewer activated parameters.

## 2 Related Works

### 2.1 multi-modal Large Language Model.

LLMs [26, 44, 41, 34, 5] have demonstrated outstanding capabilities in reasoning, comprehension, and question answering. Building on this, recent studies have extended LLMs into the visual domain, leading to the creation of MLLMs. LLaVA 1.5 [28] marks a notable advancement in MLLMs by integrating visual and textual modalities using a simple yet effective architecture, employing a pre-trained vision encoder and language model linked by a lightweight projection layer, achieving strong performance across multi-modal tasks. Recent MLLMs continue to push the boundaries of vision-language integration through novel architectures. For instance, InternVL [9] enhances fine-grained visual-semantic alignment by decomposing high-resolution images into regional patches with a dynamic multi-scale module and fusing features through pixel-shuffle-based method. Qwen2.5-VL [3] combines the Qwen language model [2] with refined vision-language alignment techniques, using fine-grained multi-modal attention and dynamic adaptation to excel in various multi-modal benchmarks. Meanwhile, scaling efforts in systems like GPT-4o [16], Gemini [39], and DeepSeek-VL [31] highlights the importance of model and data scaling, with expanded architectures showing emergent capabilities in multi-modal reasoning, code synthesis, and long-context comprehension.

### 2.2 Mixture of Experts in Multi-modal Learning.

Given the substantial computational overhead associated with training and deploying MLLMs, researchers are increasingly turning to the MoE architecture to enhance efficiency. This approach can be broadly categorized into two main strategies: one directly integrates multi-modal capabilities into an LLM with an MoE architecture, as seen in MLLMs like Qwen2.5-VL [3], Kimi-VL [40] and MiniMax-VL [21], leveraging its inherent efficiency and scalability. The other strategy extends a dense LLM to an MoE-based architecture, offering greater flexibility in adapting to diverse tasks and modalities while maintaining computational efficiency. For instance, HyperLLaVA [49] integrates additional visual experts within the vision encoder and the LLM. These hypernet-based experts dynamically capture input characteristics, thereby offering improved feature extraction capabilities within these components. LLaVA-MoLE [7] creates a set of LoRA experts and a linear router for the FFN layer to mitigate data conflicts when combining multiple distinct instruction datasets. Recently, MoE-LLaVA [25] present MoE-tuning, a novel three-stage framework that progressively converts the dense FFN layers in LLMs into a MoE structure. This method incorporates a linear router, significantly lowering the required activation parameters while maintaining or even surpassing the performance of dense models. Building on MoE-tuning, Eve [6] incorporates an extra visual expert in stage III of the LLM to distinguish image tokens from text tokens. However, these MoE-tuning approaches encounter challenges concerning Expert Uniformity and Router Rigidity.

## 3 Methods

In this paper, we present EvoMoE, a novel MoE-tuning approach addressing expert uniformity and router rigidity. Section 3.1 overviews the three-stage tuning framework of EvoMoE. In Section 3.2, we introduce the expert evolution strategy for generating diverse MoE experts during Stage II. Finally, Section 3.3 details the Stage III routing mechanism, which dynamically assigns input tokens to suitable experts based on modality and inherent token values.

### 3.1 Framework Overview

In the MoE-tuning approach, the initial pre-training phase establishes cross-modal alignment by utilizing an MLP projector to map image tokens into the LLM’s latent space, equipping the model with foundational visual-language understanding capabilities. Leveraging this groundwork, we introduce EvoMoE, building upon the pre-training phase of MoE-LLaVA [25] as our initialization base. Our methodology further advances this foundation through a three-stage framework designed to systematically evolve the dense pre-trained backbone into a sparse MoE architecture. This framework

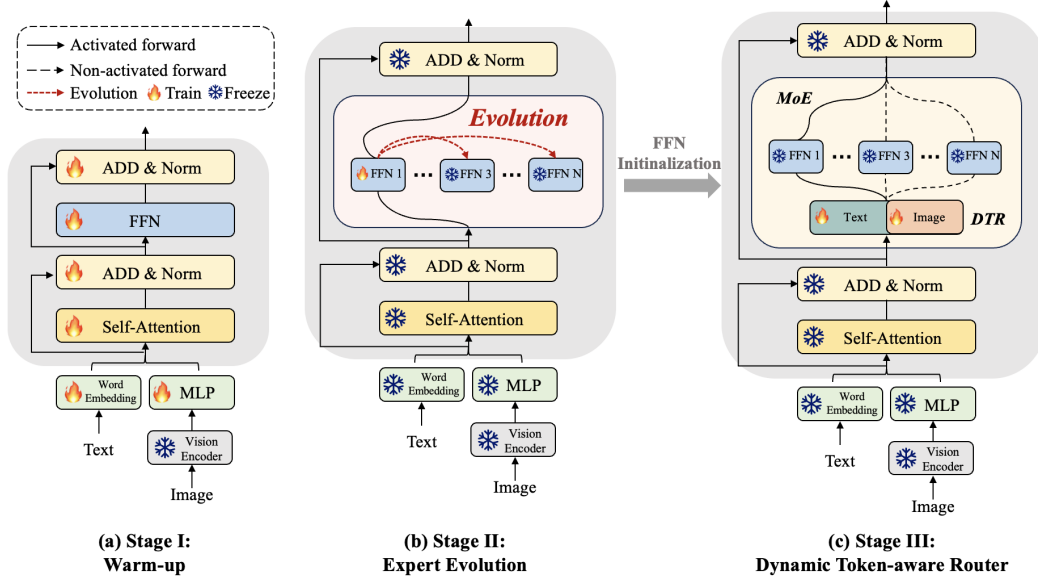


Figure 2: **The framework of EvoMoE.** EvoMoE comprises three stages of instruction-tuning: (a) Warm-up: Begin training with multi-modal instruction data to familiarize the model with understanding capabilities, utilizing parameters initialized during the MoE-LLaVA [25] pretraining stage. (b) Expert Evolution: Train only FFN1, while evolving other experts from FFN1, and (c) Dynamic Token-aware Router: Use FFNs evolved in Stage II for expert initialization and incorporate the DTR for MoE, with only the DTR trainable during this stage, while all other parameters remain frozen.

incorporates two key components: (1) Expert Evolution: Gradually generate new experts using an evolution strategy. (2) Dynamic Token-aware Routing: Implement a dynamic routing mechanism that prioritizes input-relevant experts. Figure 2, coupled with the subsequent description, clarifies the details of the three-stage framework:

**Stage I: Understanding Warm-up.** To equip the model with basic instruction-following capabilities, we utilize a collection of instruction datasets to train all parameters of the dense LLM and the corresponding Multi-Layer Perceptron (MLP) layer.

**Stage II: Expert Evolution.** In this stage, we introduce a novel methodology for constructing Mixture-of-Experts (MoE) experts, where each expert is instantiated as a unique Feed-Forward Network (FFN) layer within the LLM. By employing expert evolution, a process that dynamically balances prior expertise with gradient-based updates, we progressively derive multiple robust FFN experts from a single trainable FFN during training. This iterative approach enables continuous refinement and adaptation, enhancing the diversity, specialization, and robustness of the experts. As a result, the model’s adaptability and overall performance are significantly improved.

**Stage III: Dynamic Token-aware Router.** In this stage, we introduce the Dynamic Token-aware Router (DTR), which dynamically allocates input tokens to appropriate experts based on their modality. The DTR parameters are generated by a hypernetwork, creating routing decisions specifically tailored for each input token. The experts are initialized through the evolutionary process described in Stage II.

### 3.2 MoE Expert Evolution

The existing MoE-tuning approach typically replicates the FFN parameters to initialize multiple MoE experts, resulting in expert uniformity issues during training. To address this challenge, as illustrated in Figure 2 (b), we propose a novel initialization strategy, expert evolution, which gradually evolves new experts by dynamically balancing prior expertise with gradient-based updates:

$$\theta_n \leftarrow \beta \cdot \theta_1 + (1 - \beta) \cdot \nabla \theta_1, \quad (1)$$

where  $\theta_1$  represents the network parameters of the original trainable FFN, initialized using the output of stage I and designated as expert 1.  $\theta_n$  denotes the newly generated FFN experts, where the index  $n = [2, 3, \dots, N]$  represents each of the  $N$  experts. Meanwhile,  $\nabla\theta_1$  represents the gradient update for the trainable expert.  $\beta$  denotes the evolution value within the range  $[0,1]$  which controls the evolution rate. A larger  $\beta$  emphasizes historical data, producing a smoother average by diminishing the impact of recent changes. In our experiments,  $\beta$  is randomly assigned a value within a specified range at each training step for improved generalization, as detailed in Section 4.1.

We exclusively train  $\theta_1$  of expert 1, allowing it to evolve into different experts through varying evolution rates  $\beta$ . Importantly, these evolved experts and all other LLM and MLP parameters remain frozen during training.

### 3.3 Dynamic Token-aware Router (DTR)

In the MoE-tuning approach, a shared linear router selects the top  $K$  experts for both visual tokens  $V = [v_1, v_2, \dots, v_P] \in \mathbb{R}^{P \times C}$  and text tokens  $T = [t_1, t_2, \dots, t_M] \in \mathbb{R}^{M \times C}$ , here,  $P$  is the sequence length of visual tokens,  $M$  is the sequence length of text tokens, and  $C$  denotes the hidden size of the LLM. This setup might not differentiate between visual and text tokens, limiting effectiveness in multi-modal tasks and resulting in uniform predictions, known as router rigidity.

To address this challenge, we introduce the Dynamic Token-aware Router (DTR) as a key component of the EvoMoE framework. EvoMoE consists of  $L$  blocks, each integrating multi-head self-attention (MSA), feed-forward neural networks (FFN), layer normalization (LN) and residual connections. The framework is structured as follows:

$$z_0^V = [v_1, v_2, \dots, v_P], \quad z_0^T = [t_1, t_2, \dots, t_M], \quad l = 1 \dots L. \quad (2)$$

$$z_l^{V'} = \text{MSA}(\text{LN}(z_{l-1}^V)) + z_{l-1}^V, \quad z_l^{T'} = \text{MSA}(\text{LN}(z_{l-1}^T)) + z_{l-1}^T. \quad (3)$$

$$z_l^V = \text{FFN}(\text{DTR}(\text{LN}(z_l^{V'}))) + z_l^{V'}, \quad z_l^T = \text{FFN}(\text{DTR}(\text{LN}(z_l^{T'}))) + z_l^{T'}. \quad (4)$$

The architecture of DTR, as depicted in Figure 3, innovatively manages input visual and text tokens through two hypernetworks, denoted as  $\mathcal{H}^V$  and  $\mathcal{H}^T$ . Each hypernetwork consists of two MLP layers, allowing it to generate adaptive parameters customized to each token. For instance, when processing visual tokens  $V$ , the hypernetwork  $\mathcal{H}^V$  predicts modality-specific parameters  $\Theta^V$ , optimized for visual inputs, as detailed below:

$$\Theta^V = (w_1 z'^{(V)} + b_1) w_2 + b_2, \quad (5)$$

where  $w_1$  and  $w_2$  denote the weights for two MLPs in  $\mathcal{H}^V$ , while  $b_1$  and  $b_2$  represent the corresponding biases. Similarly,  $\Theta^T$  denotes the dynamic weights for the text token input, which are generated in a manner analogous to that of the visual.

DTR consists of a pair of down-sampling and up-sampling layers designed to dynamically extract visual and textual information from various input tokens. Finally, the prediction of expert probabilities  $\rho$  is formulated as follows:

$$\Theta_{\text{up}}^\tau, \Theta_{\text{down}}^\tau = \mathcal{H}^\tau(z'^\tau), \text{ where } \tau \in V, T \quad (6)$$

$$\mathcal{E}^\tau = \Theta_{\text{up}}^\tau(\text{SwiGLU}(\Theta_{\text{down}}^\tau(z'^\tau))), \quad (7)$$

$$\rho^\tau = (\phi(\mathcal{E}^\tau)), \quad (8)$$

where  $\tau$  denotes the visual and text input tokens.  $z'$  is the output of a single MSA block.  $\Theta_{\text{up}}$  and  $\Theta_{\text{down}}$  correspond to the weights of the up-sampling and down-sampling layers, respectively, which are generated by the hypernetwork  $\mathcal{H}$ . In this context, the SwiGLU activation function is utilized. The symbol  $\phi$  denotes an MLP layer that serves as the final router. During training, only  $\mathcal{H}^V$ ,  $\mathcal{H}^T$ , and  $\phi$  are fine-tuned. Each token is processed by the top  $K$  experts with the highest probability.

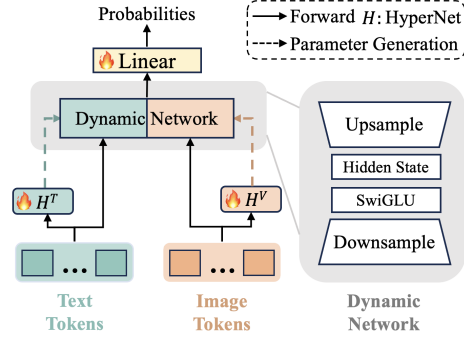


Figure 3: **Dynamic Token-aware Router (DTR)**. Two input-guided hypernetworks dynamically generate network parameters for the up-sampling and down-sampling layers based on visual and text tokens. The final linear layer predicts probabilities and selects the top-k experts. In this module, only the hypernetworks and the linear layer are trainable.

### 3.4 Training Objective

In alignment with [25], the overall loss function of EvoMoE consists of two components: the regression loss:  $\mathcal{L}_{\text{regressive}}$  and the auxiliary loss  $\mathcal{L}_{\text{aux}}$ . Regression loss is designed to optimize model performance, while auxiliary loss aims to promote a balanced load distribution across the router  $\phi$ :

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{regressive}} + \alpha \cdot \mathcal{L}_{\text{aux}}. \quad (9)$$

Here,  $\alpha$  is a hyperparameter that controls the weight of the auxiliary loss and is set to 0.001 during the training process. The detailed formulas are provided in the supplementary materials.

## 4 Experiments

### 4.1 Experiments Setup

**Model Details.** EvoMoE is built on the MoE-tuning and LLaVA 1.5 frameworks, centering on the Evolution Strategy and the Dynamic Token-aware Router (DTR). The training framework suits various sizes, with experiments on LLMs with 0.5B, 1.8B, 2.7B, and 7B parameters showing strong generalization. Importantly, EvoMoE achieves state-of-the-art performance by activating only the top-1 expert, which offers a significant advantage in terms of the number of activated parameters.

**Training Datasets.** In Stage I, following MoE-LLaVA [25], we use a diverse dataset collection, including MIMIC-IT [22], LRV [27], SViT [50], and LVIS [43], to enhance the MLLM’s general multi-modal comprehension skills. Stage II employs the LLaVA-mix-665k [19] dataset to develop evolution experts. In Stage III, the same LLaVA-mix-665k dataset is used to train the DTR.

**Evaluation.** We evaluate the effectiveness and robustness of EvoMoE across diverse scenarios through performance evaluations on an extensive range of multi-modal benchmarks, including VQA-v2 [13], GQA [15], SQA [32], TextVQA [38], POPE [24], MME [12], and MMBench [30].

**Implementation Details.** In our experiments, CLIP-L [35] and SigLIP-L [48] were utilized as the image encoders. Throughout all experiments, the batch size was consistently maintained at 4, with a gradient accumulation of 2. For all the three stages of instruction tuning, the initial learning rate was  $2e-5$ , and we consistently select the top-1 expert across all experiments. In the evolution strategy, the evolution rate is randomly chosen from one of three specified ranges at each training:  $[0.9-0.99]$ ,  $[0.8-0.89]$ , and  $[0.7-0.79]$ . Each range corresponds to one of the three experts generated through the evolution process. Including the original trainable expert, this results in a total of four MoE experts.

### 4.2 Comparison with State-of-the-Art

We evaluated our method against state-of-the-art approaches on four image question-answering benchmarks and three multi-modal understanding toolkits. As illustrated in Table 1, the models were categorized according to the size of LLM into four groups: 0–1B, 1–2B, 2–3B, and 7B.

Compared with the state-of-the-art method MoE-LLaVA, which serves as a baseline for MoE-tuning, EvoMoE demonstrates strong multi-modal understanding capabilities across various LLM sizes and image resolutions. EvoMoE outperforms MoE-LLaVA in the LLMs Qwen2-0.5B, StableLM-1.6B, Qwen-1.8B, Phi-2.7B, and OpenChat-7B. It achieves average performance gains of 1.4% for the 0.5B model, 1.1% for the 1.6B model, 1.2% for the 1.8B model, 1.1% for the 2.7B model and 0.9% for the 7B model, all with fewer activated parameters (activating only top-1 expert). In particular, EvoMoE achieves remarkable improvements in the TextVQA, VQAv2, and GQA benchmarks. For instance, with the Qwen2-0.5B model, it surpasses the baseline by 2.8%, 2.4%, and 1.3%, respectively. In StableLM-1.6B, EvoMoE improves TextVQA performance by 1.4%. Additionally, it outperforms baselines in MMBench evaluations by 2.2%, 1.6%, and 1.0% with Qwen-1.8B, Phi-2.7B, and OpenChat-7B models, respectively. This is particularly noteworthy, given that the baseline approach relies on activating the top-2 experts, which results in a significantly higher number of activated parameters. Ultimately, under the same Phi-2.7 LLM, EvoMoE outperformed the baseline by both 1.6% on input image resolutions of both 336 and 384, demonstrating the flexibility of EvoMoE. Collectively, these results demonstrate that EvoMoE not only outperforms other sparse models but also achieves this with fewer activated parameters.



Table 1: **Comparison of MLLMs on image understanding benchmarks.** ‘LLM’ is the language model component, ‘Act.’ is the number of activated parameters, and ‘Res.’ is the input image resolution. Models ‘Q’, ‘Q’\*, ‘S’, ‘P’, ‘ML’, ‘G’, and ‘O’ refer to Qwen [2], Qwen2 [45], StableLM [4], Phi-2 [17], Mobile LLaMA [18], Gemini [39] and OpenChat [42], respectively. ‘AVG’ is the weighted mean across all benchmarks, with MME values divided by 20 for simplify calculation. \* indicates results re-implemented using MoE-LLaVA [25]. Rows are colored based on the same baseline settings as our method for easier comparison.

Methods	LLM	Act.	Res.	Image Question Answering				Benchmark Toolkit			AVG
				VQA <sup>v2</sup>	GQA	SQA	VQA <sup>t</sup>	POPE	MME	MMB	
<b>0-1B</b>											
<i>Sparse Model</i>											
MoE-LLaVA*	Q'-0.5B	0.6B	336	72.0	56.1	58.0	39.6	84.4	1170.1	57.8	60.9
EvoMoE	Q'-0.5B	0.7B	336	74.4	57.4	59.1	42.4	85.0	1188.6	58.2	62.3
<b>1-2B</b>											
<i>Sparse Model</i>											
MoE-LLaVA [25]	S-1.6B	2.0B	336	76.7	60.3	62.6	50.1	85.7	1318.2	60.2	65.9
EvoMoE	S-1.6B	1.8B	336	76.9	61.2	63.5	51.5	86.4	1359.7	60.9	67.0
MoE-LLaVA [25]	Q-1.8B	2.2B	336	76.2	61.5	63.1	48.0	87.0	1281.6	59.7	65.7
MoE-LLaVA*	Q-1.8B	2.2B	336	76.2	61.0	62.6	48.0	86.5	1288.1	59.4	65.3
EvoMoE	Q-1.8B	2.0B	336	76.9	61.2	63.3	49.3	87.1	1315.6	61.6	66.5
<b>2-3B</b>											
<i>Dense Model</i>											
TinyGPT-V [46]	P-2.7B	2.7B	448	-	33.6	41.2	11.4	50.5	507.8	35.5	-
Mini-Gemini [23]	G-2B	2.0B	336	-	-	-	56.2	-	1341.0	59.8	-
MobileVLM [10]	ML-2.7B	2.7B	336	-	85.4	59.0	46.7	84.6	1296.4	57.0	-
MobileVLM v2 [11]	ML-2.7B	2.7B	336	-	61.1	70.0	57.5	84.7	1440.5	63.2	-
LLaVA-Phi [11]	P-2.7B	2.7B	336	71.4	68.4	66.4	48.6	85.0	1335.1	59.8	66.6
<i>Sparse Model</i>											
Qwen-MoE* [44]	P-2.7B	2.7B	336	77.5	61.1	67.7	52.6	85.9	1434.0	65.4	68.9
MoE-LLaVA [25]	P-2.7B	3.6B	336	77.6	61.4	68.5	51.4	86.3	1423.0	65.2	68.7
EvoMoE	P-2.7B	3.0B	336	77.8	61.6	69.5	52.0	86.6	1450.5	66.8	69.6
MoE-LLaVA [25]	P-2.7B	3.6B	384	79.9	62.6	70.3	57.0	85.7	1431.3	68.0	70.5
EvoMoE	P-2.7B	3.0B	384	80.2	62.8	71.5	57.8	86.5	1450.1	69.6	71.6
<b>7B</b>											
<i>Sparse Model</i>											
MoE-LLaVA*	O-7B	9.6B	336	78.1	61.5	62.8	52.7	86.8	1384.5	64.8	67.9
EvoMoE	O-7B	7.3B	336	78.9	62.6	63.8	53.8	87.3	1391.5	65.8	68.8

### 4.3 Comprehensive Analysis

In this section, we perform an ablation study to explore EvoMoE’s core contributions using the Qwen-1.8B model. We conduct experiments on four image QA benchmarks and three multi-modal understanding benchmarks, using the same training data as [25] for fair comparison.

**Design analysis of our framework.** We conduct several ablation studies to assess the effectiveness of the proposed framework. As depicted in Table 2, (a) represents a dense LLM without any MoE experts. (b) is a standard MoE model proposed by MoE-LLaVA [25] based on MoE-tuning, which includes four experts and a linear router. These results indicate that traditional MoE-tuning method does not provide a significant performance enhancement over dense LLMs on average accuracy. This is due to two challenges faced by MoE-tuning: expert uniformity and router rigidity. In (c), we replaced the linear router with our proposed DTR router based on the MoE-tuning approach. The results indicate that DTR addresses the issue of router rigidity, thereby enhancing performance. In (d), we tested our expert evolution strategy combined with a linear router. All newly created experts originate from the same dense LLM, which can be compared to (a). The results demonstrate that our expert evolution strategy significantly enhances performance while maintaining a model size comparable to dense LLMs, effectively addressing the issue of expert uniformity. Combined with the DTR router, our framework achieves optimal performance.

**The Effectiveness of Evolution Strategy.** Table 3 demonstrates the potential of our proposed expert evolution strategy in MLLMs. In this experiment, we eliminated the router and independently evaluated the multi-modal capabilities of each expert. By fixing  $\beta$  to a constant value, we systematically examined its impact on performance. Expert 1 is a FFN layer with  $\beta = 1.0$ , while Experts 2 to 4 are evolved from Expert 1 by progressively reducing  $\beta$  values. Notably, the evolved experts consistently outperform Expert 1 across the majority of benchmarks. This performance advantage persists even

Table 2: Ablation study on MLLM evaluation benchmarks.

	M-T[25].	Evo.	DTR	Act.	Image Question Answering				Benchmark Toolkit			AVG
					VQA <sup>v2</sup>	GQA	SQA	VQA <sup>T</sup>	POPE	MME	MMB	
(a)				1.8B	76.3	61.0	62.1	48.2	86.4	1286.7	59.7	65.4
(b)	✓			2.2B	76.2	61.0	62.6	48.0	86.5	1288.1	59.4	65.5
(c)	✓		✓	2.4B	76.2	<u>61.1</u>	<u>63.0</u>	48.6	<u>86.8</u>	1310.4	<u>61.4</u>	66.0
(d)		✓		<b>1.8B</b>	<b>77.5</b>	<b>61.2</b>	62.9	<u>48.8</u>	<u>86.8</u>	<u>1311.4</u>	61.3	<u>66.3</u>
<b>EvoMoE</b>		✓	✓	2.0B	<u>76.9</u>	<b>61.2</b>	<b>63.3</b>	<b>49.3</b>	<b>87.1</b>	<b>1315.6</b>	<b>61.6</b>	<b>66.5</b>

Table 3: Ablation study for evolution strategy.

	$\beta$	VQA <sup>v2</sup>	GQA	SQA	VQA <sup>T</sup>	POPE	MME	MMB
Expert 1	1.0	76.3	<u>61.0</u>	62.1	48.2	<u>86.4</u>	1286.7	<u>59.7</u>
Expert 2	0.9	76.4	60.8	<u>62.7</u>	48.6	<b>87.3</b>	<b>1305.7</b>	58.4
Expert 3	0.8	<u>76.7</u>	60.9	<u>62.4</u>	<b>49.0</b>	86.6	<u>1297.3</u>	<b>61.4</b>
Expert 4	0.7	<b>77.1</b>	<b>61.2</b>	<b>62.8</b>	<u>48.7</u>	<u>86.4</u>	1284.5	59.5

under large  $\beta$  decay, where the value is set to 0.9, equivalent to retaining only 10% of updates per step. These systematic improvements empirically validate our core hypothesis: experts generated through expert evolution exhibit significantly greater diversity compared to those generated through straightforward replication. Each evolved expert demonstrates specialized capabilities, excelling in different benchmarks, often outperforming the original expert and effectively addressing the challenge of expert uniformity. In the subsequent stage, we apply the proposed DTR to these evolved experts, enabling better utilization of their specialized capabilities and enhancing overall performance. In our experiments, to enhance generalization, we randomly sample the  $\beta$  value from a predefined range at each training step, rather than using a fixed value.

Table 4: Ablation study for DTR.

	Share	Image	Text	GQA	SQA	VQA <sup>T</sup>	POPE
<b>Linear</b>							
(a)	✓			61.0	62.6	48.0	86.5
(b)		✓	✓	<b>61.2</b>	<u>62.7</u>	48.3	86.6
(c)	✓	✓	✓	<u>61.1</u>	62.2	48.2	86.4
<b>HyperNet</b>							
DTR		✓	✓	<b>61.2</b>	<b>63.3</b>	<b>49.2</b>	<b>87.1</b>
(d)	✓	✓	✓	60.9	<u>62.7</u>	<u>48.4</u>	<u>86.7</u>

Table 5: Ablation study for expert diversity.

	Method	GQA	SQA	VQA <sup>T</sup>	POPE
(a)		61.0	62.6	48.0	86.5
<b>Initialization</b>					
(b)	Noise	60.8	<u>63.1</u>	47.2	86.1
(c)	V-Evo.	<u>61.3</u>	63.0	48.0	<u>86.7</u>
<b>Training</b>					
(d)	Dropout	60.6	62.3	47.4	86.2
(e)	Contrastive	<b>61.5</b>	62.6	47.5	86.3
(f)	Local Loss	60.9	62.2	<u>48.2</u>	85.9
<b>Ours</b>	<b>Evolution</b>	61.2	<b>63.3</b>	<b>49.3</b>	<b>87.1</b>

**Design analysis of DTR.** Figure 4 presents possible architectures for DTR. (a) presents a standard linear router that processes image and text tokens simultaneously. (b) introduces a modality-specific router tailored to differentiate between image and text. (c) incorporates a shared router, which influences the modality-specific router through weighted connections. (d) proposes the hypernetwork as the modality-specific router while also integrating a shared router to enhance flexibility.

Table 4 summarizes the ablation studies. The modality-specific router in (b) outperforms the single router in (a), emphasizing the importance of modality distinction. The HyperNet adaptation improves attention to input token distribution, further improving performance. However, adding a weighted shared router in (c) and (d) results in a decline in overall performance. Ultimately, we adopt the structure of the DTR in Figure 3 in our framework, as it achieves the best performance. Figure 5 visualizes modality preferences of experts on the ScienceQA benchmark, with MoE-tuning results on the left and EvoMoE results on the right. The visualization reveals that traditional MoE-tuning exhibits almost uniform distributions across different inputs,

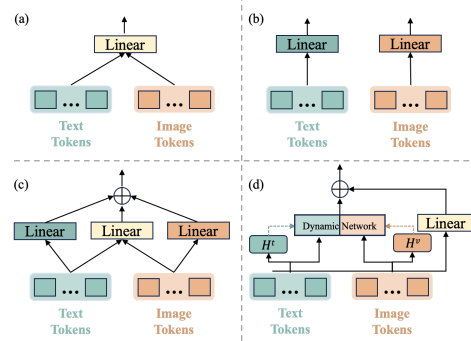


Figure 4: **Design analysis of DTR.** (a) single router; (b) modality-specific router; (c) modality-specific router with shared routing; (d) hyperNet with shared routing.



leading to router rigidity. In contrast, EvoMoE, using DTR, dynamically allocates tokens to suitable experts based on modality, allowing experts to learn specific patterns for efficient, input-guided processing.

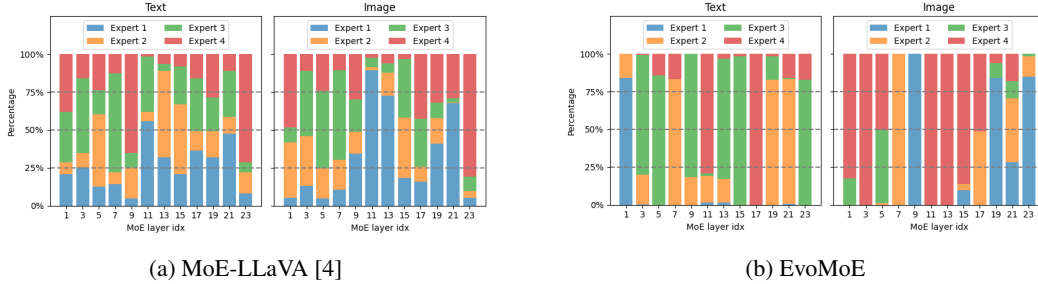


Figure 5: **Distribution of input modalities across different experts** (a) Previous methods exhibited almost uniform distributions across different inputs, leading to router rigidity. (b) EvoMoE dynamically allocates input tokens to the most suitable experts based on their modality.

**Increasing Expert Diversity.** To address homogenization from expert replication, we implemented strategies to improve expert diversity, classified into initialization and training phases. As shown in Table 5, (a) shows the MoE baseline. (b) adds noise during expert initialization, while (c) Vanilla-Evolution shifts expert evolution to Stage I and fine-tunes all experts in Stage II. For training: (d) uses random dropout; (e) incorporates NCE loss [8] among experts; and (f) introduces local loss [33] to increase router entropy for better routing balance. These diversity strategies didn’t significantly boost performance across all metrics, highlighting EvoMoE’s superiority. For detailed comparison, see the Supplementary Material.

**MoE Strategy Exploration.** We further explored additional attempts concerning MoE in Table 6. Incorporating insights from advanced LLMs like DeepSeek-V3 [26], it was found that removing the initial MoE layer, emphasized in DeepSeek-V3, is ineffective in MLLMs. While shared experts are common in LLM MoE implementations, they have not provided significant benefits in MLLMs. Additionally, we explored the introduction of additional trainable parameters at various stages: (1) In stage II, unfreezing all parameters (MSA&FFN) within the LLM led to optimal performance on several benchmarks (GQA, SQA, MMB), though improvements were not consistent across all benchmarks. (2) In stage III, training the entire set of experts alongside the DTR led to a significant performance drop. Lastly, our framework preserved performance using an alternating approach for MoE layers, whereas replacing all dense layers with MoE structures decreased performance.

Table 6: Ablation study on MoE exploration.

Strategy	VQA <sup>v2</sup>	GQA	SQA	VQA <sup>t</sup>	POPE	MME	MMB
<b>LLM MoE Insights</b>							
w/o first layer	76.3	<u>61.1</u>	62.4	48.5	86.4	1285.6	60.6
Share Expert	76.2	61.0	62.6	<u>48.8</u>	<u>86.6</u>	<u>1306.8</u>	60.8
<b>Trainable Parameters</b>							
In Stage II	75.2	<b>61.2</b>	<b>63.8</b>	46.8	86.4	1263.8	<b>61.9</b>
In Stage III	<b>77.0</b>	60.9	62.3	48.4	86.5	1271.2	60.9
<b>MoE Placement</b>							
ALL Layers	74.4	61.0	62.5	47.6	86.3	1280.1	60.4
EvoMoE	<u>76.9</u>	<b>61.2</b>	<u>63.3</u>	<b>49.3</b>	<b>87.1</b>	<b>1315.6</b>	<u>61.6</u>

## 5 Conclusion

In this paper, we introduce EvoMoE, a MoE framework specifically designed for MLLMs. EvoMoE redefines MoE-tuning through two key innovations: expert evolution and the dynamic token-aware router (DTR), effectively addressing two critical challenges in existing MoE-tuning approaches: expert uniformity and router rigidity. The superior performance of EvoMoE, validated through extensive experiments, highlights its potential to unlock new possibilities for the application of MoE in MLLMs.

## References

- [1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
- [2] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)
- [3] Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
- [4] Bellagente, M., Tow, J., Mahan, D., Phung, D., Zhuravinskyi, M., Adithyan, R., Baicoianu, J., Brooks, B., Cooper, N., Datta, A., et al.: Stable lm 2 1.6 b technical report. arXiv preprint arXiv:2402.17834 (2024)
- [5] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al.: A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology* **15**(3), 1–45 (2024)
- [6] Chen, J., Guo, L., Sun, J., Shao, S., Yuan, Z., Lin, L., Zhang, D.: Eve: efficient vision-language pre-training with masked prediction and modality-aware moe. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 1110–1119 (2024)
- [7] Chen, S., Jie, Z., Ma, L.: Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. arXiv preprint arXiv:2401.16160 (2024)
- [8] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: *International conference on machine learning*. pp. 1597–1607. PmlR (2020)
- [9] Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 24185–24198 (2024)
- [10] Chu, X., Qiao, L., Lin, X., Xu, S., Yang, Y., Hu, Y., Wei, F., Zhang, X., Zhang, B., Wei, X., Shen, C.: Mobilevlm : A fast, strong and open vision language assistant for mobile devices (2023), <https://arxiv.org/abs/2312.16886>
- [11] Chu, X., Qiao, L., Zhang, X., Xu, S., Wei, F., Yang, Y., Sun, X., Hu, Y., Lin, X., Zhang, B., Shen, C.: Mobilevlm v2: Faster and stronger baseline for vision language model (2024), <https://arxiv.org/abs/2402.03766>
- [12] Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., Li, K., Sun, X., Ji, R.: Mme: A comprehensive evaluation benchmark for multimodal large language models. *ArXiv abs/2306.13394* (2023), <https://api.semanticscholar.org/CorpusID:259243928>
- [13] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 6904–6913 (2017)
- [14] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
- [15] Hudson, D.A., Manning, C.D.: 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*. pp. 6700–6709 (2019)
- [16] Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)

- [17] Javaheripi, M., Bubeck, S., Abdin, M., Aneja, J., Bubeck, S., Mendes, C.C.T., Chen, W., Del Giorno, A., Eldan, R., Gopi, S., et al.: Phi-2: The surprising power of small language models. *Microsoft Research Blog* **1**(3), 3 (2023)
- [18] Kan, K.B., Mun, H., Cao, G., Lee, Y.: Mobile-llama: Instruction fine-tuning open-source llm for network analysis in 5g networks. *IEEE Network* (2024)
- [19] Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9579–9589 (2024)
- [20] Lee, B.K., Park, B., Won Kim, C., Man Ro, Y.: Moai: Mixture of all intelligence for large language and vision models. In: *European Conference on Computer Vision*. pp. 273–302. Springer (2024)
- [21] Li, A., Gong, B., Yang, B., Shan, B., Liu, C., Zhu, C., Zhang, C., Guo, C., Chen, D., Li, D., et al.: Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313* (2025)
- [22] Li, B., Zhang, Y., Chen, L., Wang, J., Pu, F., Yang, J., Li, C., Liu, Z.: Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425* (2023)
- [23] Li, Y., Zhang, Y., Wang, C., Zhong, Z., Chen, Y., Chu, R., Liu, S., Jia, J.: Mini-gemini: Mining the potential of multi-modality vision language models (2024), <https://arxiv.org/abs/2403.18814>
- [24] Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355* (2023)
- [25] Lin, B., Tang, Z., Ye, Y., Cui, J., Zhu, B., Jin, P., Huang, J., Zhang, J., Pang, Y., Ning, M., et al.: Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947* (2024)
- [26] Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024)
- [27] Liu, F., Lin, K., Li, L., Wang, J., Yacoob, Y., Wang, L.: Aligning large multi-modal model with robust instruction tuning. *CoRR* (2023)
- [28] Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning (2023)
- [29] Liu, J., Tang, P., Wang, W., Ren, Y., Hou, X., Heng, P.A., Guo, M., Li, C.: A survey on inference optimization techniques for mixture of experts models. *arXiv preprint arXiv:2412.14219* (2024)
- [30] Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? In: *European conference on computer vision*. pp. 216–233. Springer (2024)
- [31] Lu, H., Liu, W., Zhang, B., Wang, B., Dong, K., Liu, B., Sun, J., Ren, T., Li, Z., Yang, H., et al.: Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525* (2024)
- [32] Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems* **35**, 2507–2521 (2022)
- [33] Mustafa, B., Riquelme, C., Puigcerver, J., Jenatton, R., Houlsby, N.: Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems* **35**, 9564–9576 (2022)
- [34] Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., Mian, A.: A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435* (2023)

- [35] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
- [36] Rajbhandari, S., Li, C., Yao, Z., Zhang, M., Aminabadi, R.Y., Awan, A.A., Rasley, J., He, Y.: Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. In: International conference on machine learning. pp. 18332–18346. PMLR (2022)
- [37] Rang, M., Bi, Z., Liu, C., Tang, Y., Han, K., Wang, Y.: Eve: Efficient multimodal vision language models with elastic visual experts. arXiv preprint arXiv:2501.04322 (2025)
- [38] Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8317–8326 (2019)
- [39] Team, G., Georgiev, P., Lei, V.I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
- [40] Team, K., Du, A., Yin, B., Xing, B., Qu, B., Wang, B., Chen, C., Zhang, C., Du, C., Wei, C., et al.: Kimi-vl technical report. arXiv preprint arXiv:2504.07491 (2025)
- [41] Thirunavukarasu, A.J., Ting, D.S.J., Elangovan, K., Gutierrez, L., Tan, T.F., Ting, D.S.W.: Large language models in medicine. *Nature medicine* **29**(8), 1930–1940 (2023)
- [42] Wang, G., Cheng, S., Zhan, X., Li, X., Song, S., Liu, Y.: Openchat: Advancing open-source language models with mixed-quality data. arXiv preprint arXiv:2309.11235 (2023)
- [43] Wang, J., Meng, L., Weng, Z., He, B., Wu, Z., Jiang, Y.G.: To see is to believe: Prompting gpt-4v for better visual instruction tuning. arXiv preprint arXiv:2311.07574 (2023)
- [44] Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., Fan, Z.: Qwen2 technical report (2024), <https://arxiv.org/abs/2407.10671>
- [45] Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al.: Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115 (2024)
- [46] Yuan, Z., Li, Z., Huang, W., Ye, Y., Sun, L.: Tinygpt-v: Efficient multimodal large language model via small backbones (2024), <https://arxiv.org/abs/2312.16862>
- [47] Zhai, M., He, J., Ma, Z., Zong, Z., Zhang, R., Zhai, J.: {SmartMoE}: Efficiently training {Sparsely-Activated} models through combining offline and online parallelization. In: 2023 USENIX Annual Technical Conference (USENIX ATC 23). pp. 961–975 (2023)
- [48] Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11975–11986 (2023)
- [49] Zhang, W., Lin, T., Liu, J., Shu, F., Li, H., Zhang, L., Wanggui, H., Zhou, H., Lv, Z., Jiang, H., et al.: Hyperllava: Dynamic visual and language expert tuning for multimodal large language models. arXiv preprint arXiv:2403.13447 (2024)
- [50] Zhao, B., Wu, B., He, M., Huang, T.: Svit: Scaling up visual instruction tuning. arXiv preprint arXiv:2307.04087 (2023)
- [51] Zhao, H., Qiu, Z., Wu, H., Wang, Z., He, Z., Fu, J.: Hypermoe: Towards better mixture of experts via transferring among experts. arXiv preprint arXiv:2402.12656 (2024)

- [52] Zhong, S., Liang, L., Wang, Y., Wang, R., Huang, R., Li, M.: Adapmoe: Adaptive sensitivity-based expert gating and management for efficient moe inference. In: Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design. pp. 1–9 (2024)
- [53] Zhu, T., Qu, X., Dong, D., Ruan, J., Tong, J., He, C., Cheng, Y.: Llama-moe: Building mixture-of-experts from llama with continual pre-training. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 15913–15923 (2024)
- [54] Zong, Z., Ma, B., Shen, D., Song, G., Shao, H., Jiang, D., Li, H., Liu, Y.: Mova: Adapting mixture of vision experts to multimodal context. arXiv preprint arXiv:2404.13046 (2024)



# Supplementary Materials

## A Overview

This document provides a list of supplemental materials to support the main paper. It includes more ablation studies and visualization cases specifically based on the Qwen-1.8B model.

## B Ablation Studies

### B.1 Training Details.

Table 7 presents the training hyperparameters used across three stages for all models evaluated within our framework, including Qwen2-0.5B, StableLM-1.6B, Qwen-1.8B, Phi2-2.7B, and OpenChat-7B.

Table 7: Training hyperparameters.

Configuration	Stage I	Stage II	Stage III
Experts	-	-	4
Top-k	-	-	1
Data	Hybird-PT	LLaVA-FT	LLaVA-FT
Deepspeed	Zero2	Zero2	Zero2_offload
Image Resolution		336*336	
Image encoder		CLIP-Large/336	
Image projector		MLP with GeLU	
Epoch		1	
Learning rate		2e-5	
Learning rate schdule		Cosine decay	
Weight decay		0.0	
Text max length		2048	
Precision		Bf16	
Global batch size	256	64	64
Training steps	5200	10395	10395
Training hours	6.0	12.0	7.0
Epoch		1	
GPU		8xA100-80G	

### B.2 Training Objective.

The overall loss function of EvoMoE consists of two components: the regression loss:  $\mathcal{L}_{\text{regressive}}$  and the auxiliary loss  $\mathcal{L}_{\text{aux}}$ . Regression loss is designed to optimize model performance, while auxiliary loss aims to promote a balanced load distribution across the router:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{regressive}} + \alpha \cdot \mathcal{L}_{\text{aux}}. \quad (10)$$

Here,  $\alpha$  is a hyperparameter that controls the weight of the auxiliary loss and is set to 0.001 during the training process.

**Auto-Regressive Loss.** The output of EvoMoE is denoted by  $\Upsilon$ , which represents a sequence generated progressively, with each text element produced step-by-step:

$$\mathcal{L}_{\text{regressive}} = - \sum_{i=1}^N \log p \left( \Upsilon^{[i]} \mid \vartheta, \Gamma^{[i-1]} \right) \quad (11)$$

where  $\vartheta$  denote the output of the vision embedding from the projection layer,  $\Gamma$  represent the output of the text embedding from the word embedding layer.  $N$  is the length of the output sequence.

**Balance Loss.** A differentiable load balance loss is employed in each router layer to encourage experts to process tokens in a balanced manner, as defined below:

$$\mathcal{L}_{\text{aux}} = E \cdot \sum_{i=1}^E \mathcal{F}_i \cdot \mathcal{G}_i \quad (12)$$

where  $\mathcal{F}$  denotes the fraction of tokens processed by each expert.  $\mathcal{G}$  represents the average routing probability for each expert, and  $E = 4$  is the total number of experts in our paper.

### B.3 More experiments.

**Architecture details of EvoMoE.** Table 8 details the activated and training parameters for the dense model, EvoMoE, and our baseline, MoE-LLaVA, across multiple LLMs. The results show that EvoMoE activates only the top-1 expert while adding a minimal number of parameters through Dynamic Token-aware Router (DTR). Consequently, the number of activated parameters in EvoMoE is lower than that of MoE-LLaVA. Notably, during training, EvoMoE updates only a single FFN, with additional experts generated through the evolution of this primary FFN. This approach leads to a significant reduction in the total parameter count compared to MoE-LLaVA, thereby enhancing overall efficiency.

Table 8: Architecture details of EvoMoE.

Model	Experts	Top-k	MoE Layers	Embedding	Width	Layers	FFN	FFN Factor	Heads	Router	Activated Param.	Training Param.
Qwen2-0.5B	-	-	-							-	0.5B	0.5B
MoE-LLaVA	4	2	12	151936	1024	24	2816	3	24	2048	0.6B	0.8B
EvoMoE	4	1	12							34760	0.7B	0.7B
StableLM-1.6B	-	-	-							-	1.6B	1.6B
MoE-LLaVA	4	2	16	100352	2560	32	10240	2	32	2048	2.0B	2.9B
EvoMoE	4	1	16							34760	1.8B	1.8B
Qwen-1.8B	-	-	-							-	1.8B	1.8B
MoE-LLaVA	4	2	12	151936	2048	24	5504	3	16	2048	2.2B	3.1B
EvoMoE	4	1	12							34760	2.0B	2.0B
Phi2-2.7B	-	-	-							-	2.7B	2.7B
MoE-LLaVA	4	2	16	51200	2560	32	10240	2	32	2048	3.6B	5.3B
EvoMoE	4	1	16							34760	4.5B	7.8B
OpenChat-7B	-	-	-							-	6.7B	6.7B
MoE-LLaVA	4	2	16	32000	4096	32	14366	3	32	2048	9.6B	15.2B
EvoMoE	4	1	16							34760	7.3B	7.3B

**Performance Comparison of Different Model Sizes.** Table 9 presents a comprehensive performance comparison of dense, MoE, and EvoMoE models across various LLM sizes. Utilizing EvoMoE significantly enhances the final performance of the LLMs.

Table 9: Ablation study about the model size of EvoMoE.

Model	Size	MoE	Evo.	VQA <sup>v2</sup>	GQA	SQA	VQA <sup>t</sup>	POPE	MME	MMB	AVG
Qwen2	0.5B	×	×	71.8	56.1	57.7	39.7	84.3	1168.8	57.9	60.7
		✓	×	72.0	56.1	58.0	39.6	84.4	1170.1	57.8	60.9
		✓	✓	<b>74.4</b>	<b>57.4</b>	<b>59.1</b>	<b>42.4</b>	<b>85.0</b>	<b>1188.6</b>	<b>58.2</b>	<b>62.3</b>
StableLM	1.6B	×	×	76.6	60.1	62.5	50.1	85.2	1315.1	60.1	65.7
		✓	×	76.7	60.3	62.6	50.1	85.7	1318.2	60.2	65.9
		✓	✓	<b>76.9</b>	<b>61.2</b>	<b>63.5</b>	<b>51.5</b>	<b>86.4</b>	<b>1359.7</b>	<b>60.9</b>	<b>67.0</b>
Qwen	1.8B	×	×	76.3	61.0	62.1	48.2	86.4	1286.7	59.7	65.4
		✓	×	76.2	61.0	62.6	48.0	86.5	1288.1	59.4	65.5
		✓	✓	<b>76.9</b>	<b>61.2</b>	<b>63.3</b>	<b>49.3</b>	<b>87.1</b>	<b>1315.6</b>	<b>61.6</b>	<b>66.5</b>
Phi-2	2.7B	×	×	77.4	61.1	68.5	51.5	86.0	1418.1	65.1	68.5
		✓	×	77.6	61.4	68.5	51.4	86.3	1423.0	65.2	68.7
		✓	✓	<b>77.8</b>	<b>61.6</b>	<b>69.5</b>	<b>52.0</b>	<b>86.6</b>	<b>1450.5</b>	<b>66.8</b>	<b>69.6</b>
OpenChat	7B	×	×	78.2	61.5	62.9	52.6	86.8	1355.5	65.1	67.8
		✓	×	78.1	61.5	62.8	52.7	86.8	1384.5	64.8	67.9
		✓	✓	<b>78.9</b>	<b>62.6</b>	<b>63.8</b>	<b>53.8</b>	<b>87.3</b>	<b>1391.5</b>	<b>65.8</b>	<b>68.8</b>

**Shuffle Router in MoE-tuning.** In Table 10, we conducted multiple tests to evaluate the impact of the shuffle router on MoE-tuning. MoE tuning was performed using replicate initialization, serving

as a supplement to Figure 1(a) in the paper. The results demonstrate that the shuffle router does not affect the overall average performance. This finding confirms our hypothesis that traditional MoE-tuning methods suffer from significant expert homogeneity issues. Consequently, randomly selecting different routers makes no substantial difference, a phenomenon we refer to as **expert uniformity**.

Table 10: Shuffle Router in MoE-tuning.

Methods	Image Question Answering				Benchmark Toolkit			AVG
	VQA <sup>v2</sup>	GQA	SQA	VQA <sup>t</sup>	POPE	MME	MMB	
MoE-tuning	76.2	61.0	62.6	48.0	86.5	1288.1	59.4	65.5
<i>Shuffle Router</i>								
1	76.2	60.0	62.3	48.2	86.4	1288.5	59.6	65.3
2	76.2	60.1	62.1	48.2	86.3	1288.8	59.4	65.2
3	76.2	60.4	62.6	47.7	86.5	1290.2	59.8	65.4
4	76.3	60.6	62.5	48.3	86.4	1293.2	59.5	65.6
5	76.1	60.9	62.6	47.9	86.5	1289.9	59.3	65.5
6	76.2	60.6	62.5	47.8	86.4	1287.8	59.6	65.5
7	76.1	60.7	62.6	47.9	86.3	1286.9	59.3	65.4
8	76.0	60.1	62.2	47.9	86.4	12887.1	59.2	65.2

**Training setting.** Tables 11, 12, and 13 outline the training settings for EvoMoE. To evaluate the effect of the number of activated experts, we compare the performance using different top-k strategies. As shown in Table 11, our method demonstrates that top-1 experts performs significantly better than top-2, which is contrary to previous MoE experimental results, thereby confirming the greater efficiency of our approach. To verify the impact of the number of experts on the results, we conducted experiments presented in Table 12. Increasing the number of experts slightly enhances performance, validating previous findings that more sparse experts can achieve better results. Finally, we examine the influence of training epochs. Table 13 shows that when training for 2 epochs, performance on GQA increases significantly, while other metrics experience varying degrees of decline. This indicates that the network tends to overfit on large-scale datasets.

Table 11: The value of top-k.

Top-k	VQA <sup>v2</sup>	GQA	SQA	VQA <sup>t</sup>	POPE	MME	MMB
1	<b>76.9</b>	<b>61.2</b>	<b>63.3</b>	<b>49.3</b>	<b>87.1</b>	<b>1315.6</b>	<b>61.6</b>
2	76.6	61.0	62.4	48.8	86.8	1304.7	60.9

Table 12: The number of experts.

Experts	VQA <sup>v2</sup>	GQA	SQA	VQA <sup>t</sup>	POPE	MME	MMB
2	76.5	61.1	62.5	48.5	86.6	1302.6	61.3
<b>4</b>	<b>76.9</b>	<b>61.2</b>	<b>63.3</b>	<b>49.3</b>	<b>87.1</b>	<b>1315.6</b>	<b>61.6</b>

Table 13: Training Epochs.

Epoch	VQA <sup>v2</sup>	GQA	SQA	VQA <sup>t</sup>	POPE	MME	MMB
<b>1</b>	<b>76.9</b>	61.2	<b>63.3</b>	<b>49.3</b>	<b>87.1</b>	<b>1315.6</b>	<b>61.6</b>
2	76.2	<b>62.4</b>	61.8	48.4	86.5	1262.2	61.0

**Details in Increasing Expert Diversity.** Figure 14 illustrates the experimental details related to noise and dropout as discussed in the "Increasing Expert Diversity" section of the paper. We introduced various types of noise during expert initialization and experimented with different levels of dropout during training. However, the results did not show significant improvements, which led us to develop the expert evolution approach for MoE experts.

**Exploring the Selection of Evolution Value  $\beta$ .** To further explore the selection of the evolution value  $\beta$ , we first fixed  $\beta$ , and individually evaluated its impact on each expert. As shown in Table 1, when  $\beta > 0.5$ , different values perform optimally on their respective benchmarks. In contrast, when  $\beta$

Table 14: Noise and Dropout Parameters Comparison

(a) Noise					(b) Dropout				
Noise	GQA	SQA	VQA <sup>t</sup>	POPE	Dropout	GQA	SQA	VQA <sup>t</sup>	POPE
1e-5	60.8	<b>63.1</b>	<b>48.0</b>	<b>86.5</b>	0.1	<b>60.6</b>	62.1	47.5	86.1
1e-4	<b>61.3</b>	62.2	47.5	86.4	0.2	60.5	<b>62.2</b>	47.4	86.2
1e-3	61.2	<b>63.1</b>	47.4	85.5	0.3	<b>60.6</b>	62.1	<b>47.7</b>	<b>86.3</b>
1e-2	60.4	58.7	47.1	85.1	0.4	60.4	62.0	47.4	86.1
1e-1	58.5	51.1	45.5	84.3	0.5	60.3	62.0	47.2	86.2

$< 0.5$ , most experts exhibit update frequencies similar to the FFN with  $\beta = 0$ , resulting in performance comparable to the  $\beta = 0$  case. To achieve stronger generalization capabilities, we opted not to fix  $\beta$ . Instead, we randomly selected  $\beta$  within a specific range at each training step. For instance, as demonstrated in Table 1, randomly choosing  $\beta$  between 0.9 and 0.99 yielded the best experimental results. Therefore, in our study, we randomly select  $\beta$  from multiple ranges to generate multiple evolved experts.

Table 15: Evolution value  $\beta$ .

$\beta$	VQA <sup>v2</sup>	GQA	SQA	textbfVQA <sup>T</sup>	POPE	MME	MMB
0	76.3	61.0	62.1	48.2	86.4	1286.7	59.7
0.9	76.8	60.8	62.7	48.6	<b>87.3</b>	1290.7	58.4
0.8	76.4	60.9	62.4	<b>49.0</b>	86.6	1277.3	61.4
0.7	<b>76.9</b>	61.0	62.8	48.7	86.4	1284.5	59.5
0.4	76.3	61.0	62.2	48.3	86.5	1285.9	59.7
0.2	76.3	61.0	62.1	48.2	86.4	1286.1	59.6
[0.9 - 0.99]	76.7	<b>61.1</b>	<b>63.0</b>	48.9	87.0	<b>1300.7</b>	<b>61.5</b>

**Visualization.** In Figure 6, we present some VQA examples to demonstrate the capabilities of EvoMoE.

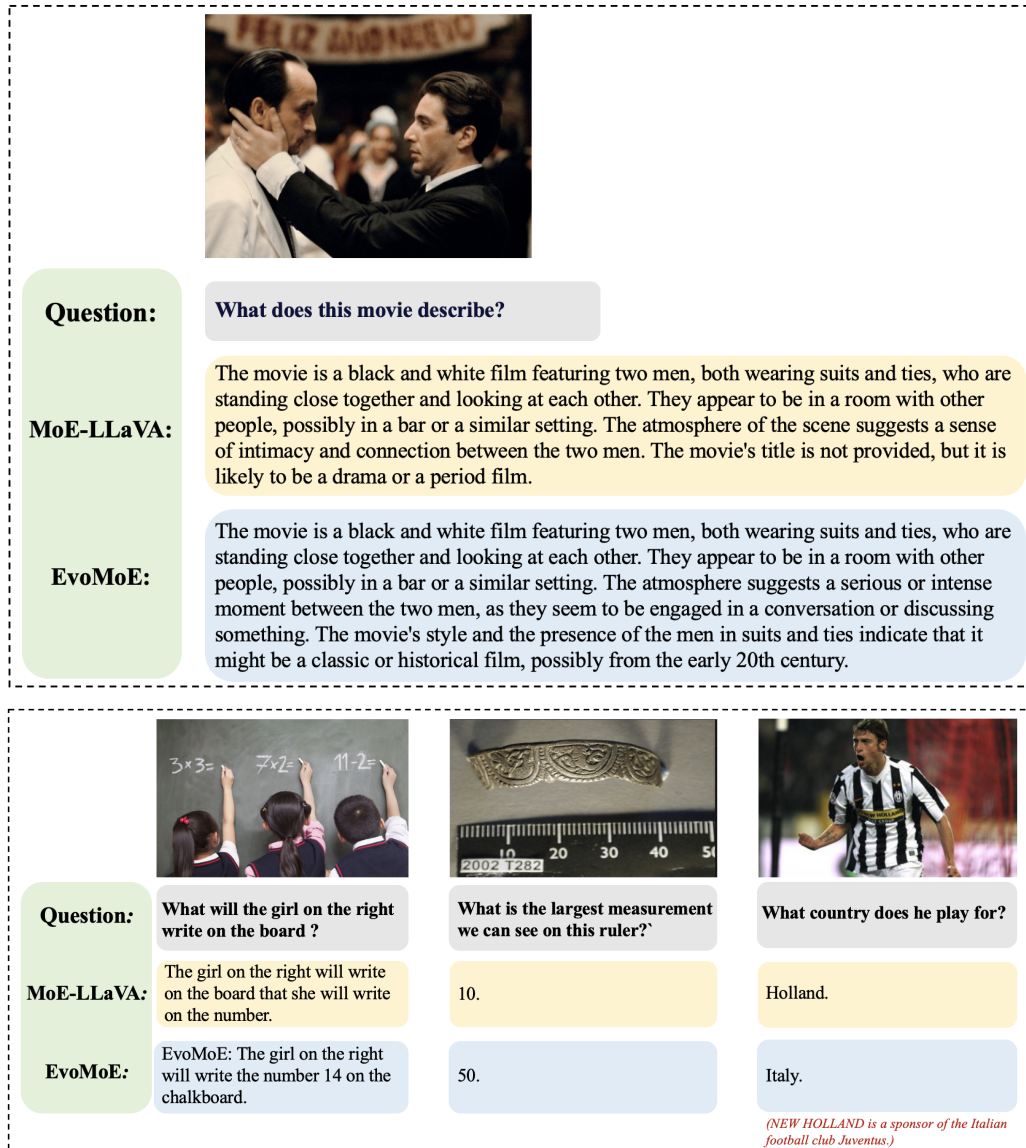


Figure 6: Visual input examples.