GeMoE: Gating Entropy is All You Need for Uncertainty-aware Adaptive Routing in MoE-based Large Vision-Language Models

Chaoxiang Cai[1†], Minghe Weng[2†], Jie Li[2], Yibo Jiang[1], Longrong Yang[2], Zequn Qin[1], and Xi Li[2✉]

[1] School of Software Technology, Zhejiang University, China
[2] College of Computer Science and Technology, Zhejiang University, China
{cxcai,wengminghe,li.jie,jiangyibo,longrongyang,qinzequn,xilizju}@zju.edu.cn

Abstract. With the increase in model parameters and training data, the instruction following and generalization capabilities of Large Vision-Language Models (LVLMs) have been significantly improved. Based on the Mixture-of-Experts (MoE) architecture, LVLMs expand their parameter capacity while maintaining the inference cost. However, traditional MoE methods employ a Top-k static routing strategy, which fails to account for variations in the input and to adaptively select the number of experts, resulting in suboptimal resource utilization. In this paper, we propose viewing token routing as an information encoding task, framing dynamic routing as a Minimum Description Length (MDL) problem in encoding. By validating the connection between MDL and gating entropy in the MoE scenario, we introduce Gating Entropy-based Uncertainty-aware Adaptive Routing (GeMoE) for MoE. Unlike traditional static or heuristic-based dynamic routing methods, GeMoE explicitly models the trade-off between model complexity and performance. By using gating entropy to assess the complexity of tokens, GeMoE adaptively determines the number of experts each token should engage. On a wide range of backbones and benchmarks, our method achieves ***99.5%*** average performance retention compared to the original static routing, while improving average expert activation sparsity by ***36.5%***. The code will be publicly available at https://github.com/caichaoxiang/GeMoE.

Keywords: Gating Entropy · Information Gain · Minimum Description Length · Dynamic Routing · Mixture-of-Experts

1 Introduction

The rapid growth of model parameters and training data has enabled Large Vision–Language Models (LVLMs) to achieve significant improvements across a wide range of tasks [3, 11, 32, 44, 51]. However, as these models continue to scale

† Equal contribution.
✉ Corresponding author.

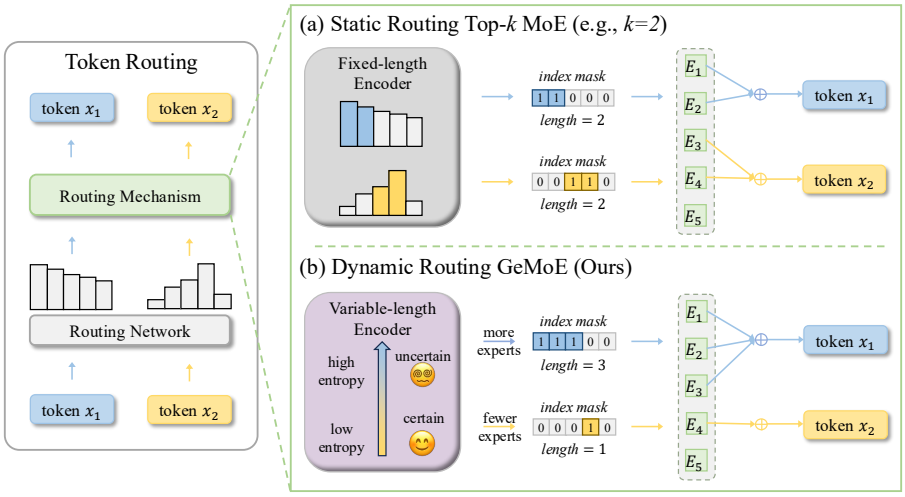


Fig. 1: Comparison of static routing Top-k MoE and our dynamic routing GeMoE: (a) Top-k assigns a fixed number of experts to each token. (b) GeMoE dynamically determines the number of experts to activate based on the token's gating entropy. Gating entropy reflects the token's uncertainty in expert selecting. Higher entropy indicates greater uncertainty, requiring more experts to be selected.

up, their size and resource demands present substantial challenges for deployment and real-world application. In response to these issues, the Mixture-of-Experts (MoE) architecture has emerged as a widely adopted solution [22, 38]. MoE replaces traditional Feed-forward Network (FFN) layers with a set of specialized experts, activating only a subset of them for each input. This approach allows the model to expand its capacity without a proportional increase in inference computation, making it an appealing solution for scaling large models [2,13].

The core of MoE is token routing, which can be viewed as an information encoding task. In this process, the set of experts acts as a codebook, with each expert representing a codeword. The goal of token routing is to select the codewords to encode the tokens. Traditional static Top-k routing as shown in Fig. 1 (a) acts as a fixed-length encoder, assuming uniform information distribution across tokens and allocating equal expert resources to each token. This leads to inefficiencies, as low-information tokens consume computational resources, while high-information tokens are under-resourced and lose semantic content.

Recent dynamic routing methods, such as pseudo-expert construction [24, 50] and expert threshold learning [19, 20, 42, 49], aim to improve both resource efficiency and model performance. As shown in Fig. 1 (b), dynamic routing can be viewed as a variable-length encoding method, optimizing the Minimum Description Length (MDL)—the total length required to encode both the model and its data fit. The goal of optimal dynamic routing is to minimize the MDL. However, current methods are based on heuristics and fail to explicitly balance model complexity with performance, resulting in inefficient expert allocation.

To this end, we propose Gating Entropy-based Uncertainty-aware Adaptive Routing for MoE (GeMoE), which incorporates two key design principles: (*i*) **Looking for a proxy to decrease the MDL**. Since MDL is difficult to compute directly, it is necessary to introduce a proxy. In this work, we use gating entropy as the proxy to reduce the number of experts assigned to tokens with low gating entropy, thus minimizing the MDL. Specifically, when adding more experts improves the model's ability to fit the data (defined as information gain) beyond a certain threshold, the addition of experts helps decrease the MDL. Higher gating entropy indicates a more evenly distributed expert capacity, and in such cases, adding experts provides a higher information gain. Therefore, more experts should be assigned to tokens with high gating entropy, while fewer experts should be allocated to tokens with low gating entropy. This is further validated by the analysis in Section. 3. (*ii*) **Expert assignment prediction based on gating entropy**. We introduce an Expert Assignment Predictor (EAP) that maps gating entropy to the required number of experts. EAP takes token features as input and outputs the corresponding expert count for each token. We design a monotonic loss function to enforce a positive correlation between the number of experts required by a token and its gating entropy, ensuring that tokens with higher entropy are assigned more experts.

In conclusion, our contribution can be summarized as:

- We view token routing from an information encoding perspective, and frame dynamic routing as an MDL problem.
- By linking MDL and token gating entropy through information gain, we demonstrate that when a token has high gating entropy, adding an expert for that token provides significant information gain, thus reducing the MDL.
- We propose a gating entropy-based adaptive routing method, GeMoE, predicting the number of experts. Extensive experiments across backbones and benchmarks compared to the original static routing show that our method achieves a good trade-off between utilization and performance.

2 Related Works

2.1 Large Vision-Language Models

Building on the success of Large Language Models (LLMs) [12, 47], recent research has expanded their capabilities to LVLMs for real-world visual applications. Frameworks like DeepSeek-VL2 [44] and Qwen3-VL [4] typically use frozen visual encoders and trainable projectors to map visual inputs to LLM-compatible representations. Current research focuses on two key areas to improve multi-modal integration: optimizing training strategies [3,5,6,8] using parameter-efficient techniques, and enhancing visual components by scaling instruction-tuning datasets [31, 51]. However, challenges remain in scaling LVLMs, such as limited task generalization and rising inference costs. To address these issues, the MoE architecture offers an effective solution, enabling efficient parameter scaling while preserving computational efficiency.

2.2 Mixture-of-Experts

The MoE architecture [22,25,36,38,39] improves the efficiency of large-scale models by selectively activating a subset of experts for each token [3,9,11,23,28,29,43, 45, 52], effectively decoupling model capacity from computational cost [15, 17]. Current MoE research primarily concentrates on two main areas: (*i*) Efficient utilization of expert resources. Load balancing strategies [1, 7, 16, 26] and expert load prediction methods [17, 46] aim to ensure uniform and effective use of resources. (*ii*) Efficient learning of specific experts. This involves reducing the interference caused by differences in data features and gradients on the internal parameter learning of each expert [10, 18, 34, 53]. To combine the strengths of both areas, research on dynamic routing has gained significant attention.

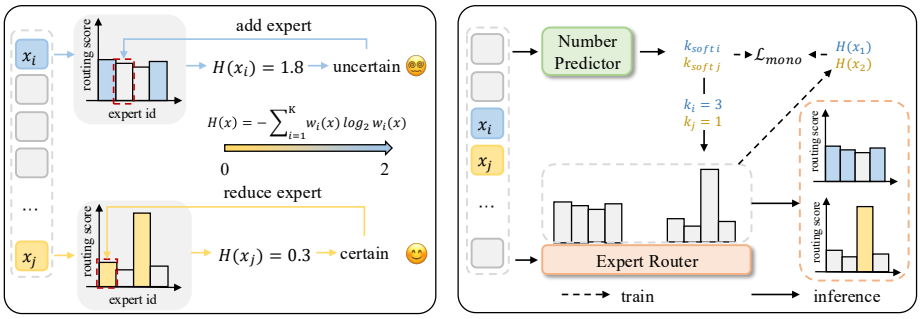
2.3 Dynamic Routing Mixture-of-Experts

Compared to static routing, dynamic routing aims to optimize resource allocation during inference, preserving representation capability while reducing computational overhead [19, 20, 24, 41, 42, 49, 50]. Existing methods mainly fall into threshold-based [19,20,41,42,49] and pseudo-expert [24,50] strategies. Threshold-based strategies dynamically limit expert activation using learnable functions or thresholds [19, 41, 49]. For instance, Top-p routing [20] selects minimum experts required to reach a learnable cumulative threshold, whereas ReMoE [42] dynamically filters experts via a ReLU activation on the router outputs. Pseudo-expert strategies implicitly reduce the computational burden of simpler tokens by introducing null or low-cost virtual experts. Specifically, AdaMoE [50] introduces *zero* experts that output zero, and MoE++ [24] further broadens this approach with *copy* experts for identity mapping and *const* experts for dynamic interpolation. However, they fail to effectively model the balance between model complexity and performance. In this paper, we frame dynamic routing as an MDL problem, linking this objective to gating entropy via information gain. By utilizing token-level gating entropy as a proxy, our mechanism adaptively aligns expert activation with information density, ensuring efficient and robust utilization.

3 Methodology

3.1 Mixture-of-Experts

The MoE layer consists of a K-expert ensemble $\mathcal{E} = [E_1, E_2, \ldots, E_K]$ and a linear-layer router $\mathcal{R}$. As shown in Eq. 1, for an input $x \in \mathbb{R}^D$, the router $\mathcal{R}$ generates weight logits $\mathcal{R}(x) = \mathbf{W} \cdot x$, which are then normalized. The matrix $\mathbf{W} \in \mathbb{R}^{K \times D}$ represents the lightweight, trainable parameters for routing, and $\mathcal{R}_{norm}(x)_i$ is the routing score of the input x for the i-th expert. The output of the MoE layer, as defined in Eq. 2, is computed as a weighted sum of the outputs from the Top-k ($k \leq K$, the number of activated experts, fixed in static routing) experts with the highest probabilities. $E_i(x)$ is the output of the i-th expert, and the weight for each expert is determined by its routing score.



(a) Gating Entropy for decreasing the MDL (b) Expert Allocation based on Gating Entropy

Fig. 2: (a). We frame dynamic routing as an MDL problem and demonstrate that gating entropy can serve as a proxy for decreasing the MDL. (b) Based on gating entropy, we model the number of experts required for a token as positively correlated with its gating entropy, allocating more experts to tokens with higher gating entropy.

$$\mathcal{R}_{norm}(x)_i = \frac{e^{\mathcal{R}(x)_i}}{\sum_{j=1}^{K} e^{\mathcal{R}(x)_j}} \tag{1}$$

$$\mathrm{MoE}(x) = \sum_{i \in \text{Top-}k \text{ ids}} \mathcal{R}_{norm}(x)_i \cdot E_i(x) \tag{2}$$

Due to the presence of multiple experts, it is necessary to impose the load balancing constraint on MoE layers. Traditional methods [17, 30] introduce a differentiable load balancing loss into MoE layers to encourage a more balanced allocation of tokens across experts. As shown in Eq. 3, $\mathcal{F}_i$ is the fraction of tokens processed by expert E_i, while $\mathcal{G}_i$ is the average routing probability of expert E_i.

$$\mathcal{L}_{\text{lb}} = K \cdot \sum_{i=1}^{K} \mathcal{F}_i \cdot \mathcal{G}_i \tag{3}$$

3.2 The Overview of Our Method GeMoE

Token routing is the core mechanism of MoE. Traditional static Top-k routing leads to inefficient expert utilization. Although existing dynamic routing methods have made significant improvements, they still fall short of explicitly modeling the trade-off between model complexity and performance. To address this, we propose a Gating Entropy-based Uncertainty-aware Adaptive Routing for MoE (GeMoE), as illustrated in Fig. 2. Our method consists of two key steps: (i) To explicitly model the balance between model complexity and performance, we frame dynamic routing as an MDL problem and demonstrate that gating entropy can serve as a proxy for decreasing the MDL. (ii) We use gating entropy as prior knowledge to predict the number of expert assignments, thereby establishing a mapping from gating entropy to specific expert allocations.

3.3 Gating Entropy As a Proxy for Decreasing the MDL

Token routing can be fundamentally understood as an information encoding process. Specifically, the token x represents the source information to be encoded. The expert ensemble $\mathcal{E}$ in MoE can be viewed as a codebook for encoding this source information, where each expert E_i corresponds to the $i-th$ codeword in the codebook. Traditional static Top-k routing can be regarded as a fixed-length encoding scheme, implicitly assuming that the information content of all tokens follows a uniform distribution, which results in the same number of experts being allocated to each token. This uniform allocation, however, overlooks variations in token complexity, leading to inefficient utilization of expert capacity.

To address this limitation, dynamic routing methods aim to implement a variable-length encoder that adaptively selects the optimal encoding length for each token by choosing the appropriate codewords from the codebook. This process can be framed as an MDL problem: minimizing the total encoding length required to describe both the model and its fit to the data, thereby balancing model complexity with performance. The formalization is as follows:

$$L(x, \mathcal{E}') = L(\mathcal{E}') + L(x \mid \mathcal{E}'), \mathcal{E}' \subseteq \mathcal{E}, \tag{4}$$

$$MDL \leftarrow \mathcal{E}^* = \arg\min_{\mathcal{E}'}[L(x, \mathcal{E}')], \mathcal{E}' \subseteq \mathcal{E} \tag{5}$$

Given a token x and a model $\mathcal{E}'$, where $\mathcal{E}'$ is a subset of the expert ensemble $\mathcal{E}$, the encoding length of the model $L(\mathcal{E}')$ refers to the number of experts selected. Longer encoding length indicates higher model complexity. While $L(x \mid \mathcal{E}')$ denotes the encoding length of x conditioned on $\mathcal{E}'$, it reflects the model's data-fitting performance based on the selected experts. Longer encoding length suggests a poorer fit of the model to the data. Generally, $L(\mathcal{E}') \uparrow \Rightarrow L(x \mid \mathcal{E}') \downarrow$. Since the encoding lengths of both components are model-dependent, determining the subset $\mathcal{E}^*$ that minimizes the encoding length is essential for effectively applying the MDL principle. However, existing dynamic routing methods rely on heuristic schemes [19,20,24,42,50], which limits the model's ability to explore the trade-off between model complexity and performance, as well as its ability to allocate experts effectively. Fortunately, in MoE-based models, redundancy among experts can occur. As shown in Eq. 6, $\mathcal{E}'_a$ is a proper subset of $\mathcal{E}'_b$, and $\mathcal{E}'_a$ may show performance comparable to that of $\mathcal{E}'_b$. This highlights the potential to reduce model complexity while maintaining performance, specifically by ensuring that $L(\mathcal{E}')$ decreases without causing an increase in $L(x \mid \mathcal{E}')$: $L(\mathcal{E}') \downarrow \not\Rightarrow L(x \mid \mathcal{E}') \uparrow$.

$$L(x \mid \mathcal{E}'_a) - L(x \mid \mathcal{E}'_b) \approx 0, \exists \mathcal{E}'_a \subset \mathcal{E}'_b \subseteq \mathcal{E} \tag{6}$$

To enable dynamic routing through MDL minimization, we establish a connection between MDL and information gain. Suppose the input token to the MoE layer is x and the target output is y. The MDL can be further written as $k \cdot c - \log \sum_{i \in \mathcal{E}'} w_i P_i(y|x)$, where $L(\mathcal{E}') = k \cdot c$, with k denoting the number of experts in $\mathcal{E}'$, and c representing the constant computational penalty for each

expert. $L(x \mid \mathcal{E}') = -\log \sum_{i \in \mathcal{E}'} w_i P_i(y|x)$, where w_i denotes the routing score and $P_i(y|x)$ denotes the predictive probability of x given by expert i. Our goal is to decrease the MDL. After adding an expert E_m to the expert set $\mathcal{E}'$, we have a new expert set $\mathcal{E}'_{new}$. We aim to have $L(x, \mathcal{E}'_{new}) < L(x, \mathcal{E}')$. Substituting this into the above formulation yields:

$$(k+1) \cdot c - \log P(y|x, \mathcal{E}'_{new}) < k \cdot c - \log P(y|x, \mathcal{E}') \tag{7}$$

Simplifying both sides yields the inequality condition: $\log \frac{P(y|x,\mathcal{E}'_{new})}{P(y|x,\mathcal{E}')} > c$, where we define $\log \frac{P(y|x,\mathcal{E}'_{new})}{P(y|x,\mathcal{E}')}$ as the information gain. If the information gain exceeds a constant c, we add the expert; otherwise, no expert is added. This serves as the principled rule for dynamic expert allocation.

While information gain offers a rigorous theoretical criterion, directly calculating it requires executing the candidate expert to obtain $P(y|x, E_m)$, which undermines the computational efficiency of dynamic routing. To address this, we establish a connection between information gain and gating entropy, thereby linking MDL and gating entropy. Based on the above analysis, when the information gain is large, adding an expert decreases the MDL. To predict information gain without requiring expert reasoning, we use gating entropy on the token routing distribution as a natural proxy. Formally, for a token x, we employ its routing distribution $\mathcal{R}_{norm}(x) = \{w_1(x), w_2(x), \ldots, w_K(x)\}$, obtained from the router $\mathcal{R}$, and calculate the gating entropy as follows:

$$H(x \mid \mathcal{R}) = -\sum_{i=1}^{K} w_i(x) \log_2 w_i(x) \tag{8}$$

Higher gating entropy indicates greater routing uncertainty outside the current subset, leading to higher expected gain from extra experts and thus lower MDL. The prediction formula under the MoE architecture can be expressed as follows: $P(y|x) = \sum_i P(z = i|x)P(y|x, z = i)$, where z is the expert variable. For a subset $\mathcal{E}'$, $P(y|x, \mathcal{E}') = \sum_{i \in \mathcal{E}'} \tilde{w}_i P_i(y|x)$, with normalized restricted weights $\tilde{w}_i$. After adding expert E_m, $P(y|x, \mathcal{E}'_{\text{new}}) = \frac{W}{W+w_m} P(y|x, \mathcal{E}') + \frac{w_m}{W+w_m} P_m(y|x)$, $W = \sum_{i \in \mathcal{E}'} w_i$. The formula follows a first-order expansion under $w_m \ll W$, rather than assuming expert independence. For instance, a concentrated distribution (*e.g.*, $w_1 = \frac{1}{2}$, $w_{2\sim n} = \frac{1}{2(n-1)}$) yields limited gain from extra experts ($2 \sim n$), whereas a flatter distribution (*e.g.*, $w_{1\sim n} = \frac{1}{n}$) yields larger gain from extra experts. Thus, gating entropy estimates the gain of changing expert number, balancing complexity and representation capacity. **When a token has high gating entropy, adding an expert for that token brings the large information gain, thereby decreasing the MDL**.

To verify the correlation between gating entropy and information gain, we conduct an analysis on the test set. For each sample, we calculate the model's prediction accuracy at Top-2 and Top-8, along with the average gating entropy of the tokens in the sample. We then sort the samples by their gating entropy and group them into fixed-size intervals. Within each interval, we compute the average accuracy and average gating entropy to analyze potential trends. As shown

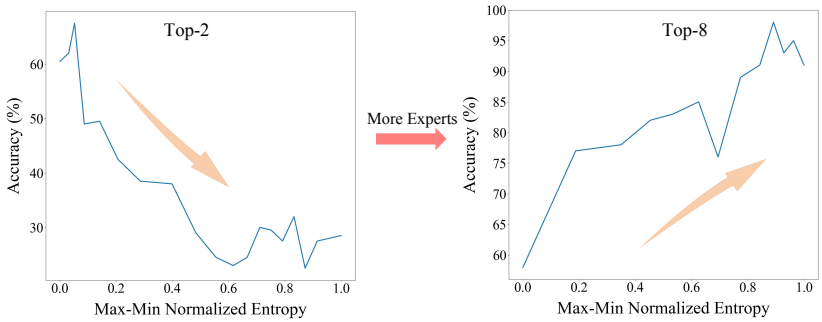


Fig. 3: Average accuracy versus maximum-minimum normalized token routing entropy on ScienceQA [35] (4000 samples), evaluated using MolmoE-1B-7B. Samples are sorted by normalized token routing entropy and grouped into intervals of 200 samples.

in Fig. 3, the results indicate that increasing the number of experts significantly improves performance for data with high average gating entropy. However, for data with low average gating entropy, increasing the number of experts does not significantly improve performance and may even result in a negative gain. These findings clearly demonstrate that adding experts for tokens with high gating entropy brings large information gain, establishing an explicit relationship between the number of experts and gating entropy. This enables an efficient dynamic routing mechanism that balances model complexity and performance.

3.4 Expert Allocation based on Gating Entropy

Having established the positive correlation between gating entropy and information gain when more experts are added, we aim to assign more experts to tokens with higher gating entropy to help achieve good performance on more uncertain tasks, and fewer experts to tokens with lower gating entropy to effectively reduce the overuse of experts for simple tasks. This mapping process requires a monotonically increasing relationship between gating entropy and the required number of experts. To achieve the above goal, we introduce a linear layer expert assignment predictor $\mathcal{P}$ to adaptively learn the mapping process of predicting the number of experts $N = [k_{low}, ..., k_{high}]$ required to be assigned to a token. k_{low} and k_{high} ($k_{low} = 1$, $k_{high} = 8$ in this work.) are the lower and upper limits for expert activation.

$$\mathcal{N}(x)_i = \frac{e^{\mathcal{P}(x)_i}}{\sum_{j=1}^{high-low+1} e^{\mathcal{P}(x)_j}}, \tag{9}$$

$$k_{\text{soft}} = \sum_{i=k_{low}}^{k_{high}} i \cdot \mathcal{N}(x)_{i-k_{low}} \in [k_{low}, k_{high}], \tag{10}$$

$$k = Round(k_{\text{soft}}) \in N, \tag{11}$$

The predictor produces weight logits $\mathcal{P}(x) = \mathbf{W} \cdot x$, where the matrix $\mathbf{W} \in \mathbb{R}^{(k_{high}-k_{low}+1)\times D}$ is the lightweight trainable parameter matrix, and $\mathcal{N}(x)_i$ is the prediction score of the input x to select N_i expert(s). Given a token x, we calculate the probability distribution over the number of experts using $\mathcal{P}$, and then calculate the expected value k_{soft}, which is a soft prediction of the number of experts. The final k is determined by rounding the k_{soft} by using a straight-through estimator. To achieve a monotonically increasing mapping relationship between gating entropy and the number of experts, we employ the monotonic loss. For any token pair (x_i, x_j) within a batch X, if the condition $H(x_i) > H(x_j)$ is satisfied, then $k_{\text{soft}\,i} > k_{\text{soft}\,j}$ is required. The loss is designed as follows:

$$\mathcal{L}_{\text{mono}} = \sum_{(x_i,x_j)\subseteq X} \begin{cases} \max(0, margin - k_{\text{soft}\,i} + k_{\text{soft}\,j}), & \text{if } H(x_i) > H(x_j) \\ \max(0, margin - k_{\text{soft}\,j} + k_{\text{soft}\,i}), & \text{if } H(x_i) < H(x_j) \end{cases} \tag{12}$$

Where margin is a boundary threshold used to define the confidence safety boundary that must be reached for a correct prediction. By combining the cross-entropy loss ($\mathcal{L}_{\text{ce}}$), the proposed monotonic loss ($\mathcal{L}_{\text{mono}}$), and the load balancing loss ($\mathcal{L}_{\text{lb}}$), the overall training objective is formally defined as illustrated in Eq. 13. α and β are weight hyper-parameters. This formula simultaneously optimizes the overall balance between task performance, uncertainty-aware expert routing and expert utilization.

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \alpha\mathcal{L}_{\text{mono}} + \beta\mathcal{L}_{\text{lb}} \tag{13}$$

4 Experiments

4.1 Experimental Setup

Backbones. Native MoE models, rather than dense-copied versions, serve as backbones, promoting greater diversity among experts and resulting in improved model performance [37]. Thus, we focus on fine-tuning the router to optimize its ability to efficiently route data to the experts. We use two backbone models with different sizes to validate the effectiveness and generalization of our method: MolmoE-1B-7B (64Top8) [14] and DeepSeek-VL2-Tiny-1B-3B (64Top6) [44].

Baselines. We evaluate four methods from threshold-based and pseudo-expert dynamic routing strategies. (*i*) DYNMoE [19]. Expert selection occurs only if the similarity exceeds a learned gating threshold. Additionally, an expert difference loss is introduced to encourage diversity among the experts. (*ii*) Top-p [20]. A learnable module is used to dynamically adjust the expert selection threshold, with a dynamic loss that focuses the probability of expert selection. (*iii*) AdaMoE [50]. Four null-experts are added to the model, and the router module is trained to decide when to activate them. (*iv*) MoE++ [24]. Two zero-experts, two copy-experts, and two constant-experts are added, with learnable router parameters. These baselines offer a comprehensive comparison of various dynamic routing strategies, enabling a fair evaluation of our method.

Table 1: Performance comparison of various routing strategies on MolmoE-1B-7B [14]. N_A represents the average number of activated parameters, Avg_k is the average number of activated experts, and Avg_P is the average performance. **Bold** highlights the best-performing results, while underlined denotes the second-best results. ↑ and ↓ indicate that higher or lower values are preferable, respectively.

Method	N_A ↓	Avg_k ↓	MMB	POPE	SQA^I	VQA^T	GQA	MM-Vet	Avg_P ↑
MolmoE-1B-7B	1.58B	8.00	58.24	85.33	87.50	56.37	50.89	45.60	63.99
DYNMoE [19]	1.40B	6.19	51.72	87.84	81.41	**55.16**	49.95	38.90	60.83
Top-p [20]	2.05B	12.65	55.75	87.54	81.66	51.64	47.41	43.10	61.18
AdaMoE [50]	1.39B	6.13	**56.95**	88.27	84.93	52.31	48.02	40.12	61.77
MoE++ [24]	1.48B	7.03	51.20	82.81	81.16	46.04	45.71	38.40	57.55
GeMoE (Ours)	**1.32B**	**5.43**	56.70	**88.38**	**86.98**	54.88	**50.64**	**44.10**	**63.61**

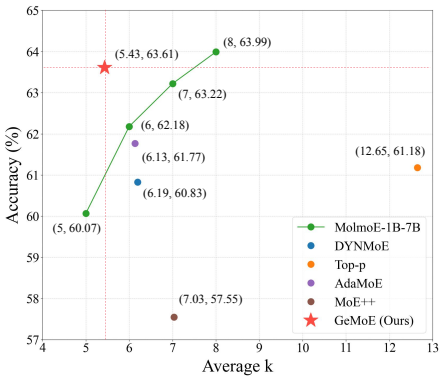


Fig. 4: Average accuracy of various Top-k routing and dynamic routing methods.

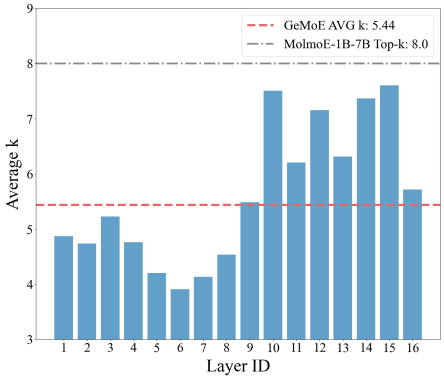


Fig. 5: Average number of experts activated in per MoE layer.

Benchmarks. To thoroughly evaluate the vision-language understanding capabilities and the expert activation rate of each method, we evaluate all methods across a comprehensive set of benchmarks, including MMBench [33], POPE [27], ScienceQA [35], TextVQA [40], GQA [21] and MM-Vet [48].

Configurations. To ensure consistency, only the router, along with method-specific trainable modules, are learnable during MoE training across all methods. All models are trained for one epoch using the LLaVA-1.5-558k [32] dataset.

4.2 Main Results

Comparison with State-of-the-Art Dynamic Routing Strategies. Tab. 1 and Fig. 4 present a comparative analysis of our gating entropy-based uncertainty-aware dynamic routing (GeMoE) against several established dynamic routing baselines on the MolmoE-1B-7B backbone. The performance of existing dynamic routing strategies is suboptimal in native MoE models. Methods such

Table 2: Performance comparison of various static Top-k routing on MolmoE-1B-7B [14] and DeepSeekVL2-Tiny-1B-3B [44].

Method	$N_A \downarrow$	$Avg_k \downarrow$	MMB	POPE	SQA^I	VQA^T	GQA	MM-Vet	$Avg_P \uparrow$
MolmoE-1B-7B (64Top8)									
Top-8	1.58B	8.00	**58.24**	85.33	**87.50**	**56.37**	**50.89**	**45.60**	**63.99**
Top-7	1.48B	7.00	57.30	84.79	86.51	56.07	50.76	43.90	63.22
Top-6	1.38B	6.00	53.86	84.55	85.72	55.34	50.21	43.40	62.18
Top-5	1.28B	5.00	48.11	84.43	84.09	51.95	48.96	42.90	60.07
GeMoE (Ours)	1.32B	5.43	56.70	**88.38**	86.98	54.88	50.64	44.10	63.61
DeepSeek-VL2-Tiny-1B-3B (64Top6)									
Top-6	1.17B	6.00	**69.56**	89.07	**88.99**	**70.16**	**60.84**	**49.10**	**71.29**
Top-5	1.13B	5.00	69.39	88.90	88.55	69.82	59.76	48.60	70.84
Top-4	1.09B	4.00	68.87	88.70	87.24	68.72	59.63	48.50	70.28
Top-3	1.06B	3.00	67.75	88.39	87.10	67.75	59.12	47.10	69.54
GeMoE (Ours)	1.08B	3.55	68.72	**89.41**	88.78	69.24	60.37	48.89	70.90

as DYNMoE, AdaMoE, and MoE++ exhibit significant performance gaps when compared to Top-8, despite using an average of 6.13 to 7.03 experts. In contrast, Top-p suffers a notable performance decline despite using an average of 12.65 experts. In contrast, our approach, which uses the fewest experts on average (5.43), results in only a 0.38% performance drop compared to Top-8, while consistently ranking in the Top-2 for performance across all datasets. This suggests that in MoE models, more experts do not necessarily lead to better performance. An excessive number of noisy experts can hinder model performance, while the optimal number of experts not only preserves but also enhances expert utilization. This finding further supports the conclusions drawn in Section. 3.

Comparison with Various Static Top-k Routing. We compare our method against various static Top-k routing strategies on two backbone models of different scales: MolmoE-1B-7B and DeepSeek-VL2-Tiny-1B-3B. As shown in Tab. 2 and Fig. 4, our method consistently demonstrates strong model performance and higher sparsity across both backbone models. Specifically, our approach ranks second only to Top-8, with only 0.38% and 0.39% performance drops, despite using 32.1% and 40.8% fewer experts than Top-8, respectively. This result indicates that while Top-8 achieves the highest performance, it suffers from low expert utilization and fails to strike an optimal balance between model complexity and performance. In contrast, GeMoE offers a better trade-off, maintaining high performance while improving expert utilization and the efficient sparsity of the MoE model. Additionally, the performance of MolmoE-1B-7B is lower than that of DeepSeek-VL2-Tiny-1B-3B. We attribute this discrepancy to differences in the distribution of training data. Specifically, we assume that the experts in the native MoE model already demonstrate strong performance, so we do not fine-tune them. However, the pretraining data distributions for MolmoE-

Table 3: Efficiency comparison of various routing strategies on an NVIDIA A800-40G.

Method	$\mathbf{Avg}_k \downarrow$	Param↓ (B)	Memory↓ (GB)	Inference FLOPs↓ (GFLOPs/token)	Throughput↑ (token / second)	Wall-clock Time↓ (second /sample)
MolmoE-1B-7B	8.00	7.05	32.95	80.65	1709	0.169
DYNMoE	6.19	7.05	32.95	72.58	1757	0.162
Top-p	12.65	7.05	32.95	98.44	1643	0.178
AdaMoE	6.13	7.05	32.95	71.22	1786	0.161
MoE++	7.03	7.05	32.95	75.45	1741	0.165
GeMoE (Ours)	5.43	7.05	32.95	**68.45**	**1828**	**0.152**

1B-7B and DeepSeek-VL2-Tiny-1B-3B may differ, with the latter's distribution being closer to that of the test set.

Comparison of Inference Efficiency with State-of-the-Art MoE Routing Strategies. We evaluate the inference efficiency of the proposed GeMoE compared to MolmoE-1B-7B and other state-of-the-art routing strategies. The results are shown in Tab. 3. In terms of memory consumption, all methods use 32.95 GB, the same as MolmoE-1B-7B, indicating that the expert assignment predictor in GeMoE introduces negligible overhead. GeMoE also delivers significant computational benefits, reducing inference FLOPs per token by 15.2% (from 80.65 to 68.45 GFLOPs), increasing throughput by 6.5% (1828 vs. 1709 tokens/s), and reducing wall-clock time per sample by 10.1% (0.152 vs. 0.169 s). Additionally, our method exhibits similar GPU memory usage during training compared to the baselines. These results demonstrate that GeMoE accelerates MoE inference without increasing memory usage, proving its efficiency.

4.3 Ablation Studies

Ablation Study of $\mathcal{L}_{mono}$ and Correlation Between Entropy and k. The core of our method lies in determining the number of experts based on the token-level distribution of gating entropy, ensuring a positive correlation between the number of activated experts and gating entropy. To validate the effectiveness of this module, we conduct two experiments. In the first experiment, we remove gating entropy as a guiding proxy and allow the expert number predictor to learn independently. In the second experiment, we reverse the constraint, setting a negative correlation instead of a positive one. As shown in Tab. 4, without the guidance of gating entropy, the predicted number of experts reduced from 5.45 to 4.49, leading to a significant performance drop (from 64.77% to 60.03%). In the second experiment, when the correlation between entropy and k becomes negative, the predicted number of experts reduced further, from 5.45 to 2.86, resulting in a greater decline in performance (from 64.77% to 51.57%). The ablation study above validates the positive correlation between gating entropy and the number of experts, demonstrating the effectiveness of the proposed $\mathcal{L}_{\text{mono}}$.

Comparison of Different Values of α in $\mathcal{L}_{\text{mono}}$. We conduct a sensitivity study of the weighting α in $\mathcal{L}_{\text{mono}}$, as shown in Tab. 5. Specifically, we evaluate loss weightings of $\alpha \in \{0.0, 0.5, 1.0, 1.5\}$. The results show that the proposed loss

Table 4: Ablation study of $\mathcal{L}_{\text{mono}}$ and correlation between entropy and k. $+/-$ indicate positive/negative correlation.

$\mathcal{L}_{\text{mono}}$	Corr	$\text{Avg}_k \downarrow$	MMB	SQA^{I}	GQA	$\text{Avg}_P \uparrow$
		4.49	52.83	80.91	46.36	60.03
✓	–	2.86	41.92	72.24	40.54	51.57
✓	+	5.45	**56.70**	**86.98**	**50.64**	**64.77**

Table 5: Comparison of different values of α used in $\mathcal{L}_{\text{mono}}$ on MolmoE-1B-7B.

α	$\text{Avg}_k \downarrow$	MMB	POPE	SQA^{I}	VQA^{T}	$\text{Avg}_P \uparrow$
0.0	4.49	52.83	81.77	80.91	47.65	65.79
0.5	5.56	56.36	86.36	85.02	54.34	70.52
1.0	5.44	**56.70**	**88.38**	**86.98**	**54.88**	**71.74**
1.5	5.45	56.42	87.87	86.12	54.68	71.27

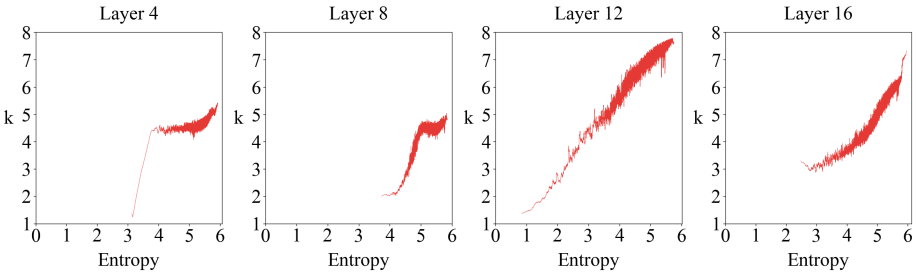


Fig. 6: Visualization of the correlation between gating entropy and the number of activated experts in layer 4, 8, 12 and 16.

function is robust to different weightings, with the best performance achieved on most datasets when $\alpha = 1.0$.

Average Number of Experts Activated in Each MoE Layer. As shown in Fig. 5, GeMoE exhibits a variation in the average number of activated experts across different MoE layers. In general, fewer experts are activated in shallow layers, while more experts are activated in deeper layers. This indicates that shallow layers deal with relatively simple semantics, requiring fewer experts, whereas deeper layers, handling more complex semantics, need more experts to effectively process the information. Moreover, the number of expert activations in each layer remains consistently below 8, which helps reduce the computational load. This results in an average 32% reduction in the number of activated experts.

Correlation between Gating Entropy and the Number of Activated Experts. We visualize the correlation between gating entropy and the number of activated experts learned by the expert assignment predictor in layers 4, 8, 12, and 16, with intervals of 4 layers. As shown in Fig. 6, the gating entropy across different layers leads to varying numbers of activated experts, but all layers exhibit a consistent positive correlation. Specifically, layers 4 and 8 have higher initial gating entropy compared to layers 12 and 16. When the gating entropy is small, the number of experts increases rapidly with increasing gating entropy, indicating that gating entropy has a significant impact on expert activation. However, when the gating entropy reaches a certain value, the increase in the number of activated experts gradually weakens. For layers 12 and 16, which have lower initial gating entropy, the increase in the number of experts becomes more

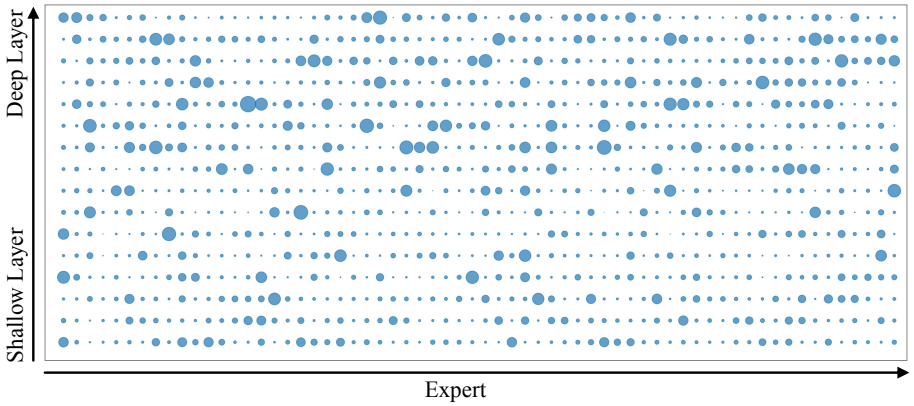


Fig. 7: Average number of tokens activated per expert. A larger circle indicates more activations.

steady. Overall, shallow layers activate fewer experts on average, while deeper layers activate more, consistent with the observations in Fig. 5.

Average Number of Tokens Activated by Experts. Fig. 7 shows the average number of expert activations. As observed, there is a variation in the number of expert activations across different MoE layers. Specifically, fewer experts are activated in the shallow layers, while more experts are activated in the deeper layers. This observation aligns with the findings in Fig. 5 and Fig. 6, suggesting that shallow layers handle simpler semantics, requiring fewer experts, while deeper layers handle more complex semantics, requiring more experts.

5 Conclusion

Our study reveals that dynamic routing in MoE fundamentally addresses a Minimum Description Length (MDL) problem by balancing model complexity and performance. To enable dynamic routing via MDL minimization, we establish a connection between MDL and the gating entropy of token routing. Specifically, when a token has high gating entropy, increasing the number of experts results in significant information gain, thereby reducing MDL, and vice versa. Based on this insight, we introduce a learnable, lightweight expert assignment predictor that assigns an adaptive number of experts to each token according to its gating entropy. To ensure a positive correlation between gating entropy and the number of experts, we incorporate a monotonicity loss. Through comparisons with various dynamic routing strategies across different backbone models and ablation studies, we demonstrate the effectiveness of GeMoE. Our method achieves a favorable balance between model complexity and performance while improving the efficiency of expert utilization.

Acknowledgements

This work is supported in part by National Science Foundation for Distinguished Young Scholars under Grant 62225605, Zhejiang Provincial Natural Science Foundation of China under Grant LD24F020016, Ningbo Science and Technology Special Projects under Grant 2025Z028, and the Department of Human Resources and Social Security of Xinjiang Uygur Autonomous Region for the financial support of the 2024 "Tianchi Talents" Introduction Program (Distinguished Expert Project).

References

1. Aminabadi, R.Y., Rajbhandari, S., Awan, A.A., Li, C., Li, D., Zheng, E., Ruwase, O., Smith, S., Zhang, M., Rasley, J., et al.: Deepspeed-inference: enabling efficient inference of transformer models at unprecedented scale. In: SC22: International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 1–15. IEEE (2022)
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, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. arXiv preprint arXiv:2308.12966 **1**(2), 3 (2023)
4. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
5. 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)
6. Bao, H., Wang, W., Dong, L., Liu, Q., Mohammed, O.K., Aggarwal, K., Som, S., Piao, S., Wei, F.: Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. Advances in neural information processing systems **35**, 32897–32912 (2022)
7. Cai, C., Yang, L., Chen, K., Yang, F., Li, X.: Long-tailed distribution-aware router for mixture-of-experts in large vision-language model. arXiv preprint arXiv:2507.01351 (2025)
8. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: Large language model as a unified interface for vision-language multi-task learning. arxiv 2023. arXiv preprint arXiv:2310.09478 (2023)
9. Chen, W., Zhou, Y., Du, N., Huang, Y., Laudon, J., Chen, Z., Cui, C.: Lifelong language pretraining with distribution-specialized experts. In: International Conference on Machine Learning. pp. 5383–5395. PMLR (2023)
10. Chen, Z., Wang, Z., Wang, Z., Liu, H., Yin, Z., Liu, S., Sheng, L., Ouyang, W., Qiao, Y., Shao, J.: Octavius: Mitigating task interference in mllms via lora-moe. arXiv preprint arXiv:2311.02684 (2023)
11. 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)

12. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
13. Dai, D., Deng, C., Zhao, C., Xu, R., Gao, H., Chen, D., Li, J., Zeng, W., Yu, X., Wu, Y., et al.: Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1280–1297 (2024)
14. Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J.S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., et al.: Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 91–104 (2025)
15. Du, N., Huang, Y., Dai, A.M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A.W., Firat, O., et al.: Glam: Efficient scaling of language models with mixture-of-experts. In: International conference on machine learning. pp. 5547–5569. PMLR (2022)
16. Eliseev, A., Mazur, D.: Fast inference of mixture-of-experts language models with offloading. arXiv preprint arXiv:2312.17238 (2023)
17. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research **23**(120), 1–39 (2022)
18. Gou, Y., Liu, Z., Chen, K., Hong, L., Xu, H., Li, A., Yeung, D.Y., Kwok, J.T., Zhang, Y.: Mixture of cluster-conditional lora experts for vision-language instruction tuning. arXiv preprint arXiv:2312.12379 (2023)
19. Guo, Y., Cheng, Z., Tang, X., Tu, Z., Lin, T.: Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. arXiv preprint arXiv:2405.14297 (2024)
20. Huang, Q., An, Z., Zhuang, N., Tao, M., Zhang, C., Jin, Y., Xu, K., Chen, L., Huang, S., Feng, Y.: Harder task needs more experts: Dynamic routing in moe models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12883–12895 (2024)
21. 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)
22. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural computation **3**(1), 79–87 (1991)
23. Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., Casas, D.d.l., Hanna, E.B., Bressand, F., et al.: Mixtral of experts. arXiv preprint arXiv:2401.04088 (2024)
24. Jin, P., Zhu, B., Yuan, L., Yan, S.: Moe++: Accelerating mixture-of-experts methods with zero-computation experts. arXiv preprint arXiv:2410.07348 (2024)
25. Komatsuzaki, A., Puigcerver, J., Lee-Thorp, J., Ruiz, C.R., Mustafa, B., Ainslie, J., Tay, Y., Dehghani, M., Houlsby, N.: Sparse upcycling: Training mixture-of-experts from dense checkpoints. arXiv preprint arXiv:2212.05055 (2022)
26. Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., Chen, Z.: Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668 (2020)
27. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. In: Proceedings of the 2023 conference on empirical methods in natural language processing. pp. 292–305 (2023)

28. Li, Y., Chen, X., Jiang, S., Shi, H., Liu, Z., Zhang, X., Deng, N., Xu, Z., Ma, Y., Zhang, M., et al.: Uni-moe-2.0-omni: Scaling language-centric omnimodal large model with advanced moe, training and data. arXiv preprint arXiv:2511.12609 (2025)
29. Li, Y., Jiang, S., Hu, B., Wang, L., Zhong, W., Luo, W., Ma, L., Zhang, M.: Uni-moe: Scaling unified multimodal llms with mixture of experts. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
30. Lin, B., Tang, Z., Ye, Y., Huang, J., Zhang, J., Pang, Y., Jin, P., Ning, M., Luo, J., Yuan, L.: Moe-llava: Mixture of experts for large vision-language models. IEEE Transactions on Multimedia (2026)
31. Liu, F., Lin, K., Li, L., Wang, J., Yacoob, Y., Wang, L.: Aligning large multi-modal model with robust instruction tuning. arXiv preprint arXiv:2306.14565 **2**(3), 6 (2023)
32. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
33. 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)
34. Liu, Z., Luo, J.: Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. arXiv preprint arXiv:2405.00361 (2024)
35. 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)
36. Puigcerver, J., Riquelme, C., Mustafa, B., Houlsby, N.: From sparse to soft mixtures of experts. arXiv preprint arXiv:2308.00951 (2023)
37. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
38. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
39. Shen, S., Hou, L., Zhou, Y., Du, N., Longpre, S., Wei, J., Chung, H.W., Zoph, B., Fedus, W., Chen, X., et al.: Mixture-of-experts meets instruction tuning: A winning combination for large language models. arXiv preprint arXiv:2305.14705 (2023)
40. 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)
41. Song, C., Zhao, W., Han, X., Xiao, C., Chen, Y., Li, Y., Liu, Z., Sun, M.: Blockffn: Towards end-side acceleration-friendly mixture-of-experts with chunk-level activation sparsity (2025), https://arxiv.org/abs/2507.08771
42. Wang, Z., Zhu, J., Chen, J.: Remoe: Fully differentiable mixture-of-experts with relu routing. arXiv preprint arXiv:2412.14711 (2024)
43. Wei, T., Zhu, B., Zhao, L., Cheng, C., Li, B., Lü, W., Cheng, P., Zhang, J., Zhang, X., Zeng, L., et al.: Skywork-moe: A deep dive into training techniques for mixture-of-experts language models. arXiv preprint arXiv:2406.06563 (2024)
44. Wu, Z., Chen, X., Pan, Z., Liu, X., Liu, W., Dai, D., Gao, H., Ma, Y., Wu, C., Wang, B., et al.: Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. arXiv preprint arXiv:2412.10302 (2024)

45. Xue, F., Zheng, Z., Fu, Y., Ni, J., Zheng, Z., Zhou, W., You, Y.: Openmoe: An early effort on open mixture-of-experts language models. arXiv preprint arXiv:2402.01739 (2024)
46. Xue, L., Fu, Y., Lu, Z., Mai, L., Marina, M.: Moe-infinity: Offloading-efficient moe model serving. arXiv e-prints pp. arXiv–2401 (2024)
47. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
48. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities. arXiv preprint arXiv:2308.02490 (2023)
49. Yue, T., Guo, L., Cheng, J., Gao, X., Huang, H., Liu, J.: Ada-k routing: Boosting the efficiency of moe-based llms. In: The Thirteenth International Conference on Learning Representations (2024)
50. Zeng, Z., Miao, Y., Gao, H., Zhang, H., Deng, Z.: Adamoe: Token-adaptive routing with null experts for mixture-of-experts language models. In: Findings of the Association for Computational Linguistics: EMNLP 2024. pp. 6223–6235 (2024)
51. Zhang, Y., Zhang, R., Gu, J., Zhou, Y., Lipka, N., Yang, D., Sun, T.: Llavar: Enhanced visual instruction tuning for text-rich image understanding. arXiv preprint arXiv:2306.17107 (2023)
52. Zhong, S., Gao, S., Huang, Z., Wen, W., Žitnik, M., Zhou, P.: Moextend: Tuning new experts for modality and task extension. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop). pp. 494–505 (2024)
53. Zhou, Y., Zhao, Z., Li, H., Du, S., Yao, J., Zhang, Y., Wang, Y.: Exploring training on heterogeneous data with mixture of low-rank adapters. arXiv preprint arXiv:2406.09679 (2024)