Neuronal Redundancy Reduction and Collaboration for Parameter-Efficient Mixture-of-Experts

Anonymous ACL submission

Abstract

Mixture of Experts (MoE) has demonstrated significant potential in scaling up the capacity of large models while maintaining low computational costs. Recent work has focused on reducing expert-level redundancy by designing various token allocation strategies within gating functions. However, the issue of knowledge redundancy persists at the fine-grained neuron level within each expert. Furthermore, the intricate internal relationships between experts have not been thoroughly explored. Leveraging the concept of the information bottleneck (IB), we propose an IB-based MoE for parameter-efficient fine-tuning, which reduces neuron-level redundancy within each expert and fosters collaboration among all experts. Specifically, we first devise a sparse neuronal activation strategy to reveal the task-specific characteristics of each expert. Subsequently, we apply a diversity constraint between the experts to maximize knowledge difference, enabling them to process tasks in a more closely cooperative manner. Experimental results on six well-known Natural Language Understanding (NLU) datasets show the great advantages of our proposed approach. In particular, we achieve comparable performance with less than 0.5% training parameters of the base model, which indicates more efficient MoE with lower computational cost.

1 Introduction

Large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023; Hoffmann et al., 2022) demonstrate impressive results across various Natural Language Processing (NLP) tasks (Wei et al., 2021, 2022), benefiting from their outstanding capabilities in language comprehension and text generation. Nevertheless, the significant computational resources required for training and inference across diverse tasks present ongoing challenges. Thus, various parameter-efficient methods are proposed to reduce the cost of computation, such as P-tuning

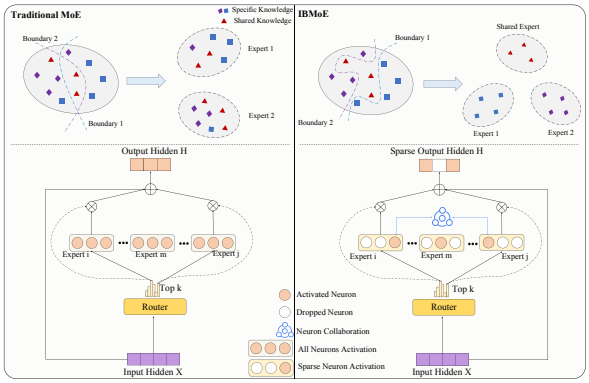

Figure 1: The differences between traditional MoE and our IBMoE. To build more explicit knowledge boundaries among experts and make each expert specialized on certain tasks, our IBMoE performs redundancy reduction in the neuron-level for each expert and enables all experts to work collaboratively with each other.

(Liu et al., 2021), Low-Rank Adaption (LoRA) (Hu et al., 2021) and Mixture of Experts (MoE) (Shazeer et al., 2017). MoE is a conditional computation framework that divides a dense model into multiple specialized experts. It employs a gating mechanism to dynamically select the most relevant experts for computing inputs, which maintains a constant computational cost even as the model scales continuously grow in size.

One key challenge in MoE is how to make the experts specialized on certain tasks. Liu et al. (2024); Zhang et al. (2024) designed specialized routing mechanisms for the gate unit, which allocated tokens to the experts according to the task features. Zuo et al. (2022) and Zhu et al. (2024) segmented the weights of dense models to obtain parameter separated experts, and invoke certain experts by measuring the correlation with tasks. To improve the efficiency of MoE, Dai et al. (2024) devised a shared expert to preserve common knowledge and other experts to focus on specific knowledge, which improves the parameter utilization. However, these methods mainly aim to improve expert-level specialization, while the internal fine-grained neurons

of each expert are not well studied.

Another issue is that most previous methods mainly focus on the design of each expert, while the interactions or collaborations among the experts are rarely investigated. Traditional methods incorporate an auxiliary loss for experts, and expect to activate different experts for different tasks. Whereas, the knowledge boundary between two experts is not clear. As shown in Figure 1, the knowledge in expert 1 and expert 2 has overlaps. Thus, it is challenging to make the experts collaborate efficiently with each other.

To address these problems, we propose IBMoE, which consists of sparse neuron activation and collaboration based on the information bottleneck. Specifically, sparse neuron activation aims to reduce the neuron-level redundancy of each expert. We set a mask for each expert, and only a small proportion of neurons will be activated by the mask, thus the hidden representation is also sparse. The parameters are optimized with the information bottleneck theory, which minimizes the mutual information between the sparse hidden representation and the original input to remove the task irrelevant information, and maximizes the mutual information between the sparse hidden representation and the output to reserve the task-related information. Additionally, we devise a neuron collaboration module to increase the divergence in the distributions of the activated neurons, encouraging more efficient collaborations among neurons.

We conduct extensive experiments on six well-known natural language understanding benchmarks. The major contributions of our work can be summarized as follows:

- In this paper, we propose a parameter-efficient method called IBMoE, which captures the relationships among multiple experts to achieve neuron-level specialization via redundancy reduction and collaboration.
- We propose the sparse neuron activation to retain the task-specific knowledge while reducing irrelevant information by masking out the unimportant neurons via information bottleneck. Additionally, a neuron collaboration module is introduced to decouple shared and specific knowledge by modeling the interaction among the experts.
- We conduct extensive experiments to validate the effectiveness and superiority of the IBMoE method. The experiments indicate that IBMoE achieves better performance compared to the baselines, even when configured with less than 0.5% of the trainable parameters of the base model and with only 8% of the parameters activated.

2 Related work

2.1 Mixture of Experts

Mixture of Experts (MoE) is a framework that utilizes a gate unit to allocate different inputs to the most relevant expert modules, which reduces computational resources. To the best of our knowledge, previous MoE-related works can be divided into three categories.

Larger model with constant cost. Early efforts, such as GShard (Lepikhin et al., 2020) and Switch Transformer (Fedus et al., 2022), achieve this by replacing every other MLP layer with MoE layers and allocating tokens to experts using a top-k mechanism. DeepSeekMoE (Dai et al., 2024) introduces shared experts and smaller experts to achieve more fine-grained specialization.

Reducing cost through knowledge separation. Llama-MoE (Zhu et al., 2024) segments the weights of the dense model to initialize the experts by measuring their relevance to different tasks. MoeBERT (Zuo et al., 2022) additionally introduces a distillation loss to extract knowledge from the dense model to the sparse MoE architecture.

Enhancing performance via adapter experts. MoLoRA (Zadouri et al., 2023) splits the LoRA adapters into multiple fine-grained experts, called LoRA-experts, allowing tiny experts to handle different tasks. Further, MoLA (Gao et al., 2024) introduces a method for layer-wise expert allocation to reduce knowledge redundancy. MoR1E (Liu et al., 2024) proposes a task-level routing mechanism to achieve expert specialization in rank-1 LoRA. However, research on neuron-level knowledge separation is still rare. Our IBMoE effectively measures the differences between tasks from a feature perspective.

2.2 Parameter Efficient Fine-Tuning

Parameter Efficient Fine-Tuning (PEFT) aims to reduce the expensive computational cost of training base models to adapt to various downstream tasks. In general, previous studies can be divided into adapter-based and prompt-based methods.

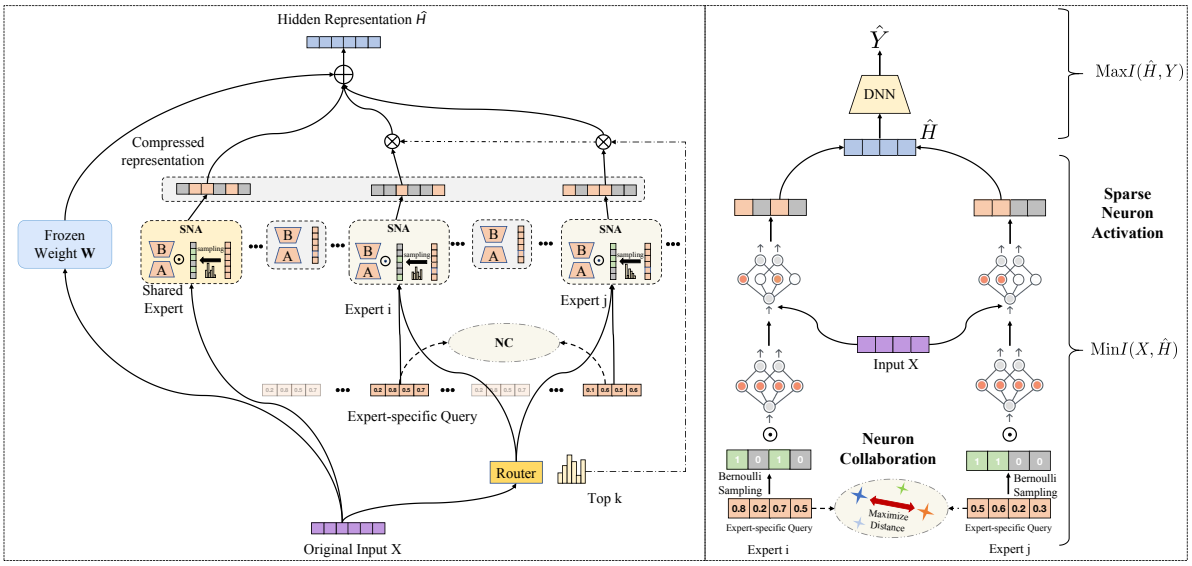

Figure 2: Overall framework of IBMoE, which consists of two parts: sparse neuron activation (SNA) and neuron collaboration (NC). SNA learns a neuron-level mask vector via information bottleneck to retain the important information while forgetting the unrelated information. Then, we propose the NC strategy to reduce redundant knowledge by learning interactions among experts.

Prompt-based methods add a set of input-level learnable prompts to improve performance by optimizing the task representation. Prompt-Tuning (Lester et al., 2021) and P-Tuning (Liu et al., 2021) insert a trainable set of discrete or continuous prompt vectors in the embedding layer to learn a better representation. However, their performance is typically affected by context length and model size. Prefix-Tuning (Li and Liang, 2021) inserts trainable virtual tokens into each transformer layer. Bitfit (Zaken et al., 2021) achieves comparable performance by updating only the bias parameters.

Adapter-based methods add trainable submodules, called adapters, to the dense model. The model is fine-tuned by updating the adaptor while keeping the base model frozen. Specifically, IA3 (Liu et al., 2022) adds three sets of learnable vectors in the key, value, and linear layers. Llama-Adapter (Zhang et al., 2023) introduces learnable adaptation prompts and zero-init attention to adaptively inject new instruction cues into Llama while effectively retaining its pre-trained knowledge. LoRA (Hu et al., 2021) uses two decomposed low-rank matrices to perform incremental updates. IBMoE utilizes the concept of PEFT to effectively reduce the training cost of traditional MoE methods while preserving the task specialization advantage of MoE.

2.3 Information Bottleneck

Information Bottleneck (IB), a representation compression technique based on mutual information theory, is used to explore the limits of dropping invalid information while preserving task-relevant information (Tishby et al., 2000). Previous studies have shown that controlling the unnecessary information in the hidden layer through the information bottleneck can effectively control the generalization error in deep learning (Kawaguchi et al., 2023). IB-related work on deep neural networks (DNNs) has been in development since 2015 (Tishby and Zaslavsky, 2015; Strouse and Schwab, 2017). Currently, several works based on DeepVIB (Alemi et al., 2016) have achieved convincing results in various domains, such as graph representation learning (Yu et al., 2021), adversarial training (Chen et al., 2021), and network compression (Lorenzen et al., 2021), etc. For example, Sheng et al. proposed an IB based temporal variational mask module to compress invalid information from the raw lip image (Sheng et al., 2022). Inspired by these works, we introduce IB into MoE, enabling experts to focus only on the most relevant features of the task. Additionally, we design a diversity constraint on IB to effectively improve the knowledge difference between experts.

3 Preliminaries

Mixture of Experts: In general, the MoE architecture consists of N feed-forward neural networks, denoted as $\{E\}_{i=1}^{N}$, which process various inputs through a gating mechanism. Specifically, the gating mechanism is a function equipped with the

trainable weight parameters $W_r \in \mathbb{R}^{d_{in} \times N}$. For each input $x \in \mathbb{R}^{d_{in} \times 1}$, the gate function maps it to an N-dimensional discrete probability vector, which represents the contribution of each expert. A routing algorithm, such as the top-k or soft routing method, is then used to allocate the input to the experts. In this paper, we select the top-k experts with the highest contributions to process the input. The output y_i at i-th token can be calculated as:

$$G_i = TopK(softmax(W_r^\top x_i) \tag{1}$$

$$y_i = \sum_{e=1}^{N} G_{i,e} E_e(x_i) \tag{2}$$

where $G_{i,e}$ represents the contribution of e-th expert at i-th token.

Low-Rank Adaptation: Inspired by the natural low intrinsic dimension, LoRA introduces low-rank decomposition for optimizing the weight matrix of pre-trained large language models (LLMs) to reduce the number of training parameters. Let the weight matrix of a pre-trained linear layer be $W_0 \in \mathbb{R}^{d_{in} \times d_{out}}$. LoRA learns two low-rank matrices $A \in \mathbb{R}^{r \times d_{in}}$ and $B \in \mathbb{R}^{d_{out} \times r}$ to update W_0, where $r \ll \min(d_{in}, d_{out})$. During training, W_0 is frozen and only matrices A and B are updated. Given an input x_i, the output of the LoRA layer can be expressed as:

$$\begin{aligned} h_i &= W_0^\top x_i + \frac{\alpha}{r} BAx_i \\ &= W_0^\top x_i + \triangle W x_i \end{aligned} \tag{3}$$

where α is a scale hyper-parameter.

MoE and LoRA: Some studies integrate the advantages of LoRA and MoE by constructing multiple LoRA adapters as experts, known as LoRA-MoE, to achieve expert specialization on different tasks. Combining Equations 2 and 3, we have:

$$h_i = W_0^\top x_i + \sum_{e=1}^{N} G_{i,e} \triangle W_e x_i \tag{4}$$

where $\triangle W_e$ is the weight matrix of e-th LoRA expert.

4 Our Proposed Approach

In this section, we describe the details of the IB-MoE framework with Sparse Neuron Activation and Neuron Collaboration (Figure 2). Specifically, for Sparse Neuron Activation, we introduce the information bottleneck into the MoE framework to reduce knowledge redundancy among the experts. We further design Neuron Collaboration to reduce redundant information among multiple experts by maximizing their diversity.

4.1 Sparse Neuron Activation

We design Sparse Neuron Activation (SAN) to compress the neurons for neuron specialization with information bottleneck. Similar to the idea of task-level sparse activation in MoE, where the LLM's knowledge is not fully utilized in processing a small task, the neurons of an expert can also be sparsely activated for its specific task. Actually, not all neurons of an expert are equally important in the processing of task inputs, which means that fine-grained expert specialization can be further improved by dropping dispensable neurons. Consequently, we expect the experts to focus only on processing the most relevant information for the tasks. Following the mutual information theory of information bottleneck. Let the inputs and labels be denoted as X and Y, respectively. The standard objective function is expressed as:

$$L_{ib} = \beta I(X, \hat{H}) - I(\hat{H}, Y) \tag{5}$$

where $\hat{H}$ is the compressed representation of X, $I(.,.)$ denotes mutual information, and β is a hyper-parameter that used to controls the compression of X. The experts are required to discard information from X that is not crucial for their tasks, while retaining sufficient task-related information to predict the label Y.

The former term $I(X, \hat{H})$ represents the mutual information between X and $\hat{H}$. The traditional objective function is calculated as:

$$I(X, \hat{H}) = \mathop{\mathbb{E}}_{(x,\hat{h}) \sim p(x,\hat{h})} \log \frac{p(\hat{h}|x)}{p(\hat{h})} \tag{6}$$

Considering the intractability of computing $p(\hat{h})$ directly, we follow the concept of Variational Information Bottleneck (Alemi et al., 2016), which utilizes a variational distribution $r(\hat{h})$ to approximate $p(\hat{h})$. This allows us to optimize the upper bound of $I(X, \hat{H})$. Based on the fact that $KL[p(\hat{h})||r(\hat{h})] \geq 0$, we have:

$$I(X, \hat{H}) \leq \mathop{\mathbb{E}}_{x \sim p(x)} KL[p(\hat{h}|x)||r(\hat{h})] \tag{7}$$

Then, for an expert E_i, we first add a task-specific query with a set of learnable parameters

$V_i \in \mathbb{R}^{d_{out}}$ for its $\triangle W_i$ to represent the importance of the neurons. Before the forward process, E_i uses V_i to generate a binary mask $M_i = [m_1, m_2, ..., m_{d_{out}}]^\top \in \{0,1\}^{d_{out}}$ via a Bernoulli-based importance sampling, where M_i indicates which neurons should be dropped. M_i is generated as follows:

$$\hat{V}_i = \frac{V_i - \min(V_i) + \epsilon}{\max(V_i) - \min(V_i) + \epsilon} \tag{8}$$

$$M_i = Ber(\hat{V}_i) \tag{9}$$

where $Ber(.)$ is a function used for Bernoulli sampling, and ϵ is small positive number to prevent division by zero errors. Then, the compressed hidden representation can be expressed as:

$$\triangle \hat{W}_i = M_i \circ \triangle W_i \tag{10}$$

$$\hat{h_i} = W_0^\top x_i + \sum_{e=1}^{N} \frac{d_{out} G_{i,e} \triangle \hat{W}_e x_i}{sum(M_e)} \tag{11}$$

where $\circ$ denotes element-wise multiplication at the column level, and $sum(.)$ is a summation function.

In this paper, Bernoulli sampling is used as the prior distribution, denoted as $r(\hat{h}) = Ber(\pi)$. We aim for the expected number of activated neurons to follow a Bernoulli distribution with a probability of π, which enables controllable compression of invalid information. Compared with h, the computation of $\hat{h}$ additionally depends on the query V of each expert, where V is independent of x. Consequently, we have $p(\hat{h}|x) = p(M|V)$ and $r(\hat{h}) = r(M)$. Therefore, a sparsity constraint is designed to minimize $KL[p(\hat{h}|x)||r(\hat{h})]$:

$$L_s = \sum_{i=1}^{N} [\bar{V}_i \cdot \log \frac{\bar{V}_i}{\pi} + (1 - \bar{V}_i) \cdot \log \frac{1 - \bar{V}_i}{1 - \pi}] \tag{12}$$

where $\bar{V}_i$ represents the mean value of V_i.

Similar to $I(X, \hat{H})$, the later term $I(\hat{H}, Y)$ denotes the mutual information between $\hat{H}$ and Y, and the objective function is expressed as:

$$I(\hat{H}, Y) = \mathop{\mathbb{E}}_{(\hat{h},y)\sim p(\hat{h},y)} \log \frac{p(y|\hat{h})}{p(y)} \tag{13}$$

Then, a variational distribution $q(y|\hat{h})$ is used to address the intractability of $p(y|\hat{h})$. Based on the fact that $KL[p(y|\hat{h})||q(y|\hat{h})] \geq 0$, the lower bound of $I(\hat{H}, Y)$ is calculated as:

$$I(\hat{H}, Y) \geq \mathop{\mathbb{E}}_{(\hat{h},y)\sim p(\hat{h},y)} \log q(y|\hat{h}) + C \tag{14}$$

where C is a constant representing the entropy of Y.

Note that $q(y|\hat{h})$ is typically achieved by a network $q_\theta(\hat{h})$ used for the next token prediction task, with the loss function being cross-entropy. The loss function is expressed as:

$$L_p = \sum_{x_i \in X} -y_i \log(q_\theta(p_\phi(x_i))) \tag{15}$$

where $p_\phi(.)$ represents a learnable neural network, such as multiple MoE layers.

4.2 Neuron Collaboration

In this section, we propose Neuron Collaboration (NC) to maximize the knowledge diversity of the experts. As previously mentioned, there is inherent similarity among the tasks handled by experts, which leads to knowledge redundancy in traditional MoE frameworks. In IBMoE, we address this issue by introducing a feature-level diversity loss, which controls the knowledge differences among experts and endows them with more specialized knowledge for handling various tasks. Specifically, for two queries $\hat{V}_i$ and $\hat{V}_j$, we aim to minimize their similarity, which encourages experts to focus on different dimensions of feature information by generating masks with diverse distributions. The diversity constraint is expressed as:

$$L_d = \sum_{i,j}^{N} cosine(\hat{V}_i, \hat{V}_j) \tag{16}$$

where $cosine(.,.)$ is a function for computing the cosine similarity between two vectors.

Following (Dai et al., 2024), IBMoE also designates a shared expert E_N to handle task-level common knowledge. According to Equation 11, the forward pass output of the IBMoE layer can be expressed as:

$$\hat{h_i} = W_0 x_i + d_{out}(\sum_{e=1}^{N-1} \frac{G_{i,e} \triangle \hat{W}_e x_i}{sum(M_e)} + \frac{\triangle \hat{W}_N x_i}{sum(M_N)}) \tag{17}$$

We also add an auxiliary loss for the router to ensure differentiated token allocation. Let the $P_r = softmax(W_r^\top X)$, and the objective function is defined as:

$$L_b = e^{[std(mean_t(P_r)) - mean(std_e(P_r))]} \tag{18}$$

where $mean_t(P_r)$ represents the token-wise mean value of P_r, and $std_e(P_r)$ represents the expert-wise standard deviation of P_r.

Combining Equations 12, 15, 16 and 18, the finally objective function is expressed as:

$$L = L_p + \beta L_s + \gamma L_d + \lambda L_b \tag{19}$$

5 Experimental Setup

5.1 Datasets

GLUE. The General Language Understanding Evaluation (GLUE) is a popular benchmark comprising three types of NLP tasks: single-sentence tasks, similarity and paraphrase tasks, and inference tasks. In this paper, we select one task from each category to evaluate the effectiveness of IBMoE. The Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019) is a binary classification task that evaluates the grammatical acceptability of sentences, originating from books and journals of linguistic theory. The Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005) is collected from news articles and is used to judge the equivalence of sentence pairs in terms of paraphrasing. The Recognizing Textual Entailment (RTE) (Wang et al., 2019) is an entailment challenge task, with the dataset constructed from Wikipedia and news.

Question Answering. Three common Question Answering (QA) tasks are used to evaluate our IBMoE. Specifically, ScienceQA (SQA) (Lu et al., 2022) is a question-answering multi-modal dataset collected from primary and high school science curricula. For this task, we remove the multi-modal question-answer samples to adapt them to our text-only LLM. OpenbookQA (OBQA) (Mihaylov et al., 2018) is designed to evaluate the model's ability to solve questions based on given elementary science facts. CommonSenseQA (CSQA) (Talmor et al., 2019) is a multiple-choice task with five options, used to assess the model's performance on questions requiring extensive commonsense knowledge.

5.2 Baselines and Evaluation Metrics

To evaluate the effectiveness of our model, we compare our IBMoE with recent parameter-efficient fine-tuning (PEFT)-based and MoE-based methods. For PEFT-based methods, we select **Prompt Tuning** (Lester et al., 2021), **Llama-Adapter** (Zhang et al., 2023), **LoRA** (Hu et al., 2021). For MoE-based methods, we employ **MoLE** (Wu et al., 2024b), **AdaMoLE** (Liu and Luo, 2024), **MoRAL** (Yang et al., 2024), **PESC** (Wu et al., 2024a), **LoRA-Switch** (Kong et al., 2024) and **MoLA** (Gao et al., 2024).

To comprehensively evaluate the performance of IBMoE, we employ multiple metrics in our experiments. Specifically, for traditional NLU tasks, average accuracy (**ACC**) and activated parameters (**Act Params**) are used to assess the model's generalization performance and parameter usage of the adapters, respectively. Additionally, we evaluate the model's physical performance. **Time** is measured to assess time consumption during inference, while **Peak Memory** indicates the maximum memory consumption of the added adapters.

5.3 Implementation Details

In the experiment, we adopt the LLAMA2-7B model as our backbone to evaluate the performance of IBMoE. Specifically, we implement our proposed algorithm based on hugging-face Transformers[1] and PEFT[2] libraries, and the hyper-parameters set as follows. 1) Training Parameters: The optimizer is set to AdamW, the learning rate is $2e^{-4}$. We set the batch size as 128 and the truncated length as 256. Across all the experiments, the epoch is set to 10 for fine-tuning. 2) LoRA Parameters: We follow the default configuration in LoRA, and the intrinsic LoRA rank of the experts is set to 8. 3) IBMoE Parameters: For the MoE aspect, we apply 5 LoRA experts to every other MLP layer, with one expert designated as a shared expert to handle all inputs. During forward processing, the top-k is set to 2, which selects a specific expert and a shared expert when processing each token. For the information bottleneck aspect, we set $\pi = 0.2$, $\beta = 0.01$, $\gamma = 0.1$, $\lambda = 0.1$, and $\epsilon = 1e^{-10}$, respectively. Notably, the setting of π to 0.2 is crucial as it allows each neuron to have a chance to be updated by the expert. All the experiments are conducted on NVIDIA 4090 GPUs.

Additionally, considering the non-differentiability of Bernoulli sampling during training, we use Gumbel-Softmax reparameterization to approximate the Bernoulli distribution (Jang et al., 2017). The sampling process for $\hat{V}_i$ is computed as follows:

$$M_i = sigmoid\frac{log\hat{V}_i + g}{\tau} \tag{20}$$

where $g = -log(-log(u)), u \sim U(0,1)$, and $\tau =$

[1] https://github.com/huggingface/transformers
[2] https://github.com/huggingface/peft

Methods	Original	Act Params ↓	Datasets (Acc %↑)						Average
			MRPC	CoLA	RTE	SQA	OBQA	CSQA	
Prompt Tuning†	-	-	49.91	59.25	54.17	36.78	37.76	46.20	47.35
Llama-Adapter†	1.2M	1.2M	71.94	47.56	72.93	73.33	73.55	71.80	68.52
LoRA†	164.5M	164.5M	83.13	86.29	85.92	91.01	75.51	77.00	83.14
MoRAL†	172.48M	43.12M	-	-	-	90.74	76.60	76.41	-
PESC†	172.48M	43.12M	-	-	-	90.02	78.40	76.00	-
LoRA-Switch†	172.48M	43.12M	-	-	-	91.39	80.40	79.03	-
MoLE†	34.7M	8.68M	-	-	86.28	88.71	-	77.15	-
MoLA†	107.8M	43.12M	83.48	**86.87**	86.28	92.36	78.95	<u>79.60</u>	84.59
AdaMoLE†	34.7M	dynamic	-	-	87.73	91.00	-	78.71	-
$\text{IBMoE}_{\text{MLP}}$	32.9M	2.62M	<u>85.08</u>	86.13	<u>88.23</u>	<u>92.62</u>	**81.68**	**79.78**	<u>85.54</u>
$\text{IBMoE}_{\text{ATTN}}$	57.3M	4.59M	**86.98**	<u>86.48</u>	**89.31**	**93.23**	<u>81.08</u>	79.08	**86.03**

Table 1: Comparison results of performance and parameter efficiency with other SOTA methods across 6 NLU datasets. All methods are based on the same backbone LLAMA-2 7B. The best and suboptimal results are emphasized in **bold** and <u>underline</u>.

0.5 is a hyper-parameter for controlling sampling randomization.

6 Experimental Results

6.1 Main Results

Table 1 shows the performance of IBMoE against other PEFT and MoE baselines, where † represents the results are provided from the corresponding method. Notably, we provide two versions of IBMoE to evaluate its effectiveness: $\text{IBMoE}_{\text{MLP}}$ denotes the default configuration mentioned above, and $\text{IBMoE}_{\text{ATTN}}$ additionally applies information-bottleneck-based LoRA experts to every other attention layer. Finally, taking into account the randomization of mask generation, we report the mean performance of IBMoE by evaluating it five times on the test datasets.

From the table, we find that $\text{IBMoE}_{\text{ATTN}}$ achieves the highest average performance across all models, demonstrating its superiority and comparability even with less than 1% of the training parameters of LLAMA-2 7B. Specifically, although the prompt-based method and Llama-Adapter have minimal training parameters, there is still a significant performance gap compared to the LoRA-based methods. MoLA assigns fewer experts to lower layers to reduce knowledge redundancy while assigning more experts to higher layers to achieve better expert specialization, but the performance remains limited without explicit specialized constraints. IBMoE not only takes advantage of task-level sparse activation of the experts but also introduces neuron-level sparse activation within the experts' specific tasks to further reduce computational costs, as represented by **Act Params** in Table 1.

6.2 Ablation Studies

To verify the effectiveness of the main components in our proposed IBMoE, we conduct ablation studies (Table 2). First, to evaluate the effectiveness of the neuron collaboration, we remove the L_d to analyze the performance of IBMoE on the RTE dataset. A key factor contributing to this performance drop is that substantial information compression requires a higher degree of expert collaboration. In our work, we set π to 0.2, which means that 80% of the expert's neurons are dropped during the processing of the allocated inputs. In this case, fewer activated neurons correspond to a smaller number of available parameters, requiring complementary knowledge among experts to mitigate the effect of knowledge redundancy. Second, building on the removal of the neuron collaboration, we further remove the sparsity neuron activation. The performance of IBMoE is significantly degraded. This is because the sparsity constraint enables the experts to focus on the most relevant information by maximizing the probability of important neurons being activated. However, in the absence of L_s, the instability in neuron sampling leads to a performance decline to some extent. Third, we remove the balance loss to evaluate its influence on IBMoE's performance. As shown in Table 2,

Model	OBQA	
	Acc (%↑)	Act Params ↓
$\text{IBMoE}_{\text{MLP}}$	81.68	2.62M
$-L_d$	81.44	2.62M
$-L_d - L_s$	80.72	5.16M
$-L_d - L_s - L_b$	78.04	5.29M

Table 2: Ablation results of IBMoE on the OBQA dataset.

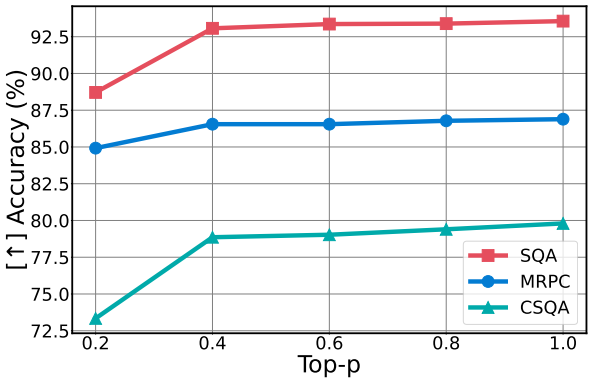

Figure 3: Impact of p for $\text{IBMoE}_{\text{ATTN}}$.

Model	Inference with 1,000 Samples	
	Time (s) ↓	Peak Memory (MB) ↓
MoLA	893.8	1452.6
AdaMoLE	567.8	146.2
$\text{IBMoE}_{\text{MLP}}$	426.3	75.7

Table 3: Results of physical performance.

IBMoE exhibits the worst performance without any constraints. A crucial reason for this is that the imbalance in the routing policy causes some experts to handle a disproportionately large amount of input data. These overworked experts activate more neurons to retain the knowledge needed for various tasks. As a result, in the absence of L_b, IBMoE activates 5.29M parameters during the inference stage.

6.3 Further Analyses

Effect of Sparse Neuron Activation. As mentioned above, each expert drops a large number of neurons by performing importance sampling on V. In this section, we activate neurons using greedy sampling to analyze the impact of neuron activation. Similar to the gating mechanism, we apply top-p sampling on V to activate the most task-related neurons for forward processing. Figure 3 shows the performance of $\text{IBMoE}_{\text{ATTN}}$ with greedy sampling during inference, where top-$p = 0.2$ denotes that the top 20% of neurons are activated. We observe that the performance of the greedy sampling is lower than that of Bernoulli-based importance sampling at an equal activation scale. This is because importance sampling inherently combines neurons of varying importance levels, enabling a global planning of feature selection, whereas greedy sampling focuses on partial information of the feature. As p increases, greedy sampling achieves a comparable performance to importance sampling, indicating that importance sampling effectively balances performance and computational cost.

Analysis of the Physical Performance of IBMoE. We also compare the physical performance of IBMoE with other sparse MoE methods, including MoLA and AdaMoLE, to evaluate its efficiency. Specifically, we use two NVIDIA 4090 GPUs with 24GB of memory to perform inference on 1,000 randomly sampled test set examples, measuring the inference time and peak memory usage of the adapters with a batch size of 8. In Table 3, our proposed method achieves the best performance in both time and memory efficiency, with improvements of 25% in time and 48.5% in memory efficiency compared to the dynamically activated AdaMoLE. We attribute this advantage to the fact that the trained IBMoE significantly reduces computational cost and memory consumption by compressing the weight matrices of LoRA experts.

7 Conclusions and Further Work

In this paper, we propose a novel parameter-efficient MoE method that introduces an information bottleneck framework and multiple constraints to achieve fine-grained reduction of neuronal redundancy, fostering closer collaboration among experts. Extensive experiments on six well-known NLU datasets demonstrate that our method achieves performance comparable to state-of-the-art methods, even with less than 0.5% of the training parameters of the base model. Additionally, an analysis of physical performance shows that our method enhances time and memory efficiency by leveraging the compression characteristics of importance parameters. In future work, we will explore applications in continuous learning by capitalizing on the knowledge specialization inherent in MoE.

8 Limitations

For the sparsity constraint, we simply set the compression ratio $\pi = 0.2$ uniformly for all experts. However, experts in different layers typically need to perform varying levels of feature abstraction on the inputs, which leads to unstable model performance in complex tasks. A potential solution is to design an adaptive compression strategy for each expert, allowing dynamic adjustment of the number of activated neurons, or to vary the number of experts across layers. On the other hand, the diversity constraint requires computing the loss for each pair of experts, which indicates that training efficiency is not always ideal as the number of experts increases.

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, arXiv:1612.00410.

Jiawei Chen, Ziqi Zhang, Xinpeng Xie, Yuexiang Li, Tao Xu, Kai Ma, and Yefeng Zheng. 2021. Beyond mutual information: Generative adversarial network for domain adaptation using information bottleneck constraint. *IEEE Transactions on Medical Imaging*, 41(3):595–607.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, arXiv:2401.06066.

William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 9–16.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The International Journal of Man-Machine Studies*, 23(120):1–39.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. 2024. Higher layers need more lora experts. *arXiv preprint arXiv:2402.08562*, arXiv:2402.08562.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, arXiv:2203.15556.

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

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *Proceedings of 5th International Conference on Learning Representations*.

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. 2023. How does information bottleneck help deep learning? In *Proceedings of the 40th International Conference on Machine Learning*, pages 16049–16096.

Rui Kong, Qiyang Li, Xinyu Fang, Qingtian Feng, Qingfeng He, Yazhu Dong, Weijun Wang, Yuanchun Li, Linghe Kong, and Yunxin Liu. 2024. Lora-switch: Boosting the efficiency of dynamic llm adapters via system-algorithm co-design. *arXiv preprint arXiv:2405.17741*, arXiv:2405.17741.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, arXiv:2006.16668.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, arXiv:2104.08691.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, arXiv:2101.00190.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35(120):1950–1965.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, arXiv:2110.07602.

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

Zefang Liu and Jiahua Luo. 2024. Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. *arXiv preprint arXiv:2405.00361*, arXiv:2405.00361.

Stephan Sloth Lorenzen, Christian Igel, and Mads Nielsen. 2021. Information bottleneck: Exact analysis of (quantized) neural networks. *arXiv preprint arXiv:2106.12912*, arXiv:2106.12912.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, arXiv:2303.08774.

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

Changchong Sheng, Li Liu, Wanxia Deng, Liang Bai, Zhong Liu, Songyang Lao, Gangyao Kuang, and Matti Pietikäinen. 2022. Importance-aware information bottleneck learning paradigm for lip reading. *IEEE Transactions on Multimedia*, 25:6563–6574.

DJ Strouse and David J Schwab. 2017. The deterministic information bottleneck. *Neural computation*, 6(120):1611–1630.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4149–4158.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*, physics/0004057.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, arXiv:2302.13971.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of 7th International Conference on Learning Representations*, pages 6–9.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 24824–24837.

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

Xun Wu, Shaohan Huang, and Furu Wei. 2024b. Mixture of LoRA Experts. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. 2024. Moral: Moe augmented lora for llms' lifelong learning.

Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. 2021. Recognizing predictive substructures with subgraph information bottleneck. *IEEE transactions on pattern analysis and machine intelligence*, 46(3):1650–1663.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. 2023. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444*, arXiv:2309.05444.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, arXiv:2106.10199.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, arXiv:2303.16199.

Rongyu Zhang, Yulin Luo, Jiaming Liu, Huanrui Yang, Zhen Dong, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, Yuan Du, et al. 2024. Efficient Deweahter Mixture-of-Experts with Uncertainty-Aware Feature-Wise Linear Modulation. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pages 16812–16820.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. 2024. Llama-moe: Building mixture-of-experts from llama with continual pre-training. *arXiv preprint arXiv:2406.16554*, arXiv:2406.16554.

Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2022. Moe-bert: from bert to mixture-of-experts via importance-guided adaptation. *arXiv preprint arXiv:2204.07675*, arXiv:2204.07675.