MaskMoE: Boosting Token-Level Learning via Routing Mask in Mixture-of-Experts

Anonymous ACL submission

Abstract

Scaling the size of a model enhances its capabilities but significantly increases computation complexity. Mixture-of-Experts models (MoE) address the issue by allowing model size to scale up without substantially increasing training or inference costs. In MoE, there is an important module called the router, which is used to distribute each token to the experts. Currently, the mainstream routing methods include dynamic routing and fixed routing. Despite their promising results, MoE models encounter several challenges. Primarily, for dynamic routing methods, the dispersion of training tokens across multiple experts can lead to underfitting, particularly for infrequent tokens. Additionally, though fixed routing methods can mitigate that issue, they compromise on the diversity of representations. In this paper, we propose **MaskMoE**, a method designed to enhance token-level learning by employing a routing **mask**ing technique within the **M**ixture-**o**f-**E**xperts model. MaskMoE is capable of maintaining representation diversity while achieving more comprehensive training. Experimental results demonstrate that our method outperforms previous dominant Mixture-of-Experts models in terms of both perplexity (PPL) and downstream task performance.

1 Introduction

Large language models have achieved promising performance on various downstream natural language tasks (Touvron et al., 2023a; Dai et al., 2022; Brown et al., 2020; Anil et al., 2023; Chowdhery et al., 2022; Radford et al., 2019; Rae et al., 2021; Biderman et al., 2023). Moreover, according to the scaling law (Kaplan et al., 2020; Hoffmann et al., 2022), as the model size increases, the model's capabilities will continue to grow. However, for dense language models, the computational costs of continuing to scale up are excessive. In order to further scale up models within computational budgets, sparse activation networks (Child et al., 2019; Du et al., 2022a) receive widespread attention due to their ability to significantly reduce computational costs by using only a part of parameters per input. A widely studied approach is Mixture-of-Experts (MoE) (Lepikhin et al., 2021; Du et al., 2022a; Dai et al., 2024; Fedus et al., 2022; Roller et al., 2021), which trains multiple expert layers but selects only a subset to process specific inputs. Compared to dense networks of the same model size, MoE effectively reduces computational costs and achieves comparable results in both PPL and downstream (Lepikhin et al., 2021; Du et al., 2022a; Dai et al., 2024).

In MoE models, for each input (e.g., a token in the language models), the router in front of the experts needs to decide which experts to feed it in. The commonly used dynamic routing method in MoE is to select the expert with the top k highest confidence based on a probability distribution output by an intermediate layer with learnable parameters that act as the router. Previous works (Lepikhin et al., 2021; Du et al., 2022a; Dai et al., 2024; Fedus et al., 2022) show that MoE models trained based on dynamic routing achieve better performance than dense models with the same amount of training computation.

However, the dynamic routing method introduces a new challenge: the routing fluctuation problem (Dai et al., 2022). It implies that as training moves on, the same token is assigned to different experts in different iterations due to the variation of learnable parameters in the router. That has two negative impacts. **a)** Each expert is trained on only a subset of the occurrences of an identical token, resulting in each expert possibly not learning enough about the token. As (Shazeer et al., 2017) has indicated, when there are too many experts, the MoE model exhibits underfitting, and even shows higher PPL than its counterparts with fewer experts. The routing fluctuation problem has

*Corresponding authors.

relatively less impact on the learning of frequent tokens, as those tokens have sufficient occurrences to ensure that each expert receives adequate training. On the contrary, for infrequent tokens, the routing fluctuation problem can disperse each of them across various experts, which probably leads to underfitting for them. **b)** The knowledge each expert learns from the corresponding occurrence subset w.r.t an identical token is hard to share among experts. Then the underfitting of experts for some tokens, together with the lack of knowledge sharing among experts, can negatively impact the model's performance (Zhao et al., 2024). Both issues above hinder scaling the number of experts for an MoE model. Because, as the number of experts increases while keeping the training data unchanged, the average number of token occurrences assigned to each expert further decreases, exacerbating the issues.

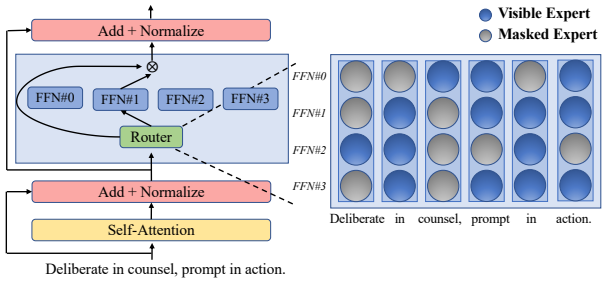


Figure 1: Illustration of random routing masking method. Frequent tokens have more visible experts, while infrequent tokens have fewer visible experts.

To alleviate the issue of routing fluctuations, fixed routing method (Roller et al., 2021; Dai et al., 2022) is proposed, e.g., Hash Layer. Specifically, Hash Layer (Roller et al., 2021) pre-assigns each token to a fixed expert. For instance, as shown in Figure 1, all occurrences of the token "deliberate" can only be routed to the expert FFN#2. The fixed routing method to send all the occurrences of a token to the same expert is advantageous for the token's thorough learning, especially for infrequent tokens. However, fixed routing based on the Hash Layer represents a deficiency in learning representation diversity, especially for those frequent tokens. Previous works (He et al., 2023; Yang et al., 2019) show that increasing the number of optional experts for all occurrences of an identical token can help the diversity of learned representations. And frequent tokens probably require to be encoded by more experts due to their broader range of usage (Koranda et al., 2018). For instance, the token "play" appears in various contexts and owns different semantic meanings, and thus having a more diverse representation helps to distinguish them.

Therefore, based on the observations above, we need a fixed routing strategy to alleviate the underfitting problem for infrequent tokens, while also requiring more experts for frequent ones to maintain the representational diversity. To meet the requirements above, in this paper, we propose a routing method called MaskMoE, which includes a **routing masking** method that adjusts the number of visible experts for tokens of different frequencies, by generating a masking vector for each token in the vocabulary before training. Specifically, for each infrequent token, the proposed MaskMoE employs routing masking to retain only one visible expert to which the token can be routed, in each MoE layer. For each frequent token, the proposed MaskMoE allow more visible experts for it in each MoE layer. MaskMoE can enhance token-level learning through such a routing mask. Specifically, MaskMoE makes the model learn more intensively about infrequent tokens by routing each to an identical expert in every MoE layer, enabling the expert to be adequately trained w.r.t that token. Meanwhile, MaskMoE maintains the diversity of representations for frequent tokens by still having multiple experts available for routing, which benefits their representation diversity (He et al., 2023; Yang et al., 2019). Despite frequent tokens being routed to multiple experts, their sufficient quantities can still ensure thorough training. Our experimental results indicate that MaskMoE outperforms previous MoE models, in terms of either PPL or downstream task performance.

2 Background

In this section, we review language models and Mixture-of-Experts, with more related work located in Appendix A.

2.1 Reviewing the Language Models

Given a tokenized input sequence $\mathbf{x} = (x_1, x_2, ..., x_T)$ consisting of T tokens, a language model generates a probability distribution $\mathbf{p}$ over the vocabulary as output for each token. In nearly all implementations, the Cross-Entropy loss is used as the loss function to maximize the predicted probability $\mathbf{P}^{x_i}_{\cdot,i}$ w.r.t the ground-truth token x_i. The training loss $\mathcal{L}_{lm}$ of the generative language model

can be formulated as follows:

$$\begin{aligned} \mathcal{L}_{lm} &= -\sum_{i=2}^{T} \log(\mathbf{P}_{\cdot,i}^{x_i}) \\ s.t., \quad \mathbf{P}_{\cdot,i} &= \text{softmax}(W\mathbf{H}_{\cdot,i}^{L}) \\ \mathbf{H}^{L} &= \text{Transformer}(x_1, x_2, \ldots, x_T) \end{aligned} \tag{1}$$

where $\mathbf{P}_{\cdot,i}$ and $\mathbf{H}_{\cdot,i}^{L}$ denote the i-th column of the matrix $\mathbf{P}$ and $\mathbf{H}^{L}$. Here, $\mathbf{H}^{L} = [\mathbf{h}_1^L, \mathbf{h}_2^L, \ldots, \mathbf{h}_T^L]$ denotes the hidden states of the last layer. With $\mathbf{H}_{\cdot,i}^{L}$, a linear projection layer W is introduced to derive the predicted probability distribution $\mathbf{P}_{\cdot,i}$ over the vocabulary. The Transformer model consists of L Transformer blocks. Each block is composed of a multi-head self-attention (MHA) module and an FFN module, where the FFN module is generally a two-layer fully connected network. Formally,

$$\hat{\mathbf{h}}_1^l, \hat{\mathbf{h}}_2^l, \ldots, \hat{\mathbf{h}}_T^l = \text{MHA}(\mathbf{h}_1^{l-1}, \mathbf{h}_2^{l-1}, \ldots, \mathbf{h}_T^{l-1}) \tag{2}$$

$$\mathbf{h}_1^l, \mathbf{h}_2^l, \ldots, \mathbf{h}_T^l = \text{FFN}(\hat{\mathbf{h}}_1^l, \hat{\mathbf{h}}_2^l, \ldots, \hat{\mathbf{h}}_T^l) \tag{3}$$

where l represents the l-th Transformer block.

2.2 Reviewing the Mixture-of-Experts

MoE methods generally replace a single FFN module or multiple/all FFN modules as MoE modules. Each MoE module consists of multiple FFNs, whose outputs are mixed with the routine function $\mathbf{r}(\cdot)$, i.e., the router, as shown in the following formula:

$$\mathbf{h}_t^l = \sum_{i}^{N} \mathbf{r}_i(\hat{\mathbf{h}}_t^l) \cdot \text{FFN}_i(\hat{\mathbf{h}}_t^l) \qquad s.t., |\mathbf{r}(\hat{\mathbf{h}}_t^l)|_0 \ll N \tag{4}$$

where N represents the number of experts in a single MoE module, $\mathbf{r}_i$ denotes the routine result w.r.t the i-th expert, and $|\cdot|_0$ denotes the L_0-norm. The majority of elements $\mathbf{r}(\cdot)$ are zeros, and thus only a small portion of experts will be activated. Therefore, increasing the total number of experts in MoE does not increase computation or inference time substantially. Regarding the router $\mathbf{r}$ of MoE, it can be divided into fixed routing (Roller et al., 2021; Dai et al., 2022) and dynamic routing with learnable parameters (Fedus et al., 2022; Lepikhin et al., 2021; DeepSeek-AI et al., 2024; Dai et al., 2024). The former commonly use random hashing to determine (Roller et al., 2021), before training, which experts each token will be sent to, i.e., $\mathbf{r}$ is unlearnable and preset. In contrast, the latter employs a routing layer with learnable parameters to decide which expert should process an input token, i.e., $\mathbf{r}$ is to be learned during model training.

3 Method

3.1 Proposed MaskMoE

As mentioned before, dynamic routing can cause routing fluctuations, while fixed routing, like hash layers, can limit representation diversity. To mitigate the above two issues in MoE models, we propose MaskMoE to boost token-level learning via routing mask, as illustrated in Figure 1. Specifically, for each token, a fixed random masking is applied to the experts, allowing them to be routed only to their corresponding subsets of experts. The formula is represented as follows:

$$\mathbf{r} = \text{softmax}(W_r \hat{\mathbf{h}}_{\mathbf{t}}^l + \mathbf{m}^t) \tag{5}$$

where $\mathbf{m}^t$ is the masking vector used to control the visibility of token-specific experts for the token t. For visible experts, the corresponding elements in $\mathbf{m}^t$ are 0, while for invisible experts, the corresponding elements in $\mathbf{m}^t$ are $-\infty$. The masking vector is determined before training and does not change during the training process. For multi-layer MoE models, we reuse the same masking vector $\mathbf{m}^t$ across different MoE layers. Here, W_r is the learnable parameter of the routing function. It can be seen that we use the preset $\mathbf{m}^t$ to control the visibility of experts like the fixed routing method and meanwhile reserve the learnable parameter W_r as the dynamic routing method, and thus we combine the strengths of both.

To determine the frequency of tokens, we need to calculate the frequency of each token in the vocabulary within the Pile training dataset before pre-training. The high-frequency tokens that cover $P \times 100\%$ of the dataset are considered frequent ones, while the remaining tokens are considered infrequent ones. P is a tunable hyperparameter, ranging from 0 to 1. Then, the initial masking vector for a token t in vocabulary can be represented by the following formula:

$$\begin{aligned} \mathbf{m}^t &:= -\infty \cdot \mathbf{1}_N \\ \mathbf{C} &= \{\mathbf{C}_i\}_{i=1}^{V} \sim \mathcal{U}(\{1:N\}) \\ \mathbf{m}_j^t &= 0, \quad \forall j \in \mathbf{C} \end{aligned} \tag{6}$$

where the number of visible experts is V. $\mathcal{U}(\{1:N\})$ denotes a uniform distribution over the set

of integers from 1 to N, and C_i denotes the index of the selected expert for the i-th random sampling. It is worth noting that for infrequent tokens, each token usually has only one visible expert, i.e., $V = 1$. That promotes more thorough learning by the selected expert for such infrequent tokens. For frequent tokens, the number of visible experts is generally greater than 1, i.e., $1 < V <= N$, and it is common for V to be less than N. We believe that an appropriately sized V not only allows frequent tokens to have higher representation diversity compared to fixed routing but also enables more thorough training for an identical token compared to dynamic routing.

Additionally, it is worth noting that the masking vectors corresponding to each token in the vocabulary are generated based on frequency *before pre-training*, and all the generated mask vectors are concatenated into a mask embedding. The masking vector can be efficiently extracted from the mask embedding based on the token ID w.r.t token t during pre-training, which hardly introduces additional computational costs.

3.2 Load Balance Loss

LLMs are generally trained in a distributed manner. Yet distributed training can lead to load imbalance of MoE models (Lepikhin et al., 2021; Fedus et al., 2022; Dai et al., 2022), where a minority of expert handle the majority of tokens while the others remain idle most of the time. That can negatively impact training efficiency. It is generally desirable for the number of tokens processed by different experts to be roughly equal. To achieve that, a load balancing loss is commonly introduced in the training of MoE models. We follow previous works (Huang et al., 2024; Fedus et al., 2022) and adopt a widely used loss as follows:

$$\begin{aligned} \mathcal{L}_{bal} &= N \cdot \sum_{i=1}^{N} w_i R_i \\ \text{s.t.,} \quad w_i &= \frac{1}{B} \sum_{j=1}^{B} \mathbb{I}\{argmax(\mathbf{r}^j) = i\} \\ R_i &= \frac{1}{B} \sum_{j=1}^{B} \mathbf{r}_i^j \end{aligned} \tag{7}$$

where B represents the number of tokens in a mini-batch. $\mathbf{r}^j$ denotes the probability distribution of the routing output for the j-th token, derived via Eq. 5, and $\mathbf{r}_i^j$ represents the specific probability value w.r.t the i-th expert. It is noteworthy that for infrequent tokens, which have only one visible expert, the load balancing loss does not apply to them. Since infrequent tokens are assigned to experts through uniform random hashing, the overall load is supposed to be natively balanced, thus having no need for the balancing loss. The balancing loss primarily regulates the routing w.r.t frequent tokens.

Our final loss is a combination of the language model loss and the load-balance loss:

$$\mathcal{L} = \mathcal{L}_{lm} + \mathcal{L}_{bal} \tag{8}$$

4 Experiments

4.1 Pre-training Dataset

Following previous works, we use the Pile dataset (Gao et al., 2021) as pre-training data. The Pile dataset is a large-scale, publicly available corpus, containing 22 domains and over 825GB of English text data. For our experiments, we use the well-known LLaMA tokenizer for tokenization, with a vocabulary size of $32k$. We follow (Xie et al., 2023a; Su et al., 2024) to calculate the sampling rate for each domain based on the number of tokens after tokenization. Due to the limited computational budget, and following the pretraining settings of (Xie et al., 2023a; Su et al., 2024; Huang et al., 2024; Xiong et al., 2024; Lian et al., 2024), all models are pre-trained with 100B tokens.

Then, to identify infrequent tokens and frequent tokens, we calculate the frequency of each token in the training set and sort them in descending order of frequency. We categorize the top tokens that cover $P \times 100\%$ of the dataset as frequent ones, and the remaining as infrequent ones.

4.2 Compared Models

We compare the proposed MaskMoE with four remarkable baselines for validation experiments. Following previous works (Roller et al., 2021; Fedus et al., 2022; Dai et al., 2022), unless otherwise specified, our experiments select the top-1 expert for each MoE layer.

- **Dense** represents a standard Transformer language model.

- **SMoE** denotes a Switch Transformer (Fedus et al., 2022), where the router is a learnable layer. Unless otherwise specified, each MoE layer has 64 experts.

Model	Params	Activated Params	Pile PPL (↓)
Standard Transformer	468M	468M	6.95
Single-MoE-Layer Setup			
SMoE	1.3B	468M	6.62
Hash Layer	1.3B	468M	6.56
Share-MoE	1.3B	468M	6.72
MaskMoE	1.3B	468M	**6.48**
Multi-MoE-Layer Setup			
SMoE	10B	468M	6.18
Hash Layer	10B	468M	6.16
Share-MoE	10B	468M	6.15
MaskMoE	10B	468M	**6.11**

Table 1: Perplexity (PPL) results of language modeling.

- **Hash Layer** (Roller et al., 2021) signifies that through a random hash method, each token is assigned to a fixed expert before training. Unless otherwise specified, each MoE layer has 64 experts.

- **Share-MoE** is a hybrid dense and MoE model created using residual connections (Rajbhandari et al., 2022). Models with shared experts are currently a popular architecture (Dai et al., 2024; DeepSeek-AI et al., 2024; Zhao et al., 2024; Rajbhandari et al., 2022). Unless otherwise specified, Share-MoE has 1 shared expert and 128 routed experts, where each expert is 0.5× of the size of a standard FFN. During both training and inference, in addition to the activation of shared experts, the top-1 expert is also selected from the pool of 128 experts. In such a setup, Share-MoE has the same number of floating-point operations (FLOPs) and activation parameters as SMoE and Hash Layer. That allows for a fair comparison between Share-MoE, SMoE, and Hash Layer.

- **MaskMoE** aims to achieve a balance between representation diversity and thorough training, via the proposed routing masking method. Similarly to Share-MoE, MaskMoE also employs the remarkable shared expert architecture, which has been proven by (Dai et al., 2024; Rajbhandari et al., 2022) to be beneficial for the training of MoE models.

4.3 Experimental Setup

Following (Touvron et al., 2023a,b; Huang et al., 2024), we adopt the LLaMA architecture with 24 Transformer blocks and a hidden-state dimension of 1024. We employ the AdamW (Loshchilov and Hutter, 2019) optimizer with a cosine learning rate decay schedule. For the MoE model, following (Roller et al., 2021), we conduct experiments under both single-layer and multi-layer settings. For the single-layer setting, we replace the FFN layer with the MoE layer only for the last Transformer block. For the multi-layer MoE models, we follow Gshard (Lepikhin et al., 2021), and replace the FFN layer with MoE layer for every other Transformer block, resulting in a total of 12 MoE layers in this setting. For single-layer MoE models and dense models, we use a learning rate of $3e^{-4}$, while for Multi-layer MoE models, following (Lewis et al., 2021), we employ a learning rate of $1e^{-4}$ to ensure stable convergence of the models.

If not specified otherwise, the top most frequent tokens that cover 40% of the training set are considered frequent tokens, while the remaining are considered infrequent tokens, i.e., $P = 0.4$. Empirically, for frequent tokens, the visible expert count V is set to 8, and for infrequent tokens, the visible expert count V is set to 1.

4.4 Main Results

In this section, we first present the model's PPL on the Pile validation set. Then, following (Touvron et al., 2023a; Brown et al., 2020; Su et al., 2024; Dai et al., 2024), we conduct tests on various downstream benchmarks, including zero-shot tests for BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), LAMBADA (Paperno et al., 2016), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), StoryCloze (Mostafazadeh et al., 2016), and Arc-e (Bhakthavatsalam et al., 2021). Following (Touvron et al., 2023a; Su et al., 2024), we conduct 5-shot tests for TriviaQA (Joshi et al., 2017) and WebQs (Berant et al., 2013). Among them, TriviaQA and WebQs use `exact match` as the metric, while the remaining benchmarks are evaluated based on `accuracy`. For a fair comparison, we use the open-source evaluation tool `lm-evaluation-harness`[1] for assessment.

4.4.1 Perplexity Results

Table 1 shows the main results of language modeling on the Pile validation sets. With the same number of activated parameters during inference (i.e., 468M), MoE models consistently achieve improvements (i.e., substantially lower PPL) over the dense model. In comparison to SMoE and Hash Layer, under the `Single-MoE-Layer` setup, MaskMoE has respectively reduced the PPL on

[1] https://github.com/EleutherAI/lm-evaluation-harness

Model	BoolQ	Hellaswag	LAMBADA	PIQA	SIQA	StoryCloze	Arc-e	TriviaQA	WebQs
Standard Transformer	56.02	40.73	52.55	67.62	40.79	63.55	51.43	7.44	4.97
Single-MoE-Layer Setup									
SMoE	55.26	43.66	53.46	68.28	41.15	64.14	51.98	10.00	6.99
Hash Layer	56.61	43.88	54.67	**69.53**	41.76	64.19	**53.70**	9.82	6.89
Share-MoE	**58.87**	43.00	53.00	68.39	41.45	64.19	51.92	**10.54**	6.89
MaskMoE	58.38	**44.47**	**55.36**	68.99	**41.86**	**65.15**	53.14	10.39	**7.14**
Multi-MoE-Layer Setup									
SMoE	56.18	46.92	56.37	69.91	41.20	64.99	54.34	13.54	**7.33**
Hash Layer	57.16	46.61	56.68	**71.00**	40.83	65.47	55.05	12.97	5.95
Share-MoE	**58.71**	46.94	56.39	69.69	40.84	65.42	**55.89**	12.77	6.94
MaskMoE	58.32	**47.46**	**57.46**	70.62	**41.91**	**65.69**	55.35	**15.20**	6.74

Table 2: Performances of language models on downstream tasks. The best score is marked in **bold**, and the second best is underlined.

the Pile validation set by 0.14 and 0.08. In the `Multi-MoE-Layer` setup, their corresponding PPL are, respectively, reduced by 0.07 and 0.05. Compared to Share-MoE, MaskMoE also achieves a PPL decrease of 0.24 in `Single-MoE-Layer` setup and 0.04 in `Multi-MoE-Layer` setup. Such results demonstrate the effectiveness of our proposed MaskMoE.

It is noteworthy that Share-MoE achieves worse results than SMoE and Hash Layer in the `Single-MoE-Layer` setup, even with a built-in shared expert. However, MaskMoE consistently maintains superiority over them in both single-layer and multi-layer setups. That indicates MaskMoE exhibits stronger robustness across various settings.

4.4.2 Benchmark Results

As shown in Table 2, we present the performance of the models on downstream tasks. We can observe that MoE models consistently and significantly outperform the dense models in downstream tasks. More importantly, MaskMoE substantially outperforms the other MoE baselines, whether in `Single-MoE-Layer` setup or `Multi-MoE-Layer` setup.

Specifically, Compared to SMoE, MaskMoE consistently outperforms it in all 9 benchmarks with the `Single-MoE-Layer` setup and 8 of 9 benchmarks with the `Multi-MoE-Layer` setup. Against Hash Layer, MaskMoE excels in 7 of 9 benchmarks with the `Single-MoE-Layer` setup and in 8 of 9 benchmarks with the `Multi-MoE-Layer` setup. Compared to Share-MoE, MaskMoE demonstrates significant improvements in 7 of 9 benchmarks with the `Single-MoE-Layer` setup and in 6 of 9 benchmarks with the `Multi-MoE-Layer` setup.

The result above well demonstrates the performance enhancement achieved by using our proposed router masking method that controls the visibility of experts for different tokens. We believe the improvements are due to two factors. Firstly, compared to SMoE and Share-MoE, infrequent tokens are only routed to a single expert, ensuring more thorough training for those tokens. Secondly, in contrast to the Hash Layer, frequent tokens have multiple expert options, which promotes diversity in their representation learning.

5 Analyses

We conduct further experiments to provide more insightful analyses on the proposed MaskMoE. Considering the constraints of computational resources and following previous works (Roller et al., 2021; Xie et al., 2023b; Dai et al., 2022), unless otherwise specified, most of the analytical experiments are conducted under the setting of a single MoE layer.

5.1 Impact of Shared Expert

To explore the impact of shared experts, we conduct ablation experiments on them (i.e., removing the shared experts), and the results are shown in Table 3. We separately report the performance on the benchmark and the PPL on the Pile validation set. Note that after removing the shared experts, we adopt the same architecture as SMoE and Hash Layer, where the MoE layer consists of 64 fully-sized FFN.

We find that even without Shared-Experts, MaskMoE still outperforms SMoE, Hash Layer, and Share-MoE in terms of benchmark scores and PPL, with Table 2 as references. Specifically, compared to SMoE, MaskMoE w/o Shared-Experts excels in 8 of 9 benchmarks. Additionally, compared to Hash Layer and Share-MoE, MaskMoE w/o Shared-Experts gains significant improvements

in 7 of 9 benchmarks. Similarly, MaskMoE w/o Shared-Experts, yields lower PPL than SMoE, Hash Layer, and Share-MoE, with Table 1 as references. Such a result further validates the soundness of our proposed MaskMoE. Moreover, incorporating Shared-Expert structures continuously improves MaskMoE's performance. As shown in Table 3, compared to MaskMoE w/o Shared-Expert, MaskMoE excels in 5 of 9 benchmarks and has a lower PPL in the Pile validation set. We attribute it to that the shared expert structure promotes greater specialization in token representation (DeepSeek-AI et al., 2024).

	MaskMoE	w/o Shared-Expert
BoolQ	**58.38**	57.40
Hellaswag	**44.47**	44.10
LAMBADA	**55.36**	54.86
PIQA	68.99	**69.10**
SIQA	41.86	**41.97**
StoryCloze	65.15	**65.69**
Arc-e	53.14	**53.91**
TriviaQA	**10.39**	10.01
WebQs	**7.14**	6.89
PPL(↓)	**6.48**	6.51

Table 3: Impact of Shared Expert on benchmark performances and the Pile validation PPL. The best score is marked in **bold**.

5.2 Impact of Hyperparameters for Router Masking

Considering the proposed MaskMoE without shared experts still outperforms the compared MoE methods, to show the impact of router mask more clear and minimize the impact of the shared expert, here we keep the same experimental setting with the shared expert removed and conduct further experiments to analyze the impact of hyperparameters of our proposed MaskMoE.

The performance of MaskMoE is predominantly influenced by two hyperparameters: the boundary threshold (i.e., P) for the separation of frequent and infrequent tokens, and the maximum number of visible experts (i.e., V) for tokens. As shown in Table 4 and 5, we conduct parameter searches for both and report their PPL values on the Pile validation sets. Specifically, here V_a represents the number of experts visible for frequent tokens, and V_b represents the number of experts visible for infrequent tokens.

As shown in Table 4, we report the impact of the boundary threshold P. Here, $P = 0.0, V_a = 8, V_b = 1$ and $P = 1, V_a = 8, V_b = 1$ indicate no distinction between frequent and infrequent tokens, with every token having 1 and 8 visible experts, respectively. Their performances are significantly worse than those under the settings of $0 < P < 1$, given that $V_a = 8, V_b = 1$. That indicates tokens of different frequencies indeed require distinct visible experts, validating the effectiveness of our design. Additionally, we observe that regardless of the settings of P where $0 < P < 1$, the PPLs are consistently lower than $P = 0.0$ or $P = 1$. That validates the robustness of MaskMoE.

P	V_a	V_b	PPL	P	V_a	V_b	PPL
0.0	8	1	6.558 ♠	0.6	8	1	6.518
0.2	8	1	6.518	0.8	8	1	6.539
0.4	8	1	**6.506**	1.0	8	1	6.566 ♠

Table 4: Impact of the frequency threshold P for the separation of frequent and infrequent tokens. ♠ denotes no distinction between frequent and infrequent tokens.

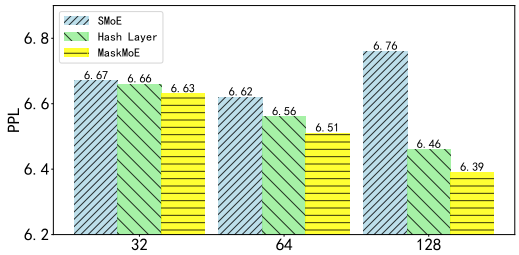
P	V_a	V_b	PPL	P	V_a	V_b	PPL
/	64	64	6.618♠	0.4	64	1	6.549
/	8	8	6.566	0.4	8	1	**6.506**
/	4	4	6.551	0.4	4	1	6.511
/	1	1	6.558 ♣	0.4	16	1	6.509

Table 5: Impact of visible experts. ♠ denotes SMoE, ♣ denotes Hash Layer.

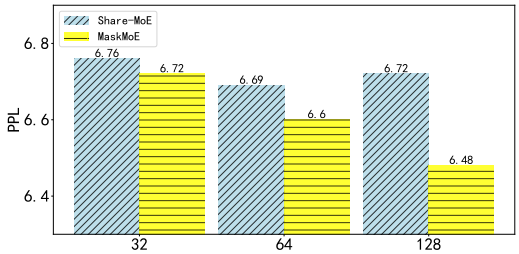
As shown in Table 5, we report the impact of the number of visible experts for frequent and infrequent tokens. Firstly, compared to SMoE with $V_a = V_b = 64$, concentrating tokens into fewer visible experts through our proposed routing mask, like $V_a = V_b = 8$ or $V_a = V_b = 4$, yields better outcomes. It indicates that routing tokens more densely to a reduced number of visible experts can enhance thorough training and may improve model performance. Moreover, we observe that $V_a = 8, V_b = 1$ outperforms $V_a = V_b = 8$. It well verifies that further reducing the number of visible experts for infrequent tokens is even more beneficial. Secondly, it is noteworthy that for MaskMoE, when $V_b = 1$, regardless of the value of V_a (whether it is 4, 8, 16, or 64), the performance of MaskMoE consistently outperforms that of SMoE($i.e., V_a = V_b = 64$) and Hash Layer($i.e., V_a = V_b = 1$). That also demonstrates the robustness of MaskMoE.

5.3 Impact of the Number of Experts

To investigate the variation in model performance under different configurations of expert numbers,



(a) None-Shared Expert Structures



(b) Shared-Expert Structures

Figure 2: Comparison of MoE-based Transformers with different numbers of experts.

we conducted experiments with 32, 64, and 128 experts. The experimental results are illustrated in Figure 2.

First of all, we observe that for SMoE and Share-MoE, increasing the number of experts does not necessarily lead to a lower PPL. In fact, when the number of experts reaches 128, both models even exhibit higher PPLs compared to having only 64 experts. We attribute it to that having more experts leads to the occurrences of an identical token being spread among more experts, and that disables each expert to learn adequately about this token, leading to model underfitting. In contrast, the performance of Hash Layer and MaskMoE continue to improve as the number of experts increases. Actually, when the number of experts increases, the average number of token occurrences assigned to each expert decrease. However, Hash Layer and MaskMoE can still ensure that each expert is fed with all or a large portion of the occurrences of a token, though the number of distinct tokens decreases for each expert. That makes the data for each expert more consistent and easier to learn.

Secondly, compared to Hash Layer, MaskMoE consistently performs better across all configurations. Although Hash Layer also enables experts to be adequately trained for specific tokens, MaskMoE excels in the diversity of representation learning for frequent tokens, which contributes to MaskMoE's overall advantage over Hash Layer.

Model	SMoE	Hash Layer	Share-MoE	MaskMoE
PPL(↓)	6.55	6.50	6.58	**6.46**

Table 6: The PPL on the Pile validation sets under the top-2 gating. The best score is marked in **bold**.

Finally, regardless of the number of experts, MaskMoE consistently yields the best performance. That further verifies its superiority and reasonableness.

5.4 Performance Analysis of Top-k Routing

We further analyze the performance of the MoE models under the top-k ($k = 2$) routing mechanism. In the SMoE model, the top-2 experts are selected from 64 fully-sized FFN experts based on the router's output scores. For the Hash Layer, two experts are randomly chosen from 64 fully-sized FFN experts using a random hash function. Regarding Share-MoE and MaskMoE models, in addition to the shared experts, the top-2 experts are selected from the 128 half-sized FFN experts based on the router's output scores. All settings share the same number of active parameters. Note that in MaskMoE, the number of visible experts is doubled from the original setting, i.e., $V_a = 16, V_b = 2$. As shown in Table 6, the model's PPLs on the Pile validation sets are presented. MaskMoE consistently yields the best performance, achieving a 0.12 decrease in PPL compared to Share-MoE. That further verifies the effectiveness.

6 Conclusions

In existing MoE language models, the widely-used dynamic routing methods such as SMoE, exhibit routing fluctuations, and route the occurrences of a token to multiple experts during training, potentially causing the model to encounter underfitting, especially for infrequent tokens. Although fixed routing like Hash Layer can mitigate the routing fluctuation issue, it struggles with capturing rich features for frequent tokens. Considering both routing fluctuation and representation diversity, we propose MaskMoE, a routing masking method that ensures adequate learning for infrequent tokens and rich representations for frequent tokens. Extensive experiments show that MaskMoE outperforms SMoE, Hash Layer, and Share-MoE on various downstream tasks and significantly reduces PPL on the Pile validation sets.

7 Limitiations

In this paper, we categorize tokens into frequent and infrequent tokens by their frequency. Such a classification method seems somewhat simple. A stronger classification method could potentially yield better results for our proposed MaskMoE. Due to computational constraints, we do not conduct further experiments. We leave the exploration of stronger partitioning methods for our future work.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. CoRR, abs/2312.11805.

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. IEEE Trans. Pattern Anal. Mach. Intell., 5(2):179–190.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1533–1544. ACL.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer AI2 reasoning challenge. CoRR, abs/2102.03315.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of Proceedings of Machine Learning Research, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 7432–7439. AAAI Press.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. CoRR, abs/1904.10509.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. CoRR, abs/2204.02311.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina

Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 2924–2936. Association for Computational Linguistics.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. CoRR, abs/2401.06066.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Stablemoe: Stable routing strategy for mixture of experts. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 7085–7095. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, Tao Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, and Xiaowen Sun. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. CoRR, abs/2405.04434.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 13042–13054.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022a. Glam: Efficient scaling of language models with mixture-of-experts. In International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pages 5547–5569. PMLR.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022b. GLM: general language model pretraining with autoregressive blank infilling. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 320–335. Association for Computational Linguistics.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. J. Mach. Learn. Res., 23:120:1–120:39.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. CoRR, abs/2101.00027.

Alex Graves. 2013. Generating sequences with recurrent neural networks. CoRR, abs/1308.0850.

Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. 2023. Merging experts into one: Improving computational efficiency of mixture of experts. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 14685–14691. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes

Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. CoRR, abs/2203.15556.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. 2024. Harder tasks need more experts: Dynamic routing in moe models. CoRR, abs/2403.07652.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. Neural Comput., 3(1):79–87.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pages 1601–1611. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. CoRR, abs/2001.08361.

Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans. Acoust. Speech Signal Process., 35(3):400–401.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In 1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP '95, Detroit, Michigan, USA, May 08-12, 1995, pages 181–184. IEEE Computer Society.

Mark Koranda, Martin Zettersten, and Maryellen C. MacDonald. 2018. Word frequency can affect what you choose to say. In Proceedings of the 40th Annual Meeting of the Cognitive Science Society, CogSci 2018, Madison, WI, USA, July 25-28, 2018. cognitivesciencesociety.org.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. BASE layers: Simplifying training of large, sparse models. In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pages 6265–6274. PMLR.

Haoran Lian, Yizhe Xiong, Jianwei Niu, Shasha Mo, Zhenpeng Su, Zijia Lin, Peng Liu, Hui Chen, and Guiguang Ding. 2024. Scaffold-bpe: Enhancing byte pair encoding with simple and effective scaffold token removal. arXiv preprint arXiv:2404.17808.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Tomás Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010, pages 1045–1048. ISCA.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. CoRR, abs/1604.01696.

OpenAI. 2023. GPT-4 technical report. CoRR, abs/2303.08774.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia

Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew J. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. CoRR, abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21:140:1–140:67.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation AI scale. In International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pages 18332–18346. PMLR.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. 2021. Hash layers for large sparse models. In Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 17555–17566.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. CoRR, abs/1904.09728.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net.

Zhenpeng Su, Zijia Lin, Baixue Baixue, Hui Chen, Songlin Hu, Wei Zhou, Guiguang Ding, and Xing W. 2024. MiLe loss: a new loss for mitigating the bias of learning difficulties in generative language models. In Findings of the Association for Computational Linguistics: NAACL 2024, pages 250–262, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. CoRR, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. CoRR, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

Yuan Xie, Shaohan Huang, Tianyu Chen, and Furu Wei. 2023b. Moec: Mixture of expert clusters. In Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pages 13807–13815. AAAI Press.

Yizhe Xiong, Xiansheng Chen, Xin Ye, Hui Chen, Zijia Lin, Haoran Lian, Jianwei Niu, and Guiguang Ding. 2024. Temporal scaling law for large language models. arXiv preprint arXiv:2404.17785.

Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. 2019. Condconv: Conditionally parameterized convolutions for efficient inference. In Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 1305–1316.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 4791–4800. Association for Computational Linguistics.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068.

Hao Zhao, Zihan Qiu, Huijia Wu, Zili Wang, Zhaofeng He, and Jie Fu. 2024. Hypermoe: Towards better mixture of experts via transferring among experts. arXiv preprint arXiv:2402.12656.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. arXiv preprint arXiv:2202.08906.

A Related Work

A.1 Language Models

Language models are statistical models designed to optimize the likelihood of token sequences in training data (Touvron et al., 2023a). Initially, language models relied on n-gram statistics (Bahl et al., 1983; Katz, 1987; Kneser and Ney, 1995). Subsequently, the emphasis shifted to neural network-based models, particularly Recurrent Neural Networks (Mikolov et al., 2010) and their variants like LSTMs (Graves, 2013). Those models have demonstrated the ability to learn intricate patterns within textual data, achieving significant success in various language modeling applications. In recent times, Transformers have become the predominant architecture for language models. Notable examples include BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), GPT-2 (Radford et al., 2019), UniLM (Dong et al., 2019), and T5 (Raffel et al., 2020). The introduction of GPT-3 (Brown et al., 2020), which boasts 175 billion parameters, marked a significant milestone due to its exceptional performance across numerous downstream tasks. And that then led to a surge in research focusing on large generative language models, including prominent works such as Gopher (Rae et al., 2021), PaLM (Chowdhery et al., 2022), Pythia (Biderman et al., 2023), OPT (Zhang et al., 2022), GLM (Du et al., 2022b; Zeng et al., 2023) and LLaMA (Touvron et al., 2023a,b). Currently, GPT4 (OpenAI, 2023) achieves truly remarkable results.

However, as the size of the model grows, the computational demands for both training and inference also increase. MoE models achieve scalability by sparsely activating a portion of the model's parameters, allowing for an increase in model size without significantly raising computational overhead. Consequently, MoE models have been receiving increasing attention recently.

A.2 Mixture-of-Experts

The concept of Mixture-of-Experts (MoE) models was initially proposed by (Jacobs et al., 1991). Later, (Shazeer et al., 2017) applied MoE to LSTM, training an LSTM model with up to 137B parameters. With the rise of the Transformer architecture (Vaswani et al., 2017; Devlin et al., 2019), Gshard (Lepikhin et al., 2021) further applied MoE to Transformers. Subsequently, powerful MoE models such as GLaM (Du et al., 2022a) and Switch Transformer (Fedus et al., 2022) emerged.

Early works (Zoph et al., 2022; Fedus et al., 2022; Du et al., 2022a; Lepikhin et al., 2021) mainly focused on dynamic routing methods for learning-to-route MoE. Then the Hash Layer (Roller et al., 2021) is proposed to use a random hashing method to route tokens to a fixed expert, which even achieved better results than dynamic routing. As an explanation to the advantage of the simple fixed routing method, Dai et al. 2022 pointed out that dynamic routing has the problem of routing fluctuation problem, meaning that an identical input is assigned to different experts as training progresses. Routing fluctuation often harms sample efficiency, especially for the learning of infrequent tokens in the language model. On the contrary, other previous works (He et al., 2023; Yang et al., 2019) also argued that having more experts allows tokens to obtain richer representations, and fixed routing can affect the diversity of representations, especially for frequent tokens.

Considering both routing fluctuation and representation diversity, we propose MaskMoE, which uses routing masks to alleviate the underfitting problem of infrequent tokens caused by routing fluctuations, while maintaining the representational diversity of frequent tokens.