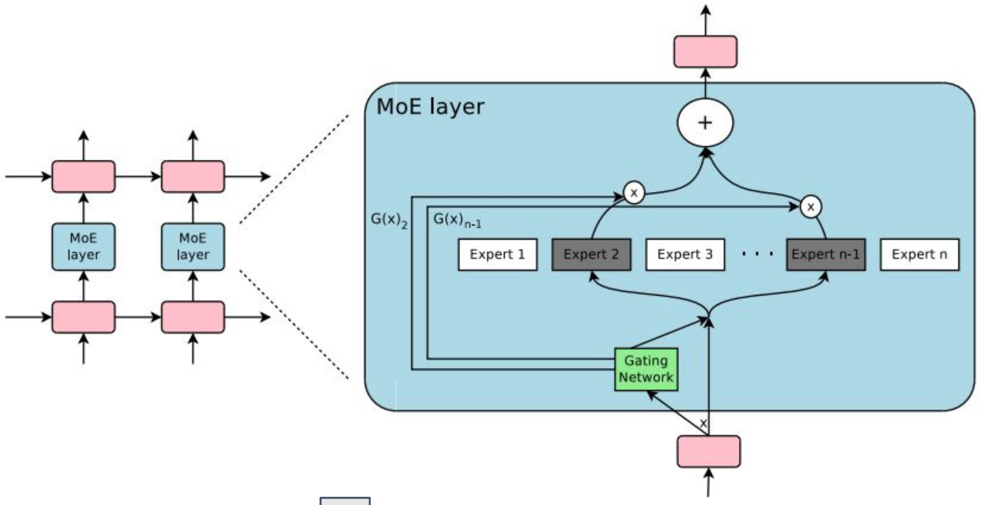
Mixture of Diverse Size Experts

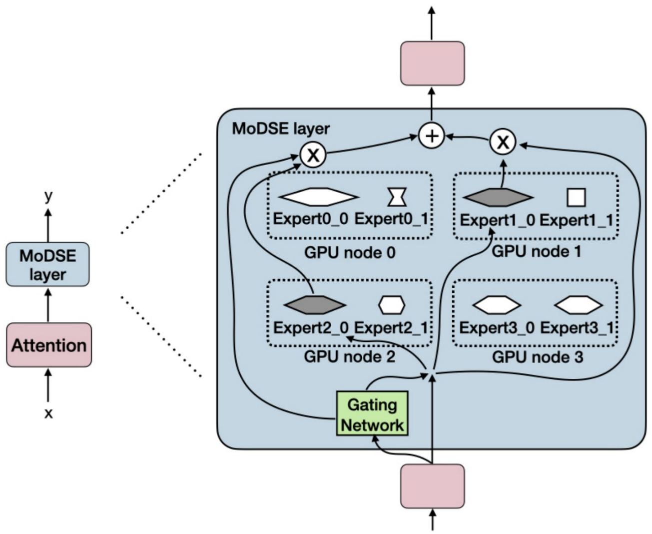
Manxi Sun, Wei Liu, Jian Luan, Pengzhi Gao, and Bin Wang
Xiaomi AI Lab, Beijing, China
{sunmanxi, liuwei40, luanjian, gaopengzhi, wangbin11}@xiaomi.com

01 Motivation

- Almost all existing MoE architectures consist of **experts with identical structures and sizes**. This homogeneous architecture becomes a significant bottleneck when generating tokens with varying difficulty; **some tokens are easier to predict, while others are more challenging**.



- To deal with the varied difficulty, we propose the Diverse Size Experts structure for each FFN layer, where each **expert has a different parameter size to handle generating tasks of varying difficulty**.



02 Model

We denote the designed Diverse Size Experts as $\{\hat{E}_1(\cdot), \cdots, \hat{E}_N(\cdot)\}$, and the dimension of the hidden layer for $\hat{E}_i(\cdot)$ is $\hat{h}_i$.

$$\hat{y} = \sum_{i=1}^{N} \hat{G}_i(x)\hat{E}_i(x) \tag{1}$$

$$(i_1^1, i_1^2), \cdots, (i_n^1, i_n^2), \text{ with } n = \frac{N}{2} \tag{2}$$

$$\hat{h}_{i_k^1} + \hat{h}_{i_k^2} = 2 \times h, \text{, with } k \in 1 \cdots n \tag{3}$$

To maintain the overall parameter size, the experts are grouped into pairs (i_k^1, i_k^2), where $k \in 1 \cdots n$ indicates the pair of the experts. The average value of $\hat{h}_i$ within each pair equals the conventional expert hidden dimension h, with one expert being larger than the average size and the other smaller.

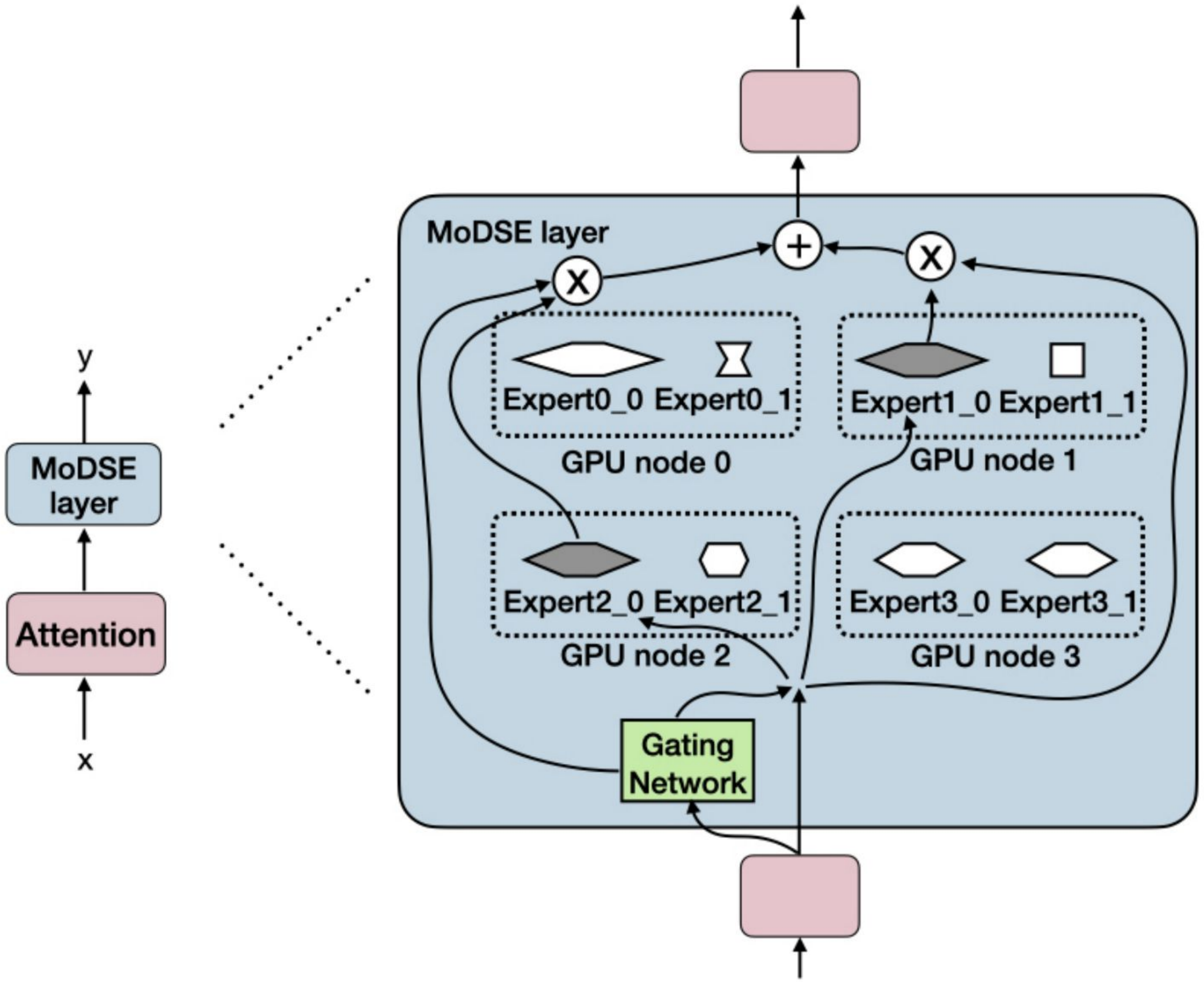


Figure 1: Overview of a MoDSE layer with different sizes of experts.

03 Experimental Setup

The MoE structure is based on the Llama 2 model with the dense FFNs layers replaced by expert layers. Table 1 summarises the model architecture parameters. For the MoDSE setting, we adjust the expert sizes in baseline by modifying the dimensions of the hidden layers in $300M \times 8$ and $700M \times 8$ settings, as listed in Table 2. There are 8 experts grouped into 4 pairs, with the ratio to the input size as $(4.5, 0.5)$, $(4.0, 1.0)$, $(3.0, 2.0)$, and $(2.5, 2.5)$. We train byte pair encoding (BPE) tokenizer with both English and Chinese datasets, and use it in the following experiments.

Parameter	$300M \times 8$	$700M \times 8$
dim	1536	2048
n_layers	8	12
# heads	12	32
# expert	8	8
top k	2	2
vocal_size	30064	30064
h	3840	5120

Table 1: MoE model architecture with $300M \times 8$ and $700M \times 8$ parameters, both with identical expert sizes.

Model	**Expert size pairs**
$300M \times 8$	[(6912,768), (6144,1536), (4608,3072), (3840,3840)]
$700M \times 8$	[(9216,1024), (8192,2048), (6144,4096), (5120,5120)]

Table 2: The list of expert pair sizes in $300M \times 8$ and $700M \times 8$ parameters.

03 Experimental Setup

Datasets We collected 100B tokens training data from various reputable sources for pre-training. This dataset includes both English and Chinese language, and spans multiple fields, including CommonCrawl, code, academic papers, books, mathematics, and Q&A.

Training configurations We utilize the Adam optimizer, with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.95$, eps $= 1e{-}8$, weight decay $= 0.1$ and gradient clipping $= 1.0$. We use a cosine learning rate schedule, such that the initial learning rate is 2e-7, the warm-up update steps are 2000 and the minimal learning rate is 3e-5. We employ the ZeRO optimization for distributed training. All experiments are carried out on clusters equipped with NVIDIA A800 GPUs. The A800 cluster features 8 GPUs per node, interconnected using NVLink and NVSwitch within nodes. Two nodes are used for the $300M \times 8$ setting, and 8 nodes are used for the $700M \times 8$ setting.

04 Main Results

Benchmark	Baseline	MoDSE
AGIEval (Acc.)	26.2	**28.1**
MMLU (Acc.)	26.5	**29.9**
INTENT (Acc.)	13.6	**16.5**
GSM8K (EM)	5.9	**7.7**
LAMBADA (EM)	36.8	**38.9**
MATH (EM)	0.8	**2.6**
TriviaQA (EM)	5.2	**8.3**
PIQA (EM)	53.1	**57.6**
SIQA (EM)	42.9	**60.9**

Table 3: Comparison between MoE baseline and MoDSE on size of $700M \times 8$. The bold font indicates the better. With the same parameter, MoDSE achieves better performance than the baseline. All the tasks are fewshot in context learning, and GSM8k includes 8 shots examples and others include 5 shots examples.

Benchmark	MoE	MoDSE
AGIEval	48s	59s
MMLU	3min 26s	3min 27s
INTENT	1min 31s	1min 34s
GSM8K	20min 26s	20min 43s
LAMBADA	40min44s	40min48s
MATH	21min 21s	21min 34s
TriviaQA	46min 53s	48min 55s
PIQA	44min56s	43min34s
SIQA	2min35s	2min36s

Table 4: The inference duration of the baseline and MoDSE models on downstream tasks. The AGIEval task contains 615 examples, the MMLU task contains 2341 examples, the INTENT task contains 741 examples and the rest tasks with 100 examples.

04 Main Results

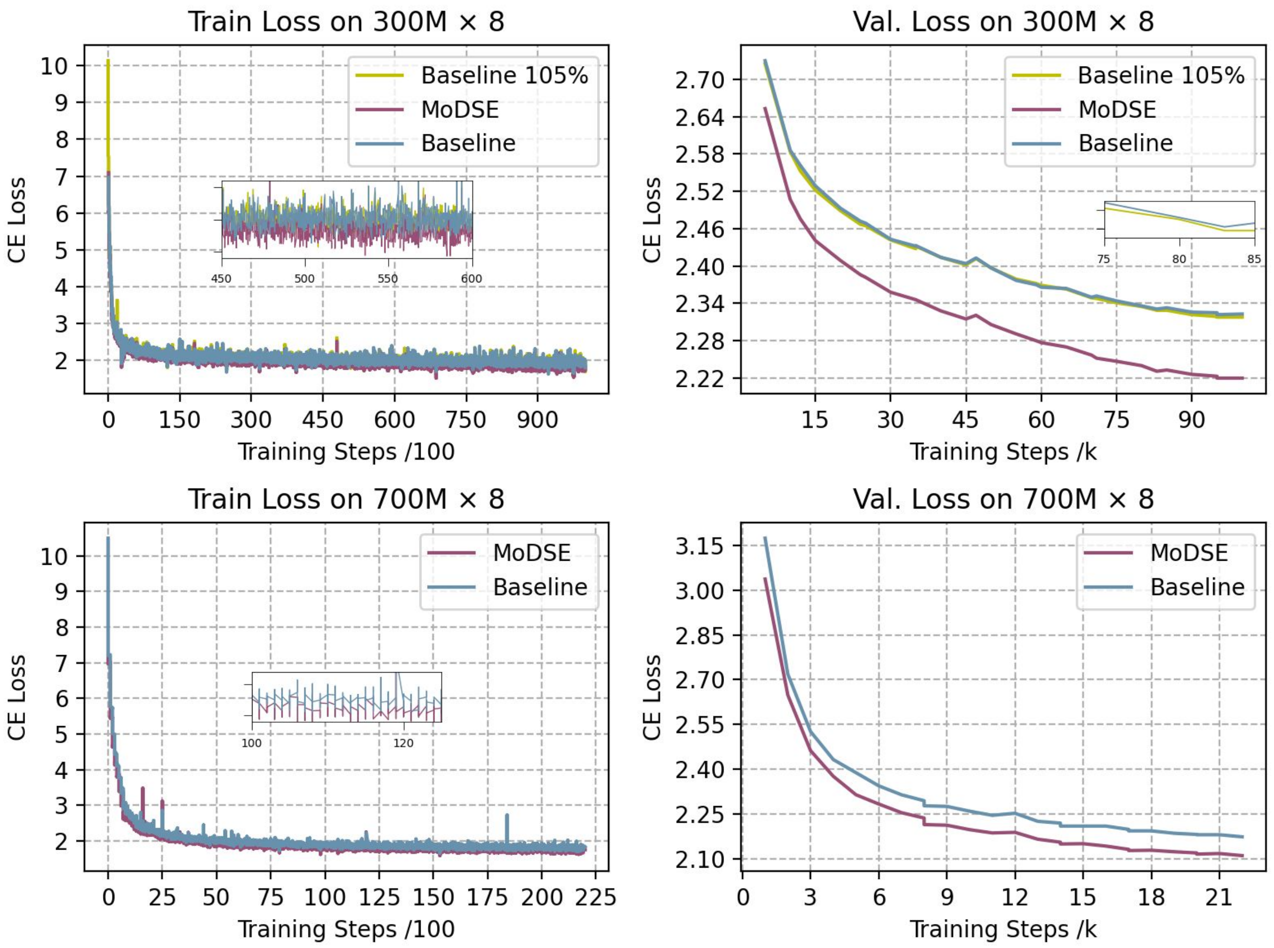


Figure 2: Training and validation loss curves for the $300M \times 8$ and $700M \times 8$ models, with cross-entropy loss values indicated on the curves.

05 Analysis on Token Routing

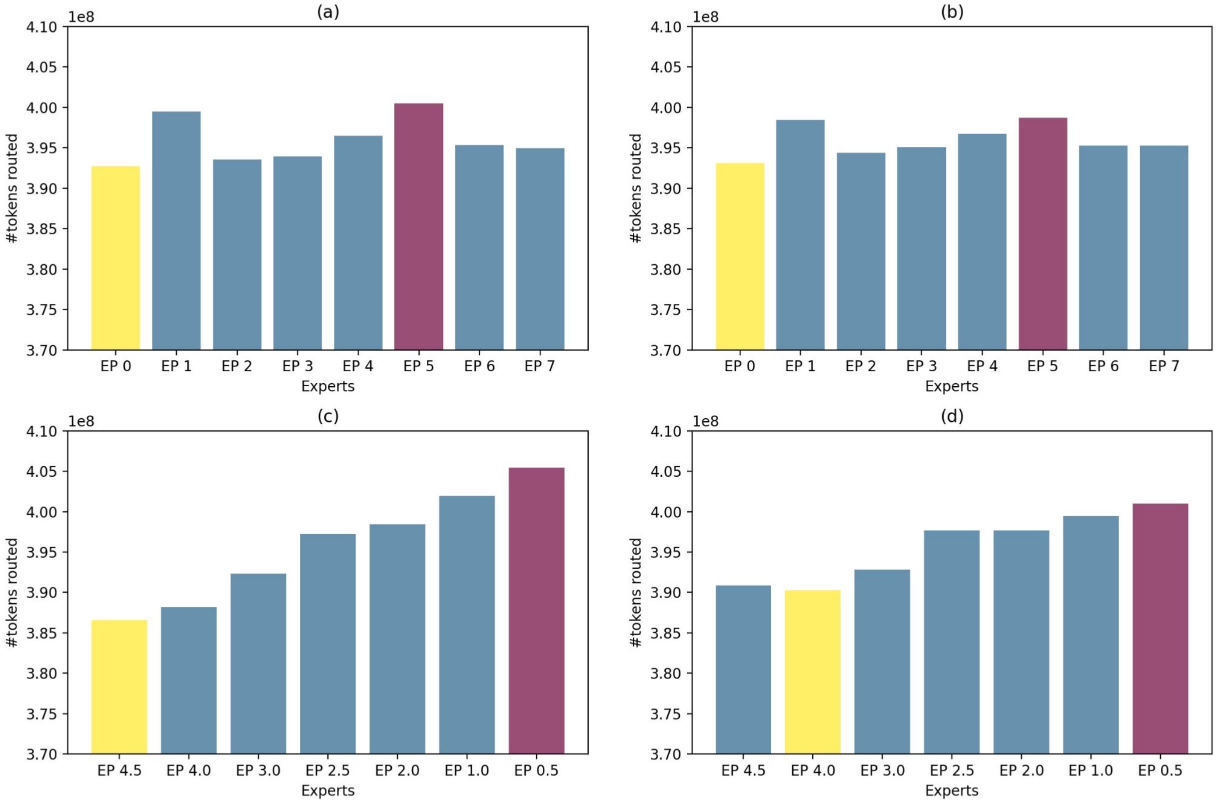


Figure 3: The number of tokens routed to each expert. The bar is the sum of the number across the layers. Figure (a) shows results in Baseline in epoch 2, and (b) in the last epoch. Figure (c) shows results in MoDSE in epoch 2, and (d) in the last epoch. The purple bar indicates the most routed expert, and the yellow indicates the least.

loss threshold	avg. loss red.	#tokens
2.0	0.58	180
1.8	0.46	222
1.6	0.36	337
1.4	0.32	730
1.2	0.22	1991
1.05	0.18	3633

Table 5: Average CE loss reduction across different intervals. The higher the initial CE loss, the more significant the improvement demonstrated by the MoDSE model. The avg. loss red. stands for the average CE loss decrease from baseline to MoDSE.

06 Difficult Tokens Routing Distribution

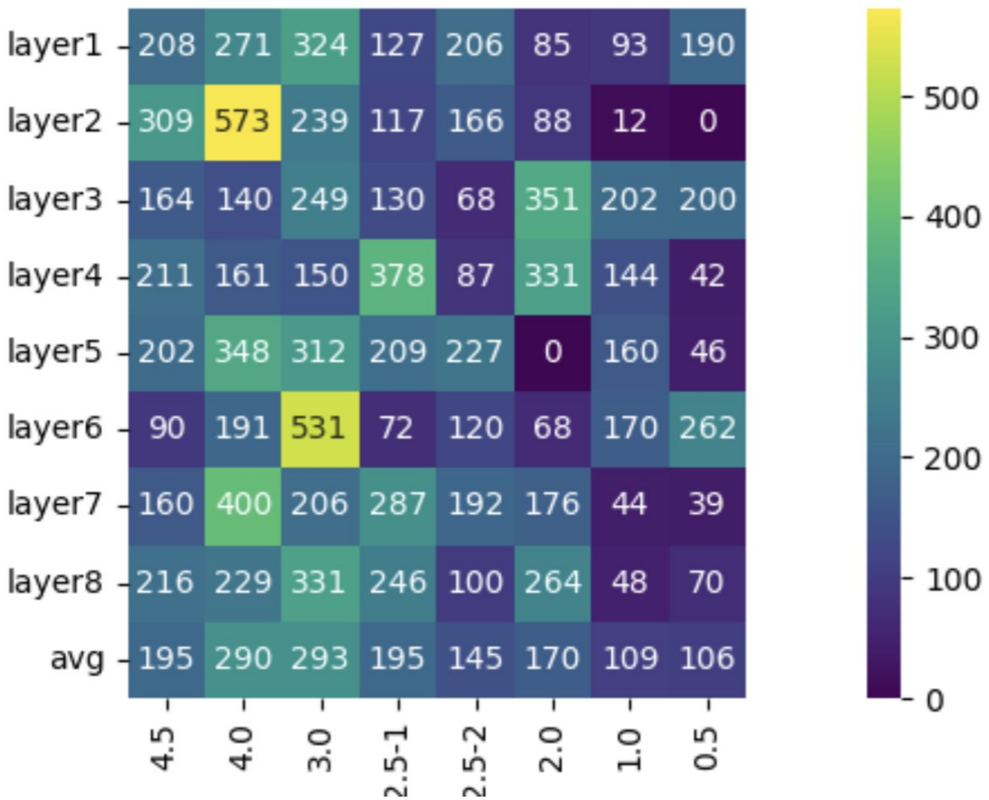


Figure 4: The top one expert choice of difficult tokens across eight layers. More tokens are routed to larger experts, distributed on the left half of the heat map.

Expert Size	**#tokens to top1 & 2**	**#tokens to top1**
4.5	2649	1560
4.0	3729	2313
3.0	4095	2342
2.5	2332	1166
2.5	2933	1566
2.0	2877	1363
1.0	2972	873
0.5	2477	849
sum(L)	10473	6215
sum(S)	8326	3085

Table 6: The distribution of difficult tokens across different experts. The sum(L) stands for the total token number routed to larger experts (4.5, 4.0, 3.0), and the sum(S) stands for the total token number routed to smaller experts (2.0, 1.0, 0.5).