Expert-Token Resonance MoE: Bidirectional Routing with Efficiency Affinity-Driven Active Selection

Anonymous ACL submission

Abstract

Mixture-of-Experts (MoE) architectures have emerged as a paradigm-shifting approach for large language models (LLMs), offering unprecedented computational efficiency. However, these architectures grapple with challenges of token distribution imbalance and expert homogenization, impeding optimal semantic generalization. We propose a novel expert routing framework that incorporates: (1) An efficient routing mechanism with lightweight computation. (2) An adaptive bidirectional selection mechanism leveraging resonance between experts and tokens. (3) A module that determines the lower bounds of expert capacity based on dynamic token distribution analysis, specifically designed to address drop-and-pad strategies. It is also integrated with orthogonal feature extraction module and an optimized loss function for expert localization. This framework effectively reduces expert homogeneity while enhancing the performance of the expert selection module. Additionally, we introduce a local expert strategy that simultaneously improves load balancing and reduces network communication overhead. It achieves a 40% reduction in token processed by each expert without compromising model convergence or efficacy. When coupled with communication optimizations, the training efficiency improvements of 5.4% to 46.6% can be observed. After supervised fine-tuning, it exhibits performance gains of 9.7% to 14.1% across GDAD, GPQA, and TeleQnA benchmarks.

1 Introduction

Large language models (LLMs) have shown exceptional proficiency in understanding deep structures and complex semantic relationships within language (Zhao et al., 2023). As these models scale up, their capabilities in language generation and logical comprehension are enhanced, but this comes at the cost of significant computational, communication, and storage demands (Jiang et al., 2024b). To scale models efficiently without disproportionately increasing computational costs, researchers have incorporated the Mixture-of-Experts (MoE) architecture into LLMs (Lepikhin et al., 2020). The MoE framework integrates multiple experts within the model, each tasked with processing specific types of inputs (Fedus et al., 2022). For a given input, only a subset of experts is activated, allowing for more efficient use of computational resources (Du et al., 2022). Recently, several LLMs employing MoE structures, such as DeepSeek-V3 (Liu et al., 2024a) and Mixtral (Jiang et al., 2024a), have demonstrated outstanding performance on various leaderboards.

Despite the efficiency benefits of MoE in scaling model sizes, it introduces several new challenges and drawbacks (Shazeer et al., 2017). The conventional MoE model's convergence and the experts' generalization capabilities are heavily dependent on the design of the routing strategy, which easily leads to an imbalanced "winner-takes-all" phenomenon among experts. The imbalance between excessively "developed" experts and those lacking adequate training may compromise or even nullify the intended functionality of routing strategies. Recent studies address these challenges from multiple perspectives (Li et al., 2023). StableMoE (Dai et al., 2022) proposes a two-stage training approach to address the issue of routing fluctuation. This method involves training the routing network independently from the backbone model and utilizing a frozen, distilled routing mechanism to allocate tokens. Dynamic-MoE (Huang et al., 2024a) designs a dynamic routing Mixture-of-Experts (MoE) policy that evaluates the sufficiency of current experts while reducing activated parameters by 90%. The characteristics of classical gated routing lead to experts being unable to learn features mastered by other experts. To address this, MoDE (Xie et al., 2024) proposes moderate distillation between experts to mitigate the general-

ization problems caused by narrow learning paths. DYNMoE (Guo et al., 2024) introduces a unique gated routing mechanism capable of adaptively determining the number of activated experts through trainable expert thresholds, even allowing for the addition or removal of experts.

In addition to the classical token choice scenario, previous researches also propose work utilizing expert choice (EC). Google Brain introduces the EC routing algorithm (Zhou et al., 2022), which assigns experts with predetermined buffer capacities to the Top-k tokens to ensure load balance. The Brainformer (Zhou et al., 2023) also adopts this routing strategy, constructing a trainable gating matrix to project the input feature space onto scores corresponding to each expert. Then, each token is routed to the Top-k experts. This strategy is proven highly effective in achieving expert load balancing and enhancing expert learning outcomes. Autonomy-of-Experts models (Lv et al., 2025) design a novel MoE paradigm in which experts autonomously select themselves to process inputs by aware of its own capacity to effectively process a token.

The design of routing strategy is crucial to the MoE structure, while not all tokens may be suitable for training (Riquelme et al., 2021). In addition to data preprocessing techniques such as dataset cleaning and deduplication, previous studies have also considered how to discard certain tokens within the model. Early work introduced the concept of expert capacity (Lepikhin et al.), which refers to the maximum number of tokens each expert can process at once. Tokens exceeding this capacity are discarded. Expert capacity helps to ensure load balance among experts while facilitating All-to-All communication implementation. However, in situations where it is uncertain whether a token contributes to training, there is a risk of discarding class-discriminative samples, potentially compromising the model's training outcomes. DeepSeek-V2 (Liu et al., 2024a) designs a device-limited routing mechanism to bound MoE-related communication cost. DeepSeek-V3 (Liu et al., 2024a) pioneers an auxiliary-loss-free strategy to minimizes the performance degradation. This approach minimizes the constraints on expert specialization imposed by knowledge hybridity and knowledge redundancy. XMoE (Yang et al., 2024) achieves more precise router by implementing a threshold-based approach. If a token reaches the specified threshold, it is processed exclusively by a single expert while being discarded by other experts within the Top-k selection. This method allows for more nuanced token selection and processing. LocMoE (Li et al., 2024) leverages orthogonal routing weights to prevent token homogenization across different expert networks and introduces the Grouped Average Pooling (GrAP) layer (Wang et al., 2023) for token feature extraction. Under these conditions, LocMoE also provides the theoretical proof for the lower bound of expert capacity.

In this paper, we propose expert-token resonance, a mechanism consisting of an expert-token bidirectional selection router and the adaptive expert capacity strategy. The primary contributions of this paper are as follows:

1. **Affinity-based Efficient Expert Routing via GrAP.** By leveraging cosine similarity between tokens and gating weights to define affinity scores, our router effectively guides experts to focus on distinct token segments, mitigating the expert homogenization problem. Meanwhile, the GrAP design reduces computational complexity by a factor of 1/2D to 1/D compared to traditional MLPs (D denotes the dimension of the intermediate hidden layer). This integrated approach demonstrates both improved routing effectiveness and substantial computational efficiency.

2. **Expert-token Bidirectional Selection.** By integrating the concepts of expert choice router (ECR) and token choice router (TCR), we propose the adaptive bidirectional selection mechanism. Contrast to conventional router, the bidirectional selection router allows MoE to enhance the training success rate while considering expert capacity constraints. Its effectiveness has been theoretically validated.

3. **The Adaptive Expert Capacity Bound.** Setting an adaptive affinity threshold allows the lower bound of expert capacity to be significantly reduced. As training iterations increase, the information density of token features grows, causing the expert capacity to initially decrease and then stabilize. Ultimately, the training efficiency of MoE can be greatly enhanced.

Expert-token resonance mechanism adopts the state-of-the-art MoE model Mixtral 8×7B as the backbone, and utilizes MindSpeed-LLM, MindSpeed, and Megatron-LM (Shoeybi et al., 2019)

libraries for training on Ascend NPU clusters. Ascend designs a new computing architecture for LLM training and inference scenarios (Liao et al., 2021), boasting powerful low-bit computing capabilities. Experiments conducted on clusters with 32, 64, and 256 NPUs indicate that our approach improves training efficiency by 5.4% to 46.6% compared to the baseline, and by 2.9% to 13.3% compared to LocMoE. Model performance is enhanced by 9.7% to 14.1% compared to the baseline, and by 1.7% to 4.1% compared to LocMoE.

The rest of this paper is structured as follows: Section **Method** presents the methods proposed in this paper, along with theoretical evidence. Section **Experiments** analyzes the experimental results of our approach regarding training efficiency and model performance. The final section summarizes the content of this paper and offers an outlook on future improvements.

2 Method

In this section, we present the efficient routing mechanism, and our adaptive bidirectional selection mechanism is detailed. Then, for traditional drop-and-pad strategies, a dynamic token distribution analysis module that optimizes the lower bounds of expert capacity are displayed. Moreover, we also describe the loss for expert load balancing.

2.1 Model Architecture.

Backbone. The MoE architecture, based on the Transformer framework, efficiently scales up model size with low computational overhead, benefiting from two primary structures: a sparse gating network for routing tokens and expert networks for processing specific token categories.

We consider the supervised classification for brevity where the training samples are $\{(\boldsymbol{x}^{(i)}, y_i)\}_{i=1}^{N} \sim \mathcal{D}$. Each training sample $\boldsymbol{x}^\top = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_s^\top) \in \mathcal{R}^{sd}$ has s tokens with token feature $\boldsymbol{x}_i \in \mathcal{R}^d, \forall i \in [s]$, and label $y \in \mathcal{N}^+$. The objective is to learn the map of $\boldsymbol{x}$ to the corresponding y. The general MoE structure are formulated as

$$\mathrm{MoE}(\boldsymbol{x}) = \sum_{t=1}^{s} \sum_{i=1}^{n} G_i(\boldsymbol{x}_t) \cdot E_i(\boldsymbol{x}_t), \tag{1}$$

where n is the number of experts, $G(\boldsymbol{x}_t)\colon \mathcal{R}^d \to \mathcal{R}^n$ is the gating weight vector of experts which maps the tokens of $\boldsymbol{x}_t$ into the coresponding experts with weights, e.g., $G_i(\boldsymbol{x}) = \mathrm{Softmax}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{\epsilon})$ where the softmax is applied to each row, and $E_i(\boldsymbol{x}_t)\colon \mathcal{R}^d \to \mathcal{R}$ is the i-th expert network, see (Liu et al., 2024b) for current different router methods. Generally, $n \ll s$, which saves much computation compared to the dense structure.

Cost-Efficient Sparse Expert-Token Affinity. $\boldsymbol{W}_{\text{aff}}$ denotes the **expert-token affinity matrix**. After processing through the GrAP routing layer, tokens generate a diagonal sparse matrix as shown. Compared to the dense matrix produced by traditional routing layers, this reduces the parameter count to 1/D of the original, significantly decreasing the computational overhead of the expert routing layer.

With GrAP as the layer of feature extraction, the formulation of $\boldsymbol{W}_{\text{aff}}$ is as followed:

$$\boldsymbol{W}_{\text{aff}} = \begin{pmatrix} \boldsymbol{w}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{w}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{w}_n \end{pmatrix} \tag{2}$$

$$\boldsymbol{w}_i = \frac{n}{d} \cdot \mathbf{1}\left\{ \frac{i \cdot d}{n} \le j < (i+1)\frac{d}{n} \right\} \quad 0 \le j < d \tag{3}$$

The expert-token affinity matrix is employed as the gating weight to calculate the affinity score between each expert and token. We define the affinity score of t-th token and i-th expert as the cosine similarity between vectors $\boldsymbol{x}_t$ and $\boldsymbol{w}_i$:

$$\delta_{ti} = \cos(\boldsymbol{x}_t, \boldsymbol{w}_i) := \boldsymbol{x}_t^\top \boldsymbol{w}_i / (\|\boldsymbol{x}_t\| \cdot \|\boldsymbol{w}_i\|) \tag{4}$$

The affinity score intuitively reflect how closely the two inputs are associated. From a perspective of semantic, the affinity scores derived from affinity metrics consisting of orthogonal vectors represent the degree of association between each token and various experts, as shown in Figure 1. Therefore, we leverage the affinity score as the principle of our affinity-driven active selection routing mechanism.

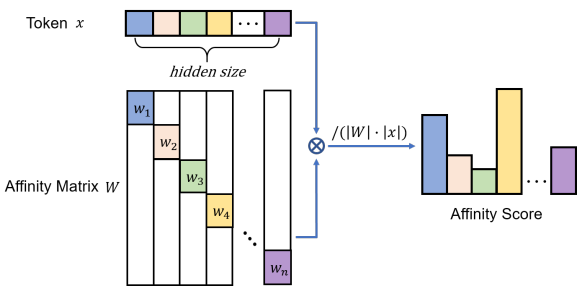


Figure 1: The illustration of affinity score.

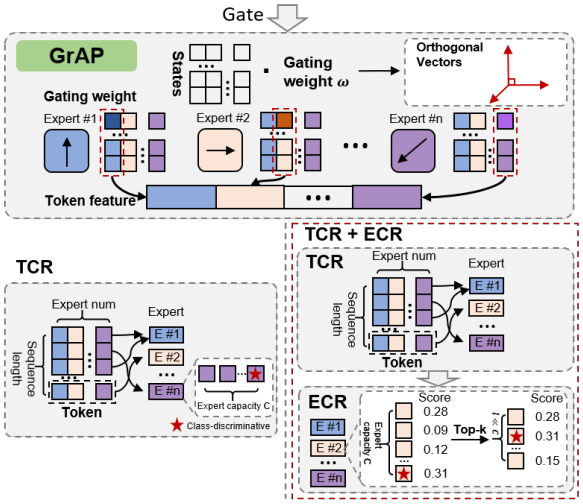


Figure 2: The architecture of the gate network along with the hybrid TCR + ECR router.

Routing Strategy. We consider our affinity-driven active selection routing as a hybrid of TCR (Clark et al., 2022; Zhou et al., 2022) and ECR. As the name suggested, TCR lets each token choose its *top-scored* experts, and ECR lets each expert choose its *top-scored* tokens. Specifically, we use the result of the expert-token affinity metrics as the affinity score between tokens and experts. In conventional TCR routing strategy, the tokens are simply route to their Top-1 expert. In our hybird **TCR+ECR** routing strategy, experts also select tokens for processing from assigned tokens according to affinity scores:

$$\begin{aligned}&\left(\tilde{E}_{t1},\ldots,\tilde{E}_{t\ell}\right)=\text{Top-}\ell\left(\{\delta_{t1},\ldots,\delta_{tn}\}\right),\\&\tilde{I}_{tk}\in[n],\forall t\in[s],k\in[\ell].\end{aligned}\tag{5}$$

and then the expert to choose its Top-ℓ tokens where ℓ is determined by a threshold of the sum of affinity scores:

$$\begin{aligned}&(I_{1i},\ldots,I_{Ci})=\text{Bottom-}C\left(\left\{t\in[s]:\exists j\in[\ell],\tilde{I}_{tj}=i\right\}\right),\\&I_{ki}\in[s]\cup\text{None},\forall i\in[n],k\in[C].\end{aligned}\tag{6}$$

Such bidirectional selection mechanism motivates each expert to receive a certain number of tokens with the highest affinity score to itself, thereby achieving a resonance effect. The resonance effect can help mitigate the homogenization in MoE.

Locality Loss. Feed-forward network (FFN) layers are commonly employed in expert networks, allowing each expert to learn independently as a separate neural network, thus preventing interference between samples. This mechanism leads to a severe load imbalance, as experts frequently selected in the early stages are more likely to be chosen in later stages. To mitigate this skewness in token allocation, the auxiliary loss (Shazeer et al., 2017) has been proposed. Building upon the auxiliary loss, our work introduces a loss bias term based on data locality, represented as $L_{\text{loc}}=\mu\text{KL}(D_{\text{c}}||D_{\text{l}})=-\mu\int D_{\text{c}}(x)\ln[\frac{D_{\text{l}}(x)}{D_{\text{c}}(x)}]\text{d}x$, i.e., the Kullback-Leibler (KL) divergence of the current distribution $D_{\text{c}}(x)$ and the fully localized distribution $D_{\text{l}}(x)$. This loss term serves as a soft constraint, encouraging tokens to be sent to experts residing on the same node, thereby mitigating the substantial overhead incurred by partial inter-node communication.

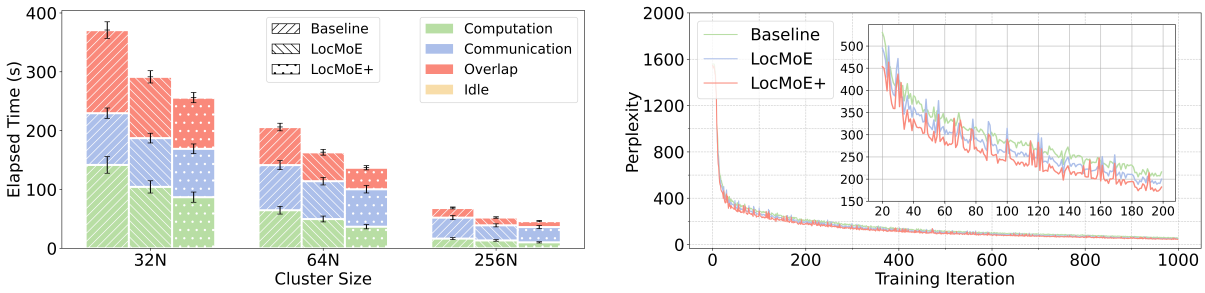
2.2 Training Strategy

Token Distribution Dynamics under Expert Routing. Under the premises of orthogonal gating weights and a data distribution approaching uniformity, the previous studies demonstrate that the expert capacity is closely related to the angle between the gating weights and tokens. For large scale of the activation, the lower bound of expert capacity is proven to exist and is represented as $C_{\min}=\frac{1}{n}\exp\{d\delta_{\max}^2/(2-\delta_{\max}^2)\}$.

The hybrid TCR+ECR bidirectional selection routing, introduced in the model structure, is exemplified in the figure. If the feature fragment corresponding to the k-th dimension of the gating weight for a particular token is more prominent, then that token will be routed to the k-th expert. If among all tokens routed to the k-th expert, there is a certain probability of the presence of class-discriminative tokens, then the capacity C must be set to a larger value to ensure the inclusion of sufficient class-discriminative tokens. The router proposed in this paper is a hybrid of TCR and ECR modes. After determining the expert to which a token will be routed, scores are calculated for the tokens assigned to each expert, and a Top-ℓ selection is performed, where ℓ^* is determined by a threshold of the sum of scores. Subsequent theoretical analysis will demonstrate the effectiveness of this hybrid routing scheme.

2.2.1 Dynamic Lower Bound Module for Expert Capacity in ETR

To explain the motivation of our method, we show some theoretical insights in this section. Our theoretical analyis is bulit on Chowdhury et al. (2023), where they make the following data assumption:



(a) (b)

Figure 3: (a) The average composition of computation, communication, overlap, and idle with different schemes and cluster sizes. (b) The perplexity during training iterations with different schemes.

Assumption 1 (data assumption)**.** *Each input $\boldsymbol{x} \in \mathcal{R}^{sd}$ with s tokens is comprised of one class-discriminative pattern $\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n \in \mathcal{R}^d$, with each decides the label in $[n]$, and $s-1$ class-irrelevant patterns $\boldsymbol{r} \sim \mathcal{N}$ for certain distribution $\mathcal{N}$. For example, $\boldsymbol{x} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{o}_1, \boldsymbol{r}_3, \ldots, \boldsymbol{r}_{s-1})$ has label 1, where $\boldsymbol{r}_i \overset{i.i.d.}{\sim} \mathcal{N}, \forall i \in [s-1]$.*

Based on Assumption 1, Chowdhury et al. (2023) demonstrated that the training of MoE go through two phases:

Phase 1: Router training (Chowdhury et al., 2023, Lemma 4.1 and Assumption 4.4), which makes class-discriminative patterns all to the corresponding expert. This process ensures that each expert only receives the class-discriminative tokens related to the specific class.

Phase 2: Expert training (Chowdhury et al., 2023, Theorem 4.2 and Theorme 4.5), which makes each expert learn to predict the label based on its class-discriminative inputs from Phase 1. This process is designed to establish each expert's ability to handle and solve problems.

Hence, the traning of an input in the current step is valid if the class-discriminative patterns is correctly dispatched. To quantitatively measure the difference between TCR and ECR, we define **training success rate** of input motivated by the training process of MoE.

Definition 2 (training success rate)**.** *We say the input $\boldsymbol{x} \in \mathcal{R}^{sd}$ with s tokens succeed in training if the class-discriminative pattern in $\boldsymbol{x}$, e.g., $\boldsymbol{o}_i$ is correctly dispatched to i-th expert. We further define **training success rate** as the probability that the input succeed in training.*

Furthermore, to show the quantitative comparison of TCR and ECR in training success rate, we need following asssumptions and notations of token patterns.

Assumption 3 (class-discriminative)**.** *We assume the location and feature of class-discriminative pattern is uniformly distribute in $[s]$ and $[n]$, i.e.,*

$$i \sim \mathrm{Unif}([s]), \boldsymbol{x}_i \sim \mathrm{Unif}\left(\{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n\}\right). \tag{7}$$

We also assume that $\forall i \in [n], \boldsymbol{o}_i$ should be sent to the i-th expert, and define the true positive probability in token choice setting is no worse than the uniform dispatch as below

$$\mathcal{P}(\delta_{\boldsymbol{o}_i,i} \geq \delta_{\boldsymbol{x}_j,i}, \forall j \in [s]) = p_i \geq 1/n, \forall i \in [n]. \tag{8}$$

Assumption 4 (class-irrelevant)**.** *The distribution of class-irrelevant patterns is isotropy, i.e.,*

$$\mathcal{P}(\boldsymbol{r} \sim \mathcal{N}, \delta_{\boldsymbol{r},i} \geq \delta_{\boldsymbol{x}_j,i}, \forall j \in [s]) = 1/n, \forall i \in [n]. \tag{9}$$

And we define the false positive probability in expert choice setting as

$$\mathcal{P}(\boldsymbol{r} \sim \mathcal{N}, \delta_{\boldsymbol{r},i} \geq \delta_{\boldsymbol{o}_i,i}) = q_i, \forall i \in [n], \tag{10}$$

which measures the possibility that expert i chooses the wrong token $\boldsymbol{r}$ instead of the correct token $\boldsymbol{o}_i$.

Assumption 3 assumes the valid token is uniformly distributed in training samples due to the massive amounts of data nowadays. Assumption 4 assumes the invalid tokens can be uniformly dispatched to experts since the invalid tokens do not provide supervised signal to router and experts in training. We consider such uniform settings are common assumptions in theoretical analyis. Now we compute the training success rate of TCR and ECR.

Theorem 5. *Under Assumptions 3 and 4, the training success rate of TCR in each sample $\boldsymbol{x}$ is*

$$\mathcal{P}(\textit{TCR succeed}) = \Theta\big(C \sum_{i=1}^{n} p_i/s\big), \tag{11}$$

and the training success rate of ECR is $\forall i \in [n]$,

$$\mathcal{P}(\textit{ECR succeed}) \begin{cases} \leq \frac{1}{n}\sum_{i=1}^{n} e^{-\frac{(s-1)q_i}{8}}, & C \leq (s-1)q_i/2, \\ \geq 1 - e^{-3C/16}, & C \geq 2sq_i. \end{cases} \tag{12}$$

Corollary 6. *In practice, For constant number of experts (Jiang et al., 2024a), i.e.,* $n = \Theta(1)$*, and* $C < s$ *to save computation cost. We have the following lower bound for capacity* C *to ensure high training success rate:*

1. *Suppose* $q_i = \Theta(1)$*. Then TCR is much better than ECR, and we only need* $C = \Theta(s)$*.*
2. *Suppose* $\forall i \in [n], sq_i \leq C^*$ *for some* $C^* > 0$*. Then ECR is much better than TCR, and we only need* $C \geq 2C^*$*.*

Remark 7. *We explain the benefit of swicthing TCR to ECR during training based on Theorem 5 and feature distrution during training.*

At the beginning of training, the model seldom learn the task. Then the feature of class-irrelevant tokens is nearly isotropy, e.g., uniformly distrbute around the sphere (see Appendix), leading to $q_i = \Theta(1)$*. The succed rate of TCR with the form* C/s *is better than ECR with the form* e^{-s}*. Thus we should choose TCR with a large capacity* $C = \Theta(s)$ *to improve the success rate of training samples.*

After training for some iterations, the experts can roughly distinguish the class-irrelevant and discriminative patterns, leading to $q_i \ll 1$ *or* $sq_i \leq C^*$ *for some* $C^* > 0$ *(see Appendix). Then ECR with success rate nearly* 1 *is better than TCR with the form* C/s *as long as* $C \geq 2C^*$*. Thus we should choose ECR with a small capacity* $C = \Theta(1)$ *to improve the success rate of training samples.*

Indeed, we find that Chowdhury et al. (2023, the definition of ℓ^**) consider the ECR setting and verify the benefit in sample complexity. They assume the maximum number of class-irrelevant patches that are close to class-discriminative patches are bounded, which has similar effect as* C^* *in our scene.*

2.3 Communication Optimization

The training framework employs the Communication Over Computation (CoC) optimization technique to address performance bottlenecks in LLM training. During forward propagation in LLMs, the ColumnParallelLinear and RowParallelLinear components involve sequentially dependent computation (matrix multiplication) and communication (collective operations like AllReduce, AllGather, and ReduceScatter). These dependencies lead to inefficient serial execution. CoC decomposes these tasks into finer-grained subtasks and merges computation and communication into single kernels, such as MATMUL_ALL_REDUCE and MATMUL_REDUCE_SCATTER, utilizing MTE's remote memory access capabilities. This approach allows for pipeline-style parallel execution and overlapping of computation and communication, significantly enhancing overall efficiency.

3 Experiments

3.1 Experimental Setup

This study employs the Mixtral 8×7B model, incorporating our proposed approach. The Mixtral model, comprising 46.7 billion parameters and utilizing Group Query Attention (GQA), features 32 sparse expert blocks with 8 experts in the MoE Feedforward layer, where each token engages the top 2 experts for processing. Given the prevalence of long-text corpora in our application scenarios, we extended the sequence length to 32,768 and implemented tailored parallel strategies for cluster scales of 32N, 64N, and 256N, encompassing tensor, pipeline, data, and expert parallelism, with a consistent global batch size of 128. For the three cluster scales of 32N, 64N, and 256N, the parallel strategies are set as follows: 32N - tensor parallel (TP=4) / pipeline parallel (PP=4) / data parallel (DP=2) / expert parallel (EP=2), 64N - TP=8 / PP=4 / DP=2 / EP=2, and 256N - TP=8 / PP=8 / DP=4 / EP=2. Other details of experimental setup including datasets, environment, and metrics, can be seen in Appendix.

3.2 Efficiency Promotion and Memory Footprint Reduction

As detailed in Section **Method**, we consistently use Top-1 routing to ensure the routing implementation aligns with our theoretical framework. The Baseline model utilizes a limited expert capacity mode instead of the *groupedGEMM* scheme, which avoids token dropping, with the capacity factor set to 1.1. LocMoE considers data distribution uniformity and estimates expert capacity using a lower bound formula derived from its theoretical conclusions in the first batch, maintaining it as a constant during subsequent training. Our approach (abbreviate to "LocMoE+" in figures) fixes the range of score sums, processes hidden states, and calculates

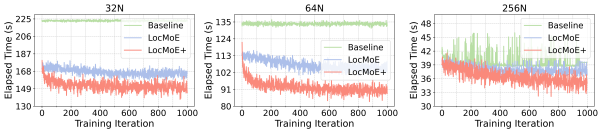


Figure 4: The time consumption during training iterations with different schemes and cluster sizes.

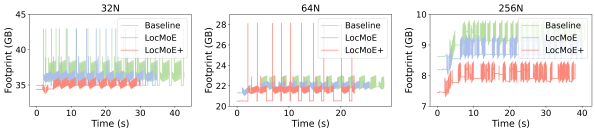


Figure 5: print recorded in one acquisition cycle with different schemes and cluster sizes.

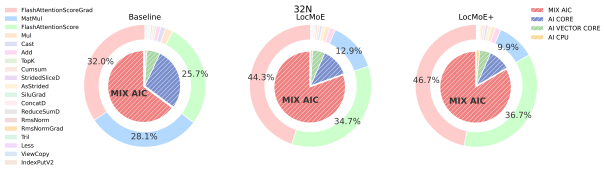


Figure 6: The distribution of time consumption for operators.

current expert capacity. The subsequent analysis addresses the training time, convergence, and memory usage efficiency of these schemes on multiple sizes of Ascend clusters.

Figure 3a illustrates the time consumption of these methods during the first 1000 iterations of training. Due to initialization and some unstable factors, time consumption is recorded starting from the 5th iteration. The Baseline model's time consumption is relatively stable. As iterations increase, LocMoE's time consumption slightly decreases, particularly in 32N and 64N, consistent with the conclusion that locality loss is effective only when the number of experts is greater than or equal to the number of nodes. Our approach incurs slightly higher time consumption than LocMoE due to the computational overhead of token rearrangement. However, as token features converge, the required tokens gradually decrease and stabilize, leading to a decline in time consumption, which remains stable in subsequent training processes. Overall, our approach reduces training time by 2.9% to 13.3% compared to LocMoE, and by 5.4% to 46.6% compared to the Baseline.

We select 10 iterations at equal intervals from the training iterations to collect data on the time consumption of computation, communication, overlap, and idle periods, as shown in Figure 3a. It is important to note that the data collection operation also introduces some overhead. After integrating LocMoE and our approach, the time consumption of each component decreases, with a significantly greater reduction in computation overhead compared to communication overhead. Additionally, as the cluster size increases, the proportion of computation/communication overlap decreases, and the magnitude of the reduction in computation overhead diminishes. Figure 3b illustrates perplexity as a measure of convergence. The convergence curves of these approaches indicate normal loss convergence, with our approach not adversely impacting convergence.

The proportional time consumption at the operator level is depicted in Figure 6. Among the components, AI CORE efficiently executes matrix multiplications and convolutions in AI algorithms; AI VECTOR CORE accelerates vector operations through parallel processing; MIX AIC integrates different types of operators and optimizes for multiple tasks; AI CPU is optimized in hardware and instruction sets to better support AI algorithms. Our approach selects fewer tokens, resulting in a 17× performance improvement in the FFN *MatMul* operator compared to the Baseline and a 2.6× improvement compared to LocMoE. This leads to an overall 2.8× reduction in the cumulative time consumption of the *MatMul* operator and a 2.6× decrease in Cube computing load. However, the proportions of *TopK* and *IndexPutV2*, related to rearrangement, show a slight increase.

We select a single iteration during the stable training period and describe the per-device memory usage (Allocated) using the first 100,000 samples from its memory monitoring, as shown in Figure 5. Overall, our approach achieves memory usage reduction of 4.57% to 16.27% compared to the Baseline and 2.86% to 10.5% compared to LocMoE. As cluster size increases, the proportion of computational overhead decreases, and the gap in memory usage narrows. Additionally, instantaneous memory peaks gradually disappear, and the fluctuation amplitude of short-term memory also diminishes.

3.3 The Performance of Downstream Tasks

To enhance the model's conversational capabilities and adaptability to downstream task, we fine-tuned the pre-trained models. As shown in Figure 7, with sufficient supervised fine-tuning (SFT), our approach achieves an average improvement of approximately 20.1% in 16 sub-capabilities of

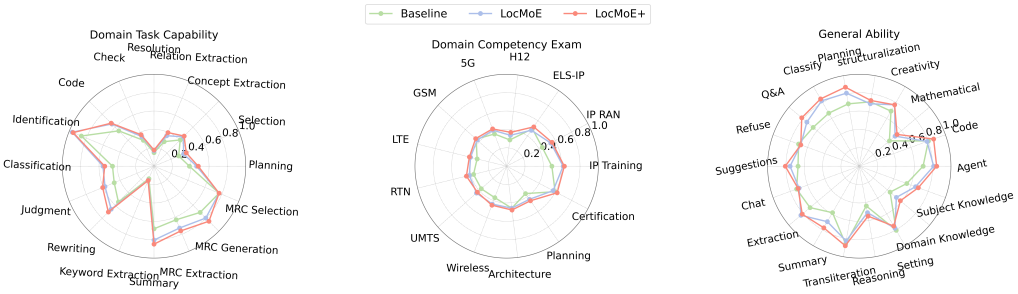


Figure 7: The performance on three categories of GDAD.

Domain Task Capability, which is a portion of General and Domain-specific Assessment Dataset (GDAD), compared to the Baseline, and an increase of about 3.5% compared to LocMoE. The *Rewriting* and *Summary* capabilities show the highest improvement, with a 28.2% increase compared to the Baseline and a 6.7% increase compared to LocMoE. In the 13 tests of *Domain Competency Exam*, our approach demonstrates an average improvement of 16% relative to the Baseline and an average increase of approximately 4.8% compared to LocMoE. The *IP Training* in the digital communications domain shows the most significant improvement, with a 27.3% increase compared to the Baseline and a 3.0% increase compared to LocMoE. Among the 18 sub-capabilities of *General Ability*, our approach exhibits an improvement of about 13.9% relative to the Baseline and an average increase of 4.8% compared to LocMoE. The capability of *Planning* demonstrates the highest improvement, with a 26.8% increase compared to the Baseline and a 2.92% increase compared to LocMoE.

Table 1 presents the holistic evaluation results for multiple datasets, where GDAD-1 represents *Domain Task Capability*, and the other metrics follow accordingly. Notably, due to the 6:4 ratio of Chinese to English data in our incremental pre-training domain data and the 7:3 ratio in the fine-tuning data, our approach achieves an improvement of approximately 13.6% compared to the Baseline and 2.8% compared to LocMoE in the GPQA (Rein et al., 2023) evaluation, despite the limited data available for training. During incremental training and fine-tuning, we incorporated substantial telecommunications domain knowledge, questions, and case studies. TeleQnA (Maatouk et al., 2023), the first benchmark dataset designed to evaluate the knowledge of LLMs in telecommunications, effectively measures the model's capabilities in this domain. Consequently, our approach comprehensively surpasses both the Baseline and LocMoE on this specific dataset.

Table 1: Performance promotion obtained by our approach on different datasets.

	GDAD					
	GDAD-1	GDAD-2	GDAD-3	Avg	GPQA	TeleQnA
Baseline	47.8	43.0	65.4	52.8	29.5	62.1
LocMoE	55.5	47.6	71.1	59.0	32.6	67.6
LocMoE+	**57.4**	**49.9**	**74.5**	**61.5**	**33.5**	**68.8**

4 Conclusion

In this paper, we propose a novel expert routing framework that enhances MoE efficiency through three key innovations: an efficient routing mechanism with lightweight computation, a bidirectional expert-token resonance selection mechanism, which combined ECR and TCR, and a dynamic capacity bounds module. The framework integrates orthogonal feature extraction and optimized expert localization loss, effectively addressing expert homogeneity while improving routing performance. Our local expert strategy demonstrates advantages in both load balancing and communication efficiency. Experimental results validate the effectiveness of the proposed framework across multiple benchmarks. Our approach achieves performance improvements up to 46.6% (32N) compared to the Baseline and 13.3% (32N) compared to LocMoE, while reducing memory usage by up to 16.27% and 10.5%, respectively. To evaluate model performance, all models are evaluated with the open-source datasets GPQA and TeleQnA, and closed domain benchmark GDAD. In downstream tasks, our approach outperforms the Baseline by 14.1%, 13.6%, and 9.7% on GDAD, GPQA, and TeleQnA, respectively. Future work may explore methods to compress communication data to further reduce communication overhead.

Limitations

Despite our comprehensive evaluation efforts, several limitations of this study warrant acknowledgment. First, our assessment framework does not encompass certain important capabilities, such as role-playing scenarios and multilingual performance evaluations. These aspects could provide additional insights into the model's versatility and practical applications. Furthermore, due to computational resource constraints, our investigation was limited to models with parameters under 100B. This restriction prevented us from extending our experimental framework to larger-scale models, including current state-of-the-art architectures such as DeepSeek V3/R1. A more extensive study incorporating these larger models could potentially reveal additional insights about the scalability of our approach. Additionally, our cluster's inter-node bandwidth limitations and our primary focus on large-scale sparse expert architectures resulted in a less thorough investigation of pipeline parallelism and All-to-All communication strategies. A more comprehensive analysis of these aspects could potentially yield superior computation and communication efficiency. Future work could explore these directions to achieve more optimal performance in distributed training scenarios.

References

Mohammed Nowaz Rabbani Chowdhury, Shuai Zhang, Meng Wang, Sijia Liu, and Pin-Yu Chen. 2023. Patch-level routing in mixture-of-experts is provably sample-efficient for convolutional neural networks. In *International Conference on Machine Learning*, pages 6074–6114. PMLR.

Fan Chung and Linyuan Lu. 2006. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics*, 3(1):79–127.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, and 1 others. 2022. Unified scaling laws for routed language models. In *International conference on machine learning*, pages 4057–4086. PMLR.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Stablemoe: Stable routing strategy for mixture of experts. *arXiv preprint arXiv:2204.08396*.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, and 1 others. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

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

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, and Tao Lin. 2024. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. *arXiv preprint arXiv:2405.14297*.

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

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, and 1 others. 2024b. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024a. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Ziheng Jiang, Haibin Lin, Yinmin Zhong, Qi Huang, Yangrui Chen, Zhi Zhang, Yanghua Peng, Xiang Li, Cong Xie, Shibiao Nong, and 1 others. 2024b. {MegaScale}: Scaling large language model training to more than 10,000 {GPUs}. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 745–760.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

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*.

Jiamin Li, Yimin Jiang, Yibo Zhu, Cong Wang, and Hong Xu. 2023. Accelerating distributed {MoE} training and inference with lina. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 945–959.

Jing Li, Zhijie Sun, Xuan He, Li Zeng, Yi Lin, Entong Li, Binfan Zheng, Rongqian Zhao, and Xin Chen. 2024. Locmoe: A low-overhead moe for

large language model training. *arXiv preprint arXiv:2401.13920*.

Heng Liao, Jiajin Tu, Jing Xia, Hu Liu, Xiping Zhou, Honghui Yuan, and Yuxing Hu. 2021. Ascend: a scalable and unified architecture for ubiquitous deep neural network computing: Industry track paper. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 789–801. IEEE Computer Society.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Tianlin Liu, Mathieu Blondel, Carlos Riquelme, and Joan Puigcerver. 2024b. Routers in vision mixture of experts: An empirical study. *arXiv preprint arXiv:2401.15969*.

Ang Lv, Ruobing Xie, Yining Qian, Songhao Wu, Xingwu Sun, Zhanhui Kang, Di Wang, and Rui Yan. 2025. Autonomy-of-experts models. *arXiv preprint arXiv:2501.13074*.

Ali Maatouk, Fadhel Ayed, Nicola Piovesan, Antonio De Domenico, Merouane Debbah, and Zhi-Quan Luo. 2023. Teleqna: A benchmark dataset to assess large language models telecommunications knowledge. *arXiv preprint arXiv:2310.15051*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2021. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595.

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*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Xing Wang, Han Zhang, and Zhaohui Du. 2023. Multi-scale noise reduction attention network for aero-engine bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*.

Zhitian Xie, Yinger Zhang, Chenyi Zhuang, Qitao Shi, Zhining Liu, Jinjie Gu, and Guannan Zhang. 2024. Mode: A mixture-of-experts model with mutual distillation among the experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16067–16075.

Yuanhang Yang, Shiyi Qi, Wenchao Gu, Chaozheng Wang, Cuiyun Gao, and Zenglin Xu. 2024. Enhancing efficiency in sparse models with sparser selection. *arXiv preprint arXiv:2403.18926*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yanqi Zhou, Nan Du, Yanping Huang, Daiyi Peng, Chang Lan, Da Huang, Siamak Shakeri, David So, Andrew M Dai, Yifeng Lu, and 1 others. 2023. Brainformers: Trading simplicity for efficiency. In *International Conference on Machine Learning*, pages 42531–42542. PMLR.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, and 1 others. 2022. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114.

A Appendix

B Missing Proof

B.1 Auxiurary Results

Lemma 8 (Theorem 4 in (Chung and Lu, 2006))**.** *Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be n independent random variables with*

$$\mathcal{P}(\boldsymbol{X}_i = 1) = p_i, \mathcal{P}(\boldsymbol{X}_i = 0) = 1 - p_i. \tag{13}$$

We consider the sum $\boldsymbol{X} = \sum_{i=1}^n X_i$, with expectation $\mathcal{E}(X) = \sum_{i=1}^n p_i$. Then we have

$$\begin{aligned} &\text{(Lower tail)} \quad \mathcal{P}(\boldsymbol{X} \leq \mathcal{E}\boldsymbol{X} - \lambda) \leq e^{-\frac{\lambda^2}{2\mathcal{E}\boldsymbol{X}}}, \\ &\text{(Upper tail)} \quad \mathcal{P}(\boldsymbol{X} \geq \mathcal{E}\boldsymbol{X} + \lambda) \leq e^{-\frac{\lambda^2}{2(\mathcal{E}\boldsymbol{X} + \lambda/3)}}. \end{aligned} \tag{14}$$

B.2 Proof of Theorem 5

Proof. 1) For the TCR, denote

$$s_i = \left|\{t < k : \boldsymbol{x}_t \text{ sent to expert } i, \boldsymbol{x}_k = \boldsymbol{o}_i\}\right|, \forall i \in [n] \tag{15}$$

as the top class-irrelevant token number candidated to the i-th expert before the valid token. Then by Assumption 4, each class-irrelevant token uniformly gives to any expert, leading to $s_i | (\boldsymbol{x}_k = \boldsymbol{o}_i) \sim \mathcal{B}(k-1, 1/n)$ (Binomial distribution), i.e., $\forall t \in [k-1]$,

$$\mathcal{P}(s_i = t | \boldsymbol{x}_k = \boldsymbol{o}_i) = \binom{k-1}{t} \cdot \left(\frac{1}{n}\right)^t \left(1 - \frac{1}{n}\right)^{k-1-t}. \tag{16}$$

Then we could derive that

$$\begin{aligned} &\mathcal{P}(\boldsymbol{x} \text{ succeed in training}) \\ &= \sum_{i=1}^n \mathcal{P}(\boldsymbol{o}_i \text{ sent to expert } i | \boldsymbol{o}_i \text{ is in } \boldsymbol{x}) \cdot \mathcal{P}(\boldsymbol{o}_i \text{ is in } \boldsymbol{x}) \\ &= \frac{1}{ns} \sum_{i=1}^n \sum_{k=1}^s p_i \mathcal{P}(s_i < C | \boldsymbol{x}_k = \boldsymbol{o}_i) \\ &= \frac{1}{ns} \sum_{i=1}^n p_i \left(C + \sum_{k=C+1}^s \mathcal{P}(s_i < C | \boldsymbol{x}_k = \boldsymbol{o}_i) \right). \end{aligned}$$

Note that $\mathcal{E}s_i = (k-1)/n$. When $k \geq 2nC$, by lower tail bound in Lemma 8, we get

$$\mathcal{P}(s_i < C | \boldsymbol{x}_k = \boldsymbol{o}_i) \leq e^{-\frac{(k-1-n(C-1))^2}{2(k-1)n}} \leq e^{-\frac{k-1}{8n}}. \tag{17}$$

Hence, we get the upper bound that

$$\begin{aligned} &\mathcal{P}(\boldsymbol{x} \text{ succeed in training}) \\ &\overset{\text{(B.2)}}{\leq} \frac{1}{ns} \sum_{i=1}^n \sum_{k=1}^s p_i \mathcal{P}(s_i < C | \boldsymbol{x}_k = \boldsymbol{o}_i) \\ &= \frac{1}{ns} \sum_{i=1}^n p_i \left(2nC + \sum_{k=2nC+1}^s \mathcal{P}(s_i < C | \boldsymbol{x}_k = \boldsymbol{o}_i) \right) \\ &\leq \frac{1}{ns} \sum_{i=1}^n p_i \left(2nC + \sum_{k=2nC}^{s-1} e^{-\frac{k}{8n}} \right) \\ &\leq \frac{1}{ns} \sum_{i=1}^n p_i \left(2nC + \frac{e^{-\frac{C}{4}}}{1 - e^{-\frac{1}{8n}}} \right) \\ &\overset{(i)}{\leq} \frac{1}{ns} \sum_{i=1}^n p_i \left(2nC + (8n+1) e^{-\frac{C}{4}} \right) \leq \frac{10C \sum_{i=1}^n p_i}{s}, \end{aligned}$$

where (i) uses the inequality that $e^{-t} \leq 1/(1+t), \forall t \geq 0$.

Moreover, for $1 + \frac{nC}{4} \leq k \leq 1 + \frac{nC}{2}$, i.e., $2(k-1) \leq nC \leq 4(k-1)$, by upper tail bound in Lemma 8, we get

$$\begin{aligned} \mathcal{P}(s_i < C | \boldsymbol{x}_k = \boldsymbol{o}_i) &= 1 - \mathcal{P}(s_i \geq C | \boldsymbol{x}_k = \boldsymbol{o}_i) \\ &\geq 1 - e^{-\frac{3(nC-k+1)^2}{2n[2(k-1)+nC]}} \geq 1 - e^{-\frac{k-1}{4n}}. \end{aligned}$$

Hence, we get the lower bound that

$$\begin{aligned} &\mathcal{P}(\boldsymbol{x} \text{ succeed in training}) \\ &\overset{\text{(B.2)}}{\geq} \frac{1}{ns} \sum_{i=1}^n \sum_{k=1}^s p_i \mathcal{P}(s_i < C | \boldsymbol{x}_k = \boldsymbol{o}_i) \\ &= \frac{1}{ns} \sum_{i=1}^n p_i \left(\sum_{k=\lceil 1+nC/4 \rceil}^{\lfloor 1+nC/2 \rfloor} \mathcal{P}(s_i < C | \boldsymbol{x}_k = \boldsymbol{o}_i) \right) \\ &\geq \frac{1}{ns} \sum_{i=1}^n p_i \left(\frac{nC}{4} - 1 - \sum_{k=\lceil 1+nC/4 \rceil}^{\lfloor 1+nC/2 \rfloor} e^{-\frac{k-1}{4n}} \right) \\ &\geq \frac{1}{ns} \sum_{i=1}^n p_i \left(\frac{nC}{4} - 1 - \frac{e^{-\frac{C}{16}}}{1 - e^{-\frac{1}{4n}}} \right) \\ &\overset{(i)}{\geq} \frac{1}{ns} \sum_{i=1}^n p_i \left(\frac{nC}{4} - 2 - (4n+1) e^{-\frac{C}{16}} \right) \geq \frac{C \sum_{i=1}^n p_i}{5s}, \end{aligned}$$

where (i) uses the inequality that $e^{-t} \leq 1/(1+t), \forall t \geq 0$, and the final inequality needs $C \geq 48$, which can be satisified in common experiments. Combining the upper and lower bounds, we obtain the desired result.

2) For the ECR, denote s_i as the class-irrelevant token number with the score larger than $\boldsymbol{o}_i$ for i-th expert. By Assumption 4, we derive that $s_i \sim \mathcal{B}(s-1, q_i), \forall i \in [n]$.

$$\begin{aligned} &\mathcal{P}(\boldsymbol{x} \text{ succeed in training}) \\ &= \sum_{i=1}^n \mathcal{P}(\text{expert } i \text{ choose } \boldsymbol{o}_i | \boldsymbol{o}_i \text{ is in } \boldsymbol{x}) \mathcal{P}(\boldsymbol{o}_i \text{ is in } \boldsymbol{x}) \\ &= \frac{1}{n} \sum_{i=1}^n \mathcal{P}(s_i \leq C - 1, s_i \sim \mathcal{B}(s-1, q_i)) \end{aligned}$$

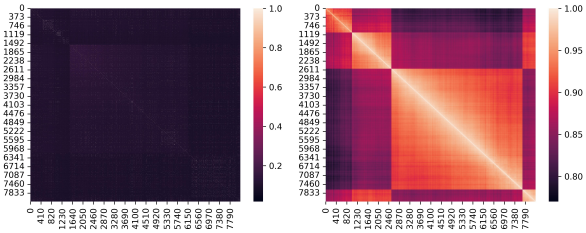


Figure 8: The correlation matrix of one training sample feature before (left) and after (right) training.

If $C-1 \leq (s-1)q_i/2$, by lower tail bound in Lemma 8 with $\lambda = (s-1)q_i - (C-1) < \mathcal{E}s_i$, we obtain that

$$\mathcal{P}(s_i \leq C-1) \leq e^{-\frac{(s-1)q_i}{2}\left(1-\frac{C-1}{(s-1)q_i}\right)^2} \leq e^{-\frac{(s-1)q_i}{8}}. \tag{18}$$

If $C \geq 2(s-1)q_i$, by upper tail bound in Lemma 8 with $\lambda = C-(s-1)q_i > 0$, we obtain that

$$\begin{aligned}\mathcal{P}(s_i \leq C-1) &= 1-\mathcal{P}(s_i \geq C) \\ &\geq 1-e^{-\frac{[C-(s-1)q_i]^2}{2(C+2(s-1)q_i)/3}} \geq 1-e^{-\frac{3C}{16}}.\end{aligned}$$

Hence, we conclude Eq. (12). □

C Token Feature Distribution

We also validate the feature distribution before and after MoE training shown in Figure 8. We can see before training, all 8192 tokens in one training sample are nearly orthogonal with correlation coefficient near zero, which verifies the isotropy distribution assumption in the first bullet of Remark 7. After training, the token features are nearly aligned with correlation coefficien large than 0.8. We can also observe that neighbouring tokens share similar features, and clear block feature behavior, meaning that the token features are relatively separated and the number of tokens in each cluster is bounded, which somehow matches the distribution assumption in the second bullet of Remark 7.

D Experimental Setup

D.1 Datasets for Training and Fine-Tuning

The dataset used in this paper is a self-constructed dataset that integrates knowledge from multiple domains, including wireless, data communication, and cloud-core technologies. It comprises Chinese, English, and bilingual corpora. The corpora are parsed from various internal technical documents, such as iCase, blogs, Wiki, and feature documents. Taking iCase as an example, iCase is a case record of problem localization and handling processes, containing code, instructions, and corresponding logs. In addition, the above-mentioned domain-specific knowledge corpora are mixed with general corpora in a ratio of 1:5. The general corpora are collected from hundreds of websites, including online novels, cooking guides, movie reviews, and more. After cleaning, deduplication, and review operations, the dataset is thoroughly shuffled. A total of 4.19 billion tokens is sampled as the experimental pre-training dataset. To evaluate downstream tasks, this paper also adopt hybrid sft data items to fine-tune the pre-trained model. The dataset comprises 762,321 general question-answer pairs and 11,048 domain-specific question-answer pairs, with a general-to-domain ratio of 68:1. The general characteristics encompass multi-tasking, mathematical ability, coding ability, logical reasoning, multi-turn dialogue, knowledge reasoning, language understanding, text generation, multi-tasking, FunctionCall, CoT, MRC summarization, refusal to answer, Chinese, and English. The domain-specific characteristics include domain knowledge understanding, RAG, FunctionCall, information extraction, multi-turn dialogue, reading comprehension, paraphrasing, and intent recognition.

D.2 Experimental Environment

The experiments are conducted on a cluster composed of Ascend 910B3 NPUs, divided into three groups: 32 NPUs (hereinafter referred to as 32N, and so on), 64N, and 256N. The 910B3 series NPU contains 20 AI cores with a main frequency of 1.8GHz and a theoretical computing power of 313T under fp16 precision. The physical High Bandwidth Memory (HBM) of the 910B3 NPU is 64G, with an HBM frequency of 1.6GHz and an HBM bandwidth of 1.6T. Every 8 NPUs are mounted on the same Atlas 800T A2 server, which internally adopts a fullmesh networking scheme, meaning that any two NPUs are interconnected. The version of the Ascend Hardware Development Kit (HDK) is 23.0.2.1, and the version of the Compute Architecture for Neural Networks (CANN) suite is 7.0.0, which is the commercial release version for Q4 2023. The models in this paper use ModelLink, an LLM training framework based on the Ascend architecture, and run in the torch_npu 5.0.0 environment.

D.3 Evaluation Metrics and Datasets

To evaluate model performance, this paper designs a comprehensive metric called the General

and Domain-specific Assessment Dataset (GDAD), which consists of three evaluation systems: domain task capability, domain capability certification exam, and general capability. Among them, the domain task capability includes a total of 16 categories and 2,657 questions, such as domain logical reasoning; the domain capability certification exam includes a total of 13 categories and 13,968 questions, such as data communication; and the general capability includes a total of 18 categories and 1,435 questions, such as programming ability. The questions include objective and subjective questions in Chinese, English, and bilingual formats. For subjective questions, the cosine similarity between the model output and the standard answer is used as the score. In addition, this paper also employs GPQA (Rein et al., 2023) and TeleQnA (Maatouk et al., 2023) to evaluate the model's Chinese language capability.