

# Advancing MoE Efficiency: A Collaboration-Constrained Routing (C2R) Strategy for Better Expert Parallelism Design

Mohan Zhang\*, Pingzhi Li\*, Jie Peng, Mufan Qiu, Tianlong Chen

University of North Carolina at Chapel Hill

 <https://github.com/UNITES-Lab/c2r-moe>

## Abstract

Mixture-of-Experts (MoE) has successfully scaled up models while maintaining nearly constant computing costs. By employing a gating network to route input tokens, it selectively activates a subset of expert networks to process the corresponding token embeddings. However, in practice, the efficiency of MoE is challenging to achieve due to two key reasons: (1) *imbalanced expert activation*, which leads to substantial idle time during model or expert parallelism, and insufficient capacity utilization; and (2) *massive communication overhead*, induced by numerous expert routing combinations in expert parallelism at the system level. Previous works typically formulate it as the *load imbalance* issue characterized by the gating network favoring certain experts over others or attribute it to *static execution* which fails to adapt to the dynamic expert workload at runtime. In this paper, we exploit it from a brand new perspective, *i.e.*, a higher-order view and analysis of MoE routing policies: *expert collaboration and specialization* — where some experts tend to activate broadly with others (*collaborative*), while others are more likely to activate only with a specific subset of experts (*specialized*). Specifically, our experiments reveal that most experts tend to be overly collaborative, leading to increased communication overhead from repeatedly sending tokens to different accelerators. To this end, we (1) propose a novel collaboration-constrained routing (C2R) strategy to encourage more specialized expert groups, as well as to improve expert utilization, and (2) present an efficient implementation of MoE that further leverages expert specialization. With our proposed C2R design, we achieve an average performance improvement of 0.51% and 0.33% on LLaMA-MoE and Qwen-MoE respectively across **ten** downstream NLP benchmarks, and reduce the all2all communication costs between GPUs, bringing an extra 20%-30% total running time savings

on top of the existing SoTA, *i.e.* MegaBlocks.

## 1 Introduction

Scaling up the capacity of transformer models has proven to be an effective approach to enhancing model accuracy. However, as the amount of parameters increases, the immense computational and memory overheads have become a critical bottleneck (Kaplan et al., 2020; Clark et al., 2022). To address this challenge, the sparsely activated Mixture-of-Experts design has been introduced as a substitute for conventional Feed-Forward Networks (FFNs). It integrates the conditional computation mechanism in the network, where only a subset of parameters is activated at runtime. With such a property, MoE architecture has been demonstrated to successfully expand the model capacity without increasing the corresponding computational requirements (Fedus et al., 2021; Lepikhin et al., 2020), making it popular in various domains, such as language, vision, and others (Lepikhin et al., 2020; Riquelme et al., 2021; Kumatani et al., 2021).

Despite promising potentials, the dynamic nature of MoE, stemming from well-designed routing mechanisms (Pan et al., 2024; Fedus et al., 2021; Roller et al., 2021; Zhou et al., 2022), also introduces new challenges during training and inference. Extensive literature shows that the dynamic routing strategy of MoE often results in most tokens being routed to a specific subset of experts, which is termed as the load imbalance issue (Lewis et al., 2021; Clark et al., 2022; Hazimeh et al., 2021; He et al., 2022). This ineffective expert utilization not only leads to training instability but also hampers the full exploitation of model capacity (Fedus et al., 2021). Previous works have attempted to address this issue by introducing noise to gating network or using load-balancing loss to force uniform expert activation (Shazeer et al., 2017). However, the MoE model still suffers from massive communication overhead at the system level due to the

\*Equal contribution

inherently large space of expert routing combinations. When implementing expert parallelism, each input token is dispatched to several best-fit parallel experts and the outputs are then aggregated to forward to the next layer. Such a dispatch-aggregation process will incur tremendous redundancy if the selected experts are distributed across multiple accelerators (GPUs or nodes), making communication the inference bottleneck (He et al., 2022; Gale et al., 2023; Liu et al., 2024a).

This paper delves into this problem from a novel perspective: *expert collaboration and specialization*, to elucidate the routing behavior of MoE models. This newly proposed aspect of the MoE property enables a better understanding of its inference behavior and paves a novel path for the MoE routing mechanism tailored to efficient expert parallelism design. Specifically, we define an expert as *collaborative* if it tends to collaborate (co-activate) extensively with many other experts and *specialized* if it is primarily co-activated with a small group of specific experts. Our empirical analysis reveals that most experts are overly collaborative, meaning that each expert could potentially collaborate with any other expert to handle certain tokens, which leads to the aforementioned communication overhead. Consequently, we propose a novel collaboration-constrained routing (C2R) strategy at the model level to deliver a property of specialized expert groups, which provides a new chance to address this issue. Initially, we derive the most closely collaborating expert group for each expert from our experimental analysis. For each token, instead of directly routing it to the top-K experts, we first select its top-1 expert and then choose the remaining  $K - 1$  experts from its corresponding collaborating expert group. This routing mechanism dynamically reduces the space of expert routing combinations, encouraging more specialized expert groups where only experts from the same group are collaborative. Subsequently, we co-locate each expert within the same expert group on the same computing unit to minimize the communication overhead at the system level. Finally, the model-system co-design achieves a Pareto optimal balance between the accuracy and efficiency of MoEs. Evaluation across multiple benchmarks indicates the promising potential of our approach. Our key contributions are summarized below:

- We present a new perspective, *i.e.*, expert collaboration and specialization, to elucidate

the routing behavior of MoE models and propose a novel collaboration-constrained routing (C2R) strategy to enhance expert utilization.

- Leveraging the new property of specialized expert groups, we further propose an optimized expert parallelism design at the system level to reduce communication overhead by minimizing the communication redundancy of tokens.
- By combining the proposed techniques at both the model and system levels, we achieve up to 24.9% potential reduction in total inference wall-clock time, without compromising model accuracy. Furthermore, we even observe an average performance improvement of 0.51% and 0.33% on LLaMA-MoE and Qwen-MoE, respectively, across ten benchmarks.

## 2 Related Works

**Mixture of Experts (MoE).** MoE is a distinct neural network architecture where the model’s parameters are divided into multiple sub-modules, known as experts. Computations are conditionally performed by activating certain experts based on the input (Jacobs et al., 1991; Jordan and Jacobs, 1994; Chen et al., 1999; Yuksel et al., 2012). Traditional dense MoEs are computationally heavy because they engage all experts for every input token (Eigen et al., 2013). Recent advancements (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2021) have demonstrated the effectiveness of sparsely activated MoEs (SMoEs) during both training and inference. SMoEs significantly reduce computing costs and enable language models to scale to unprecedented sizes, reaching trillions of parameters (Fedus et al., 2021). This efficient methodology has led to the growing adoption of SMoEs in a variety of natural language processing (Shazeer et al., 2017; Lepikhin et al., 2020; Zhou et al., 2022; Zhang et al., 2021; Zuo et al., 2022; Jiang et al., 2021) and computer vision tasks (Riquelme et al., 2021; Eigen et al., 2013; Ahmed et al., 2016; Gross et al., 2017; Wang et al., 2020; Yang et al., 2019; Abbas and Andreopoulos, 2020; Pavlitskaya et al., 2020).

**Challenges in Efficient MoE Training and Inference.** Mixture of Experts (MoE) models face significant challenges in efficient training and inference, primarily due to insufficient specialization, load imbalance, and dynamic routing strategies (Fedus et al., 2021; Lepikhin et al., 2020; Shazeer

et al., 2017). To address these issues, researchers have focused on improving routing algorithms and enhancing communication efficiency (Lewis et al., 2021; Clark et al., 2022; Nie et al., 2022; Yu et al., 2024; Roller et al., 2021; Zhou et al., 2022; Hwang et al., 2023; He et al., 2022). Specifically: (a) MegaBlocks (Gale et al., 2023) expresses MoE layer computation as block-sparse operations to accommodate imbalanced token-expert assignments. They introduced dropless-MoEs (dMoEs) and developed high-performance GPU kernels for block-sparse matrix products. (b) Tutel (Hwang et al., 2023) introduced a framework that implements adaptive parallelism switching, allowing dynamic adjustment of parallelism strategies without overhead. They also developed adaptive pipelining and a 2-dimensional hierarchical All-to-All algorithm for efficient MoE computation. (c) BASE Layer (Lewis et al., 2021) formulates MoE routing as a linear assignment problem, maximizing token-expert affinity under fully balanced allocation constraints. This method eliminates the need for load-balancing loss functions and capacity factors used in previous approaches.

Despite these advancements, existing research still focuses primarily on accelerating MoE models by adjusting the routing relationship between tokens and experts. Our method, however, goes a step further by discussing routing behavior from a higher perspective, *i.e.*, the expert collaboration and specialization.

### 3 Methodology

#### 3.1 Preliminary

**Sparsely Activated MoE.** In this paper, we focus on sparsely activated MoE models, which can increase model capacity with nearly constant computational overhead. The key components include an input-dependent sparse routing network  $g(x)$  and a group of  $N$  experts  $\mathcal{E} = \{E_i\}_{i=1}^N$ , as shown in Figure 1 (a). For each input token  $x$ , the routing network first calculates the probability of  $x$  with respect to all experts and then dispatches it to  $K$  experts with the highest probability:

$$g(x) = \text{softmax}(\text{Top-K}(x \cdot W_g)), \quad (1)$$

where  $W_g$  is the learnable parameter of  $g(x)$ , the " $x \cdot W_g$ " outputs a vector of length  $N$ , and the Top-K function keeps only  $K$  largest values whose index corresponds to the selected expert.

After the routing network, each input token  $x$  is fed to its selected experts to get the output  $E_i(x)$ . The final output is obtained by calculating the sum of the outputs of the selected experts weighted with probabilities  $g(x)$ .

**All-to-All Communication.** Training and deploying MoE models require distributed computing due to their immense computational and memory demands (Dai et al., 2024; Fedus et al., 2021). For efficiency, both data parallelism and MoE-specific expert parallelism (a specialized form of model parallelism) are utilized (Hwang et al., 2023; Gale et al., 2023; Liu et al., 2024a). Current MoE systems assign experts to separate computing devices (*e.g.*, GPU) in expert parallelism (Fedus et al., 2021; Lepikhin et al., 2020). This necessitates an all-to-all communication to *dispatch* tokens to their respective experts as determined by the routing network (Gale et al., 2023). A second all-to-all communication is then required to *return* (*combine*) the tokens to their original device in data parallelism, completing the forward pass (Gale et al., 2023). Existing frameworks, however, fail to fully exploit redundant tokens, resulting in unnecessary communication costs. Specifically, tokens that are routed to multiple experts hosted on different GPUs have to be redundantly transmitted multiple times, leading to significant inefficiencies. Our approach addresses this by minimizing such redundant token transfers, thereby reducing communication overhead and improving efficiency.

#### 3.2 Expert Profiling from Pre-trained Model

In contrast to previous studies (Zoph et al., 2022; Fedus et al., 2021) that attempt to improve MoE efficiency from the perspective of imbalance issues (Lewis et al., 2021; Clark et al., 2022; Hazimeh et al., 2021; He et al., 2022), in this paper, we return to the primitive goal of MoE design: different experts contain specialized knowledge, and the routing policy dynamically selects experts to process given inputs. From this angle, we notice that the widely adopted load balancing loss (Zoph et al., 2022; Fedus et al., 2021) in popular MoE models is insufficient (Dai et al., 2024; Team, 2024; Zhu et al., 2024), as it only targets evenly distributed expert activation during training. Such a design neglects the combinatorial aspect of the current routing mechanism, where multiple experts are activated simultaneously to collaborate on processing a given input token. This limitation indicates

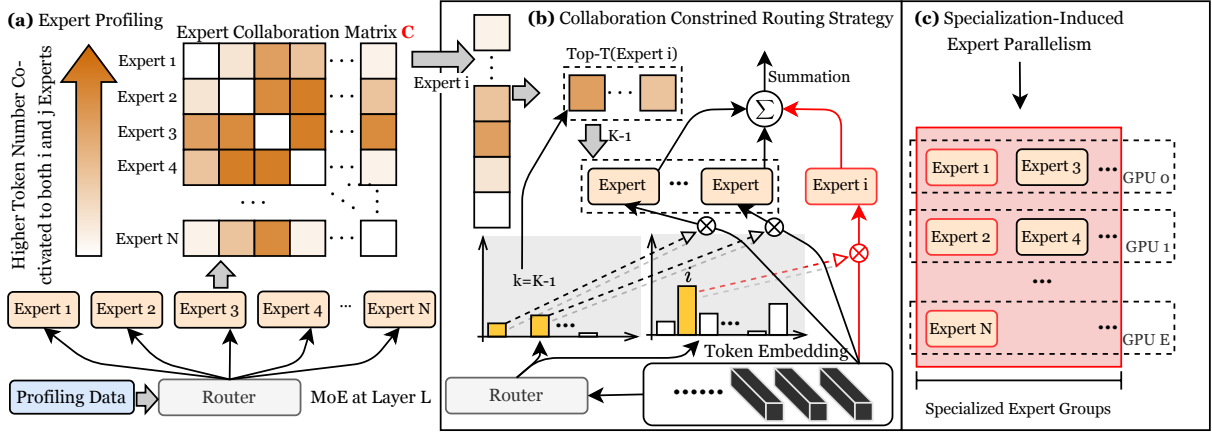


Figure 1: Overview of C2R. (a) shows the process of expert profiling where we obtain the expert collaboration matrix for each layer of the MoE model; (b) describes the mechanism of our C2R strategy. It first selects the top-1 expert for a given token (Expert  $i$  here) and then selects the remaining  $K - 1$  experts from list  $\text{Top-T}(\text{Expert } i)$ ; (c) shows our efficient expert parallelism design.

that their specialization is not enough (Chen et al., 2022; Xue et al., 2022; Shen et al., 2023), and the communication overhead remains unacceptable. In addition, one common intuition is that the specialized knowledge in the training data is not evenly distributed, and pursuing uniformly activated experts contradicts the goal of specialization.

Therefore, based on the aforementioned observations, we aim to improve the specialization in the existing MoE model from a higher-order and novel viewpoint: collaboration and specialization among experts, specifically their co-activation patterns.

**Collaboration among Experts.** Given a batch of input tokens  $\mathcal{B}$ , let  $C_{ij}$  denote the number of times expert  $i$  and expert  $j$  are activated simultaneously, defined as:

$$C_{ij} = \sum_{x \in \mathcal{B}} \mathbb{1}\{g(x)_i \neq 0 \wedge g(x)_j \neq 0\}, \quad (2)$$

where  $g(x)_i$  represents the routing score of expert  $i$  for input  $x$  and  $g(x)_i = 0$  means the expert  $i$  is not selected by token  $x$ . The collaboration matrix  $C$  among experts is illustrated in Figure 1 (a). Note that we compute a collaboration matrix independently for each layer of the model because all input tokens are synchronously forwarded to the last layer. Empirical analysis reveals that inputs with specific patterns are often handled by fixed combinations of experts. Based on this observation, we propose a metric to measure whether an expert tends to be collaborative or specialized. The collaboration degree  $P_i$  of an expert is defined as the entropy of its collaboration frequency distribution

with other experts:

$$P_i = - \sum_{\substack{j=1 \\ i \neq j}}^n p_{ij} \log(p_{ij}), \quad (3)$$

where  $p_{ij} = c_{ij} / \sum_{j=1}^N c_{ij}$  is the collaboration frequency between expert  $i$  and  $j$ . A higher  $P_i$  indicates that expert  $i$  has a more uniform collaboration frequency distribution with other experts, suggesting a greater tendency for collaboration. Conversely, a lower  $P_i$  implies that the expert tends to collaborate with specific experts, indicating a higher degree of specialization. Further, we take the average of the collaboration degree  $P_i$  of all experts within the same layer as the collaboration degree of that layer.

### 3.3 C2R Strategy for Specialized Expert Groups

Our empirical analysis of expert collaboration dynamics in pre-trained Mixture-of-Experts (MoE) models reveals that every expert is prone to collaborate with a specific subset of experts than others. This observation forms the foundation of our proposed strategy, which aims to foster specialized expert groups by constraining the potential combinations of expert collaborations.

**C2R Mechanism** To summarize, our proposed approach restricts expert collaboration combinations by leveraging empirical observations of collaboration patterns, thereby encouraging the formation of more specialized and efficient expert groups. ① To implement the C2R strategy, we start with analyzing expert collaboration patterns in pre-trained

Table 1: Comparison of the performance on **reasoning tasks** and efficiency of the two evaluated network architectures using our C2R, Random-C2R, and conventional top-K routing strategies, respectively. Bold numbers highlight the higher accuracy or speedup ratio between our method and baselines.

Methods	Reasoning Tasks					Speedup (%)	
	WSC	GPQA	LogiQA	PIQA	PROST	EP=2	EP=4
LLaMA-MoE							
Top-K	79.12	<b>24.37</b>	25.04	77.58	25.76	4.0 ↑	3.0 ↑
Random-C2R	78.39	24.15	<b>25.19</b>	76.28	<b>26.18</b>	4.2 ↑	4.8 ↑
C2R	<b>80.22</b>	24.11	<b>25.19</b>	<b>77.86</b>	25.75	<b>4.5 ↑</b>	<b>13.5 ↑</b>
Qwen-MoE							
Top-K	<b>77.66</b>	30.34	30.88	78.84	30.59	15.8 ↑	18.9 ↑
Random-C2R	76.19	28.09	30.26	78.35	29.70	16.2 ↑	21.0 ↑
C2R	76.92	<b>30.41</b>	<b>31.64</b>	<b>78.94</b>	<b>31.05</b>	<b>17.6 ↑</b>	<b>24.9 ↑</b>

models using a representative corpus that simulates realistic token distributions. Specifically, we take tokens from the corpus as input to the MoE model and feed them forward to the final layer. For each layer  $l$  of the model, we first obtain an expert collaboration matrix  $C_l \in \mathbb{N}^{N \times N}$  as described in Section 3.2 where  $N$  is the number of experts in one layer and  $C_l(i, j)$  denote the number of tokens routed to both expert  $i$  and expert  $j$  simultaneously. Then, we sort each row of this matrix and select the top  $T$  indices  $\mathcal{I}_T \in \mathbb{N}^{N \times T}$ , resulting in a list of the  $T$  most frequently collaborating experts for each expert, denoted as:

$$\text{Top-T}(\mathbf{E}_i) = \{e_j \in \mathcal{E} \mid j \in \mathcal{I}_T[i]\} \quad (4)$$

where  $T \in [1, N]$  is the hyperparameter controlling the degree of collaboration, and will be analyzed in detail in Section 4.3. Building upon this analysis, we propose the C2R strategy. For each token, instead of directly routing it to the corresponding top-K experts, we first select its top-1 expert  $\mathbf{E}_i$ . Then we restrict the selection of the remaining  $K - 1$  experts to the list  $\text{Top-T}(\mathbf{E}_i)$  identified as having the most frequent collaborations with expert  $\mathbf{E}_i$ . Note that the selection is still based on routing scores but is constrained to the list  $\text{Top-T}(\mathbf{E}_i)$ . This approach substantially reduces the potential combinations of experts involved in the routing process, fostering more specialized expert groups while preserving the flexibility needed for dynamic routing.

### 3.4 Specialization-Induced Zero-Redundancy All-to-All

MoE layers often underutilize GPUs due to sequential all-to-all communications and feed-forward layers for token dispatching and combining. The all-to-all communication typically

Table 2: Comparison of the performance on **NLU tasks** and efficiency of the two evaluated network architectures using our C2R, Random-C2R, and conventional top-K routing strategies, respectively. Bold numbers highlight the higher accuracy or speedup ratio between our method and baselines.

Methods	NLU Tasks					Speedup (%)	
	RACE	SciQ	RTE	BoolQ	COPA	EP=2	EP=4
LLaMA-MoE							
Top-K	39.52	88.90	49.10	72.63	84.00	4.0 ↑	3.0 ↑
Random-C2R	39.52	89.20	50.18	70.92	80.00	4.2 ↑	4.8 ↑
C2R	<b>39.90</b>	<b>89.60</b>	<b>50.54</b>	<b>72.91</b>	<b>85.00</b>	<b>4.5 ↑</b>	<b>13.5 ↑</b>
Qwen-MoE							
Top-K	<b>39.14</b>	<b>94.70</b>	72.92	80.03	80.00	15.8 ↑	18.9 ↑
Random-C2R	38.66	94.60	67.87	75.17	<b>84.00</b>	16.2 ↑	21.0 ↑
C2R	<b>39.14</b>	<b>94.70</b>	<b>73.29</b>	<b>80.24</b>	82.00	<b>17.6 ↑</b>	<b>24.9 ↑</b>

consumes over 30% of runtime (Hwang et al., 2023), with this proportion increasing as the number of GPUs grows, leading to inefficient MoE expert parallelism.

Our proposed C2R routing strategy offers additional opportunities to optimize communication overhead. By recognizing that certain expert collaborations occur more frequently, we can co-locate closely collaborating experts on the same computational units (e.g., GPUs). For tokens routed to multiple experts residing on the same device, we employ a novel design that sends only a single copy of the token to the shared device, where it is subsequently replicated locally for processing by each assigned expert. This approach significantly reduces inter-device communication, thereby alleviating all-to-all communication bottlenecks. As expert collaboration patterns stabilize through specialization, this optimized strategy can yield substantial reductions in communication costs, ultimately improving the model’s overall efficiency.

## 4 Experiments

### 4.1 Implementation Details

**Network Architecture and Baselines.** We select two representative open-source MoE models, including LLaMA-MoE (Zhu et al., 2024) and Qwen-MoE (Team, 2024). LLaMA-MoE has 32 transformer layers, hidden size 4096, 32 attention heads, and 8 experts per MoE layer with top-2 routing. Qwen1.5-MoE has 24 layers, hidden size 2048, 16 attention heads, and 60 experts per MoE layer with top-4 routing, plus a shared expert. We implement our method by replacing the original top-K routing with our C2R strategy in both models, leaving Qwen-MoE’s shared experts unchanged. For baselines, we use model’s original routing policy as our

baseline, and replace the expert profiling process of C2R with random initialization as another baseline. Each expert in LLaMA-MoE is an MLP with an intermediate dimension of 1376 and input/output dimension of 4096, while in Qwen1.5-MoE, experts have an intermediate dimension of 1408 and input/output dimension of 2048, with the shared expert’s intermediate dimension being 5632.

**Datasets and Benchmarks.** We conduct supervised fine-tuning of the two selected models under the original routing policy and ours, respectively. For LLaMA-MoE, we follow the script provided by its official repository to fine-tune on the Deita-6K dataset (Liu et al., 2024b). For Qwen-MoE, we choose the popular LIMA instruction tuning dataset (Zhou et al., 2024). We examine the superior performance of our proposed routing strategy on popular benchmarks and potential inference speedup. Specifically, 10 benchmarks across two types of downstream tasks are examined in this paper, including natural language understanding (RACE (Lai et al., 2017), SciQ (Johannes Welbl, 2017), RTE (Bentivogli et al., 2009), BoolQ (Clark et al., 2019), COPA (Roemmele et al., 2011)) and reasoning (WSC (Levesque et al., 2012), GPQA (Rein et al., 2023), LogiQA (Liu et al., 2020), PIQA (Bisk et al., 2020), PROST (Aroca-Ouellette et al., 2021)). We use WikiText (Merity et al., 2016) for expert parallelism profiling in efficiency evaluation.

**Training Configuration.** For LLaMA-MoE, we follow the official training settings to do full-parameter SFT, using AdamW optimizer (Yao et al., 2021) and the learning rate is set to  $2 \times 10^{-5}$  with a warm-up ratio of 0.03 and cosine scheduler. We use a total batch size of 16 with gradient accumulation steps as 8. The max length of the input sequence is set to 2048. We train the model for 2 epochs without freezing the gate. For Qwen-MoE, we basically follow the above settings, except for using: (1) a warm-up step of 10 instead of a warm-up ratio of 0.03, (2) a total batch size of 32 with gradient accumulation steps set to 2, and ZeRO-3 Offload to avoid out-of-memory (OOM) issues. We set hyperparameters T as 5 and 30 for LLaMA-MoE and Qwen-MoE, respectively.

**Expert Parallelism Speedup Estimation.** To quantify the potential speedup of our zero-redundancy all-to-all in C2R, we conducted a comprehensive analysis using a calibration dataset

of 64 random 2048-token segments from WikiText. Our evaluation process comprised three key steps: First, we implemented expert parallelism based on the MegaBlocks framework (Gale et al., 2023) and used PyTorch Profiler<sup>1</sup> to measure the wall-clock time proportion  $P^{EP}$  of all-to-all communication relative to total inference time at various expert parallelism degrees EP. Second, we calculated the token redundancy  $r^{EP}$  for each GPU during all-to-all communication across different EP values. Finally, we derived the estimated speedup at each EP by computing  $P^{EP} \times r^{EP}$ . This methodology allowed us to quantify the efficiency gains of our approach, considering both communication overhead and potential reductions in data transfer across various scales of expert parallelism. We evaluate the speedup across EP in  $\{2, 4\}$ .

## 4.2 Superior Performance of Our Method

We select LLaMA-MoE and Qwen-MoE models to train on Deita-6K and LIMA datasets, respectively, and the evaluation results are summarized in Table 1 and Table 2. The following observations can be drawn: (1) Our C2R strategy demonstrates superior performance compared to the baseline routing strategies. Specifically, LLaMA-MoE with our approach achieves an average performance improvement of 0.26% on the reasoning tasks and 0.76% on the natural language understanding tasks, respectively, compared to the top-K baseline. Similarly, Qwen-MoE demonstrates improvements of 0.13% and 0.52% on these tasks, respectively. This validates the effectiveness of our proposed method. (2) Our method shows consistent performance benefits on both MoE Architectures. Specifically, we obtain a total average performance improvement of 0.51% on LLaMA-MoE and 0.33% on Qwen-MoE, respectively, across all datasets. This verifies the generalization of our proposed method. (3) Our speedup results demonstrate the significant efficiency gains of our C2R framework. At  $EP = 4$ , Llama-MoE with C2R achieves around 10% more speedup ratio compared to the baselines. This improvement highlights the effectiveness of our approach in optimizing communication in all-to-all and reducing overhead in MoE, potentially enabling more efficient scaling.

<sup>1</sup><https://pytorch.org/docs/stable/profiler.html>

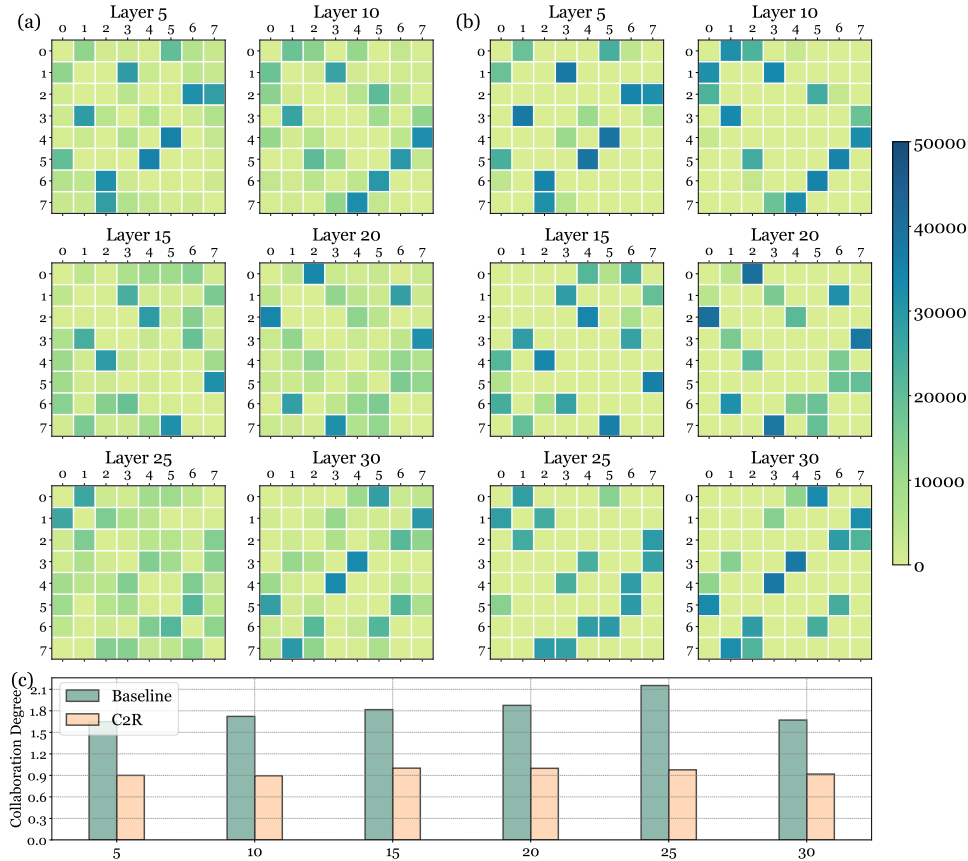


Figure 2: Visualization of expert collaboration matrix in several intermediate layers of LLaMA-MoE after SFT. (a): Results with conventional top-K routing strategy. (b): Results with our C2R strategy ( $T = 2$ ). (c): The average collaboration degree comparison between Baseline and our C2R strategy. A darker pixel in (a) and (b) indicates a higher number of tokens routed simultaneously to the corresponding experts (indexed by row and column) within the given layer, which means these two experts collaborate more frequently. Note that many pixels in (b) have a value of 0, meaning that the corresponding two experts will never be selected simultaneously, while most of the pixels in (a) have a light color indicating a non-0 value. (c) demonstrates that experts in our model exhibit a higher degree of specialization.

### 4.3 Expert Collaboration Analysis on LLaMA-MoE

In this part, we take LLaMA-MoE as an example to conduct an in-depth analysis of expert collaboration under our C2R strategy compared to the conventional top-K routing strategy. Specifically, we randomly sampled a total of 200 context sentences from different domains of the LongBench dataset (Bai et al., 2023), truncating each sentence to 1024 tokens. Overall, we used these 0.2M tokens to simulate token distribution in real-world scenarios and fed them into the model to derive the expert collaboration matrices  $C_l$  for each layer  $l$ . We visualized these matrices using heatmaps as illustrated in Figure 2 (a) and (b). Here, we show the results of 6 intermediate layers selected at an interval of five layers for both settings after fine-tuning. Figure 2 (c) shows the calculated values of collaboration degree introduced in Section 3.2. From analysis, our findings are as follows: (1) Under the conventional top-K routing strategy, as shown in Figure 2 (a), the space of expert routing combina-

tions is considerably large. While the collaborative tendency among experts is not uniform, where each expert tends to be co-activated more with certain experts than with others, every expert has the opportunity to collaborate with any other expert. This results in significant communication overhead when implementing expert parallelism, as many tokens tend to activate experts distributed across different accelerators. (2) In contrast, under our C2R strategy, as shown in Figure 2 (b), each expert is predominantly co-activated with a small group of specific experts. This greatly reduces the routing space and thus contributes to specialized expert groups, which is also supported by the calculated collaboration degree shown in Figure 2 (c). With this property, we can easily distribute experts from the same group on the same accelerator, thereby reducing communication costs.

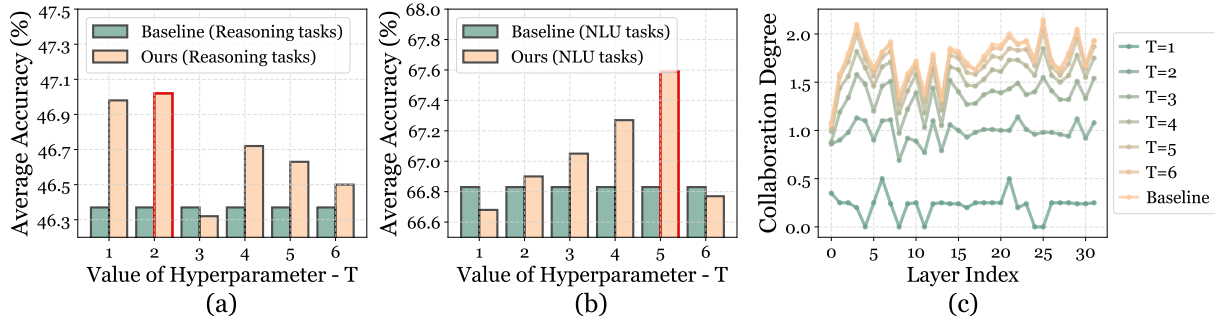


Figure 3: Performance and collaboration degree comparison of LLaMA-MoE. (a) and (b) respectively show the performance comparison between our C2R strategy (*Ours*) and conventional top-K routing strategy (*Baseline*) on two downstream tasks, namely Reasoning tasks and NLU tasks, with hyperparameter  $T$  varying from 1 to 6. (c) shows the collaboration degree comparison between *Baseline* and *Ours* under different values of hyperparameter  $T$  in different layers of the model. Note that since the LLaMA-MoE model we use is to select 2 out of 8 experts per layer, our method degenerates to a conventional top-K routing strategy (i.e., the baseline) when  $T = 7$ , so we omit this case.

#### 4.4 Pareto Optimal Balance between Collaboration and Specialization

In this section, we aim to answer the research question of "how much collaboration is needed." To validate our claim that there exists a Pareto optimal balance between collaboration and specialization, we vary the hyperparameter  $T$  from 1 to 6 and examine the average performance on the aforementioned benchmarks as well as the collaboration degree in each layer of the model. A smaller  $T$  indicates that each expert collaborates with only a few specific experts, leading to greater specialization, whereas a larger  $T$  implies more collaboration. This is demonstrated experimentally in Figure 3 (c). As shown in Figure 3 (a) and (b), we observe that as  $T$  increases, the trend of model performance initially improves and then declines. This observation supports the proposition that it is possible to find a Pareto optimal point where model performance is maximized. We also notice that there is a minimum value of model accuracy in reasoning tasks when  $T = 3$ , which we treat as noise that does not affect the overall trend. Taking communication overhead into account, the value of  $T$  effectively controls the trade-off between model performance and efficiency. Specifically, in some cases, we might accept a slight performance drop in exchange for more specialized expert groups (i.e., a smaller  $T$ ), allowing each accelerator to host an entire expert group and thereby reducing communication costs.

#### 4.5 Study on the Expert Parallelism Degree

There is a trade-off between the expert parallelism degree EP and inference speed in C2R. As the expert parallelism degree EP increases: (1) the vanilla all-to-all communication (i.e. w/o zero-redundancy design) cost tends to increase due to

more GPUs needed; (2) the redundancy tends to decrease due to the fewer experts on each GPU. Therefore, the final inference speed with zero-redundancy all-to-all needs further investigation. Therefore, we evaluate the efficiency performance on Qwen-MoE across EP in  $\{2,3,4,5,6\}$ , as shown in Table 3. The highest potential speedup rate can be achieved at  $EP = 5$ , balancing the two opposing trends when equipped with zero-redundancy all-to-all.

Table 3: Efficiency performance analysis of Qwen-MoE model across expert parallelism (EP) dimensions from 2 to 6. As EP increases, communication redundancy decreases, but total all-to-all time rises. The highest potential speedup rate can be achieved at  $EP = 5$ , balancing these opposing trends.

EP	Redundancy	All-to-All Time	Speedup
2	<b>58.3%</b>	30.1%	17.6%
3	47.6%	40.7%	19.4%
4	40.2%	61.9%	24.9%
5	38.4%	76.3%	<b>29.3%</b>
6	32.9%	<b>77.2%</b>	25.4%

## 5 Conclusion

In this paper, we present a brand new perspective for analyzing MoE routing behavior, namely expert collaboration and specialization. We design a novel collaboration-constrained routing (C2R) strategy to improve expert utilization, which further delivers a property of specialized expert groups. Based on such characteristics, we propose an efficient expert parallelism design to reduce communication overhead at runtime. Extensive experiments on two representative MoE models across multiple downstream benchmarks exhibit a consistent performance improvement, demonstrating the ef-



fectiveness of our approach. Additional runtime analysis shows significant reductions in total running time savings, underscoring our design as a promising direction for addressing both training and inference efficiency challenges.

## 6 Limitations

Our study reveals an intriguing yet unexplored observation (Figure 3): different downstream tasks (*i.e.*, Reasoning and NLU) require varying degrees of expert collaboration for optimal performance. This finding suggests two promising directions for future research: (1) developing task-specific routing strategies that achieve a Pareto-optimal balance between collaboration and specialization, potentially yielding significant improvements in task-specific performance; and (2) implementing our zero-redundancy all-to-all approach efficiently on real-world GPU architectures. These avenues for future work address both the theoretical foundations and practical implementations of MoE models, potentially leading to more efficient and task-adaptive LLMs.

## Acknowledgment

Pingzhi Li and Tianlong Chen are supported by NIH OT2OD038045-01 and the UNC SDSS Seed Grant.

## References

- Alhabib Abbas and Yiannis Andreopoulos. 2020. Biased mixtures of experts: Enabling computer vision inference under data transfer limitations. *IEEE Transactions on Image Processing*, 29:7656–7667.
- Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani. 2016. Network of experts for large-scale image categorization. In *European Conference on Computer Vision*, pages 516–532. Springer.
- Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. 2021. Prost: Physical reasoning of objects through space and time. *arXiv preprint arXiv:2106.03634*.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. **Longbench: A bilingual, multitask benchmark for long context understanding**. Preprint, arXiv:2308.14508.
- Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.
- Ke Chen, Lei Xu, and Huisheng Chi. 1999. Improved learning algorithms for mixture of experts in multiclass classification. *Neural networks*, 12(9):1229–1252.
- Tianyu Chen, Shaohan Huang, Yuan Xie, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. 2022. Task-specific expert pruning for sparse mixture-of-experts. *arXiv preprint arXiv:2206.00277*.
- Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. 2022. Unified scaling laws for routed language models. *arXiv preprint arXiv:2202.01169*.
- 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 *NAACL*.
- Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. 2024. Deepseek-moe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.
- David Eigen, Marc’Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*.
- William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.
- Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. 2023. Megablocks: Efficient sparse training with mixture-of-experts. *Proceedings of Machine Learning and Systems*, 5:288–304.
- Sam Gross, Marc’Aurelio Ranzato, and Arthur Szlam. 2017. Hard mixtures of experts for large scale weakly supervised vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6865–6873.
- Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. 2021. Dselectk: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34.
- Jiaao He, Jidong Zhai, Tiago Antunes, Haojie Wang, Fuwen Luo, Shangfeng Shi, and Qin Li. 2022. Faster-moe: modeling and optimizing training of large-scale dynamic pre-trained models. In *Proceedings of the*

- 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pages 120–134.
- Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, et al. 2023. Tutel: Adaptive mixture-of-experts at scale. Proceedings of Machine Learning and Systems, 5:269–287.
- Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. Neural computation, 3(1):79–87.
- Hao Jiang, Ke Zhan, Jianwei Qu, Yongkang Wu, Zhaoye Fei, Xinyu Zhang, Lei Chen, Zhicheng Dou, Xipeng Qiu, Zikai Guo, et al. 2021. Towards more effective and economic sparsely-activated model. arXiv preprint arXiv:2110.07431.
- Matt Gardner Johannes Welbl, Nelson F. Liu. 2017. Crowdsourcing multiple choice science questions.
- Michael I Jordan and Robert A Jacobs. 1994. Hierarchical mixtures of experts and the em algorithm. Neural computation, 6(2):181–214.
- 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. arXiv preprint arXiv:2001.08361.
- Kenichi Kumatani, Robert Gmyr, Felipe Cruz Salinas, Linqun Liu, Wei Zuo, Devang Patel, Eric Sun, and Yu Shi. 2021. Building a great multi-lingual teacher with sparsely-gated mixture of experts for speech recognition. arXiv preprint arXiv:2112.05820.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. **RACE: Large-scale ReAding comprehension dataset from examinations**. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.
- 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.
- Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In Thirteenth international conference on the principles of knowledge representation and reasoning.
- Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. Base layers: Simplifying training of large, sparse models. In International Conference on Machine Learning, pages 6265–6274. PMLR.
- Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. arXiv preprint arXiv:2405.04434.
- Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. arXiv preprint arXiv:2007.08124.
- Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024b. **What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning**. In The Twelfth International Conference on Learning Representations.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. **Pointer sentinel mixture models**. Preprint, arXiv:1609.07843.
- Xiaonan Nie, Pinxue Zhao, Xupeng Miao, Tong Zhao, and Bin Cui. 2022. Hetumoe: An efficient trillion-scale mixture-of-expert distributed training system. arXiv preprint arXiv:2203.14685.
- Bowen Pan, Yikang Shen, Haokun Liu, Mayank Mishra, Gaoyuan Zhang, Aude Oliva, Colin Raffel, and Rameswar Panda. 2024. Dense training, sparse inference: Rethinking training of mixture-of-experts language models. arXiv preprint arXiv:2404.05567.
- Svetlana Pavlitskaya, Christian Hubschneider, Michael Weber, Ruby Moritz, Fabian Huger, Peter Schlicht, and Marius Zollner. 2020. Using mixture of expert models to gain insights into semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 342–343.
- 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.
- Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In 2011 AAAI spring symposium series.
- Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, et al. 2021. Hash layers for large sparse models. Advances in Neural Information Processing Systems, 34:17555–17566.

- 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](#).
- Yikang Shen, Zheyu Zhang, Tianyou Cao, Shawn Tan, Zhenfang Chen, and Chuang Gan. 2023. Moduleformer: Learning modular large language models from uncurated data. [arXiv preprint arXiv:2306.04640](#).
- Qwen Team. 2024. [Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters](#)".
- Xin Wang, Fisher Yu, Lisa Dunlap, Yi-An Ma, Ruth Wang, Azalia Mirhoseini, Trevor Darrell, and Joseph E Gonzalez. 2020. Deep mixture of experts via shallow embedding. In *Uncertainty in artificial intelligence*, pages 552–562. PMLR.
- Fuzhao Xue, Xiaoxin He, Xiaozhe Ren, Yuxuan Lou, and Yang You. 2022. One student knows all experts know: From sparse to dense. [arXiv preprint arXiv:2201.10890](#).
- Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. 2019. Condconv: Conditionally parameterized convolutions for efficient inference. [Advances in Neural Information Processing Systems](#), 32.
- Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. 2021. Adahessian: An adaptive second order optimizer for machine learning. In [proceedings of the AAAI conference on artificial intelligence](#), volume 35, pages 10665–10673.
- Dianhai Yu, Liang Shen, Hongxiang Hao, Weibao Gong, Huachao Wu, Jiang Bian, Lirong Dai, and Haoyi Xiong. 2024. Moesys: A distributed and efficient mixture-of-experts training and inference system for internet services. [IEEE Transactions on Services Computing](#).
- Seniha Esen Yuksel, Joseph N. Wilson, and Paul D. Gader. 2012. [Twenty years of mixture of experts](#). [IEEE Transactions on Neural Networks and Learning Systems](#), 23(8):1177–1193.
- Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Moefication: Conditional computation of transformer models for efficient inference. [arXiv preprint arXiv:2110.01786](#).
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. [Advances in Neural Information Processing Systems](#), 36.
- Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. 2022. Mixture-of-experts with expert choice routing. [arXiv preprint arXiv:2202.09368](#).
- Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. 2024. Llama-moe: Building mixture-of-experts from llama with continual pre-training. [arXiv preprint arXiv:2406.16554](#).
- 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](#).
- Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Jianfeng Gao, and Tuo Zhao. 2022. [Taming sparsely activated transformer with stochastic experts](#). In [International Conference on Learning Representations](#).