Maximum Score Routing For Mixture-of-Experts

Bowen Dong[1], **Yilong Fan**[2], **Yutao Sun**[1], **Zhenyu Li**[1], **Tengyu Pan**[1], **Xun Zhou**[3†], **Jianyong Wang**[1†]

[1]Tsinghua University, [2]Tianjin University,
[3]Seed-Foundation-Model Team, ByteDance
dbw22@mails.tsinghua.edu.cn

Abstract

Routing networks in sparsely activated mixture-of-experts (MoE) dynamically allocate input tokens to top-k experts through differentiable sparse transformations, enabling scalable model capacity while preserving computational efficiency. Traditional MoE networks impose an expert capacity constraint to ensure GPU-friendly computation. However, this leads to token dropping when capacity is saturated and results in low hardware efficiency due to padding in underutilized experts. Removing the capacity constraint, in turn, compromises load balancing and computational efficiency. To address these issues, we propose Maximum Score Routing (**MaxScore**), a novel MoE routing paradigm that models routing as a minimum-cost maximum-flow problem and integrates a $\mathrm{SoftTopk}$ operator. MaxScore resolves the fundamental limitations of iterative rerouting and optimal transport formulations, achieving lower training losses and higher evaluation scores at equivalent FLOPs compared to both constrained and unconstrained baselines. Implementation details and experimental configurations can be obtained from `https://github.com/dongbw18/MaxScore.git`.

1 INTRODUCTION

The Mixture of Experts (MoE) paradigm has emerged as a compelling architectural strategy for scaling neural networks while maintaining computational efficiency. This approach dynamically combines multiple subsets of parameters (experts) by a learnable routing network, aiming to improve model capacity and computational efficiency. The routing network of sparsely activated MoE (Shazeer et al., 2017) dynamically allocates input tokens to top-k experts through differentiable sparse transformations, enabling conditional computation that scales model parameters without proportionally increasing FLOPs.

Softmax is conventionally employed to compute token-expert affinity coefficients in MoE routing networks, which promotes inter-expert competition. To mitigate winner-takes-all and preserve load balance, both hard constraints using expert capacity (Eigen et al., 2014), and soft constraints using auxiliary losses (Bengio et al., 2016), are incorporated into the routing network (Shazeer et al., 2017). GShard (Lepikhin et al., 2020) pioneers the integration of MoE with Transformer architectures (Vaswani et al., 2017), where expert capacity constraints enable GPU-friendly computation patterns. ExpertChoice (Zhou et al., 2022) directly enables experts to select tokens based on capacity constraints. However, token dropping occurs when inputs are routed to capacity-saturated experts, while padding operations in underutilized experts create hardware inefficiencies. Empirical analysis reveals that approaches such as expanding capacity (Hwang et al., 2023) or removing capacity constraints altogether (Gale et al., 2022; Muennighoff et al., 2024) effectively eliminate token dropping, but inevitably introduce a trade-off between computational efficiency and load balancing performance. Efforts to prevent token dropping via refined routing strategies (Fedus et al., 2022; Clark et al., 2022) have not yielded performance improvements, highlighting unresolved challenges in dynamic resource allocation.

This work introduces **Maximum Score Routing (MaxScore)**, a novel MoE routing paradigm that formulates token-expert routing as a minimum-cost maximum-flow problem (Waissi, 1994), integrated with a $\mathrm{SoftTopk}$ operator. To the best of our knowledge, this is the first successful integration of network flow modeling and $\mathrm{SoftTopk}$ in MoE routing.

MaxScore preserves GPU-compatible expert capacity constraints and achieves better load balanc-

[†] indicates corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2025, pages 12619–12632
July 27 - August 1, 2025

ing. Under the same FLOPs, MaxScore exhibits lower training loss and higher evaluation scores compared to both constrained and unconstrained baselines. Ablation studies demonstrate the necessity of both network flow modeling and the $\mathrm{SoftTopk}$ operator, revealing fundamental limitations in the iterative rerouting mechanism of Fedus et al. (2022) and the optimal transport-based routing of Clark et al. (2022). The synergistic combination of two methodological enhancements yields superadditive performance gains, with empirical results demonstrating that their integrated efficacy surpasses the linear summation of individual improvements. Scaling experiments show that MaxScore delivers consistent performance improvements with larger activated parameter budgets, and achieves more gains when increasing the number of experts, compared with standard MoE approaches.

2 PRELIMINARIES

2.1 Top-k Sparsely Activated MoE

The top-k routing mechanism is a cornerstone of sparsely activated MoE architectures, enabling efficient scaling of model capacity while maintaining computational tractability. Originally popularized in language modeling (Shazeer et al., 2017), this paradigm dynamically routes each input token to a subset of k expert networks (where $k \ll e$, for e total experts). Unlike dense models that activate all parameters per input, top-k routing induces conditional computation by selecting experts based on learned gating scores, typically computed via softmax over a trainable projection of input embeddings (Lepikhin et al., 2020).

For a given input x, the output y of the MoE module can be written as follows:

$$y = \sum_{i=1}^{E} R(x)_i \cdot E_i(x), \tag{1}$$

$$R(x) = \mathrm{KeepTopk}(\mathrm{Softmax}(x \cdot W_g)), \tag{2}$$

where $R(x)$ is the sparsely activated routing function, $\mathrm{KeepTopk}(\cdot)$ retains the top-k largest values while setting others to zero, W_g is the weight matrix of the routing function, $E_i(x)$ is the output of the i-th expert network and the computation is performed only when $R(x)_i > 0$.

By leveraging sparse activation, MoE decouples total capacity $\mathcal{O}(e)$ from per-step computational cost, activating only $\mathcal{O}(k)$ parameters during both training and inference.

2.2 Operators in Top-k MoE Routing

Routings in MoE commonly use $\mathrm{Softmax}(\cdot)$ to calculate the token-expert affinity coefficients, which encourages competition between experts. However, $\mathrm{Softmax}(\cdot)$ serves as a smooth approximation to the one-hot $\mathrm{Argmax}(\cdot)$ function, which can lead to inefficiencies in top-k routing, as the top-1 expert often receives a disproportionately large affinity score compared to the remaining $k-1$ experts.

Alternative routing operators have also been investigated. DeepSeek-AI et al. (2024b) replaces $\mathrm{Softmax}(\cdot)$ with $\mathrm{Sigmoid}(\cdot)$ to align with its auxiliary-loss-free load balancing strategy, while ReMoE (Wang et al., 2025) explores the feasibility of using $\mathrm{ReLU}(\cdot)$ for routing decisions.

We define $\mathrm{SoftTopk}(\cdot)$ as a smooth approximation to $\mathrm{ArgTopk}(\cdot)$, which represents the top-k selection in a one-hot form, formally given by:

$$\mathrm{ArgTopk}(\mathbf{a})_i = \begin{cases} 1, a_i \in \mathrm{Topk}(\mathbf{a}) \\ 0, \mathrm{otherwise}, \end{cases} \tag{3}$$

where $\mathbf{a} = (a_1, a_2, ..., a_e)$ represents the affinity coefficients between the token and e experts.

Martins and Astudillo (2016) and Peters et al. (2019) proposed $\mathrm{Sparsemax}(\cdot)$ and $\mathrm{Entmax}(\cdot)$ as differentiable approximations for top-k probability truncation. Su (2024) further introduced a broader family of $\mathrm{SoftTopk}(\cdot)$ operators. However, their integration into MoE routing has not been investigated, leaving a promising direction underexplored.

2.3 Expert Capacity Constrained

To counteract the winner-takes-all phenomenon and maintain load balancing in the routing network, traditional routing architectures integrate dual constraint mechanisms: (i) hard limits through expert capacity (Eigen et al., 2014), and (ii) soft regularization via differentiable auxiliary losses (Bengio et al., 2016; Shazeer et al., 2017; Zoph et al., 2022).

GShard (Lepikhin et al., 2020) strategically harmonizes capacity-constrained MoE design with Transformer architectures (Vaswani et al., 2017). For a batch of n tokens, GShard fixes per-expert capacity with $c = \frac{k*n}{e}$ to enable parallel-friendly computation patterns. This routing mechanism, however, poses optimization challenges due to imbalanced expert utilization. While underloaded experts incur computational overhead through padding (mathematically sound but hardware-inefficient), overloaded experts lead to token dropping. Increasing expert capacity $c' = c_f * \frac{k*n}{e}$ by a

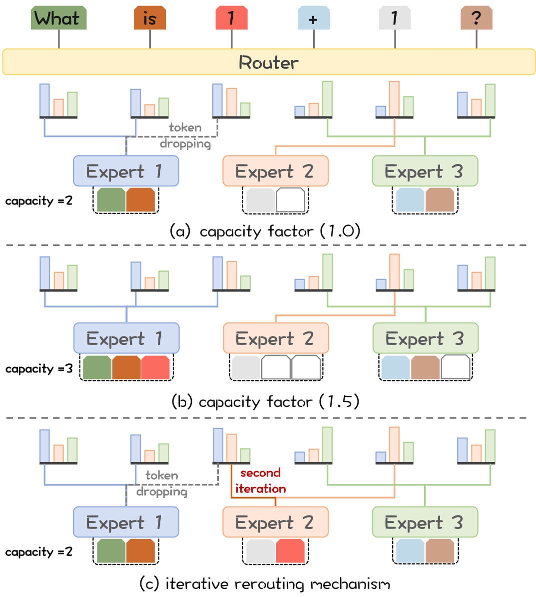

Figure 1: Different top-2 routing paradigms for 3 experts and 6 tokens. (a) sets capacity-factor $c_f = 1.0$, and token dropping occurs; (b) sets capacity-factor $c_f = 1.5$, there is no more token dropping, but more computation is wasted; (c) uses iterative rerouting mechanism, the dropped token is reassigned to expert with remaining capacity.

capacity-factor c_f can alleviate token dropping. Tutel (Hwang et al., 2023) uses a highly scalable stack design and sets the c_f dynamically, but it would lead to additional computational costs and reduced load balancing. Figure 1(a) and 1(b) shows the trade-off between token dropping and additional computation by increasing expert capacity. Figure 2(a) shows the token dropping proportion in the MoE routing of each layer in a GShard model with $e = 16$ and $k = 2$, and approximately 35% of tokens routed to the second experts experience dropping.

ExpertChoice (Zhou et al., 2022) inverts the conventional routing paradigm by allowing experts to select their top-c tokens, thereby achieving optimal load balancing. However, this strategy allows each token to be assigned to an arbitrary number of experts, including zero, which exacerbates token dropping. More importantly, it introduces a data leakage issue: determining whether a token belongs to the top-c set of a given expert requires comparisons not only with preceding tokens but also with subsequent ones, thereby violating the causal structure required by autoregressive models.

Another class of approaches, referred to as DropLess MoE, eliminates capacity constraints entirely to prevent token dropping. Those methods allocate an indefinite number of tokens to experts via direct indexing (e.g., DeepSeekMoE (Dai et al., 2024; DeepSeek-AI et al., 2024a,b), OLMoE (Muennighoff et al., 2024; Gale et al., 2022)).

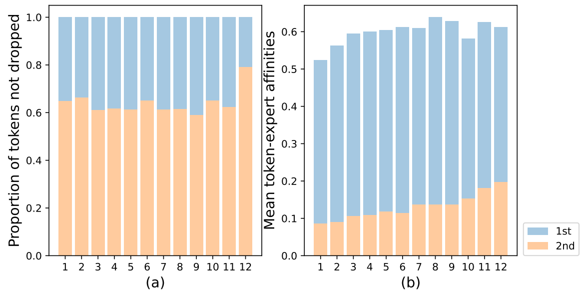

Figure 2: The proportion of tokens not dropped and the mean token-expert affinities in top-2 routing are analyzed separately. The data is derived from the GShard MoE with $e = 16$ after training on 65 billion tokens. (a) shows that tokens assigned to the top-1 experts are rarely dropped, whereas approximately 35% of tokens routed to the second experts experience dropping. (b) illustrates that the top-1 token-expert affinities are typically much higher than those of other experts.

Switch Transformers (Fedus et al., 2022) explored an iterative rerouting mechanism for dropped tokens as shown in Figure 1(c): in the first stage, tokens are assigned to experts using the top-k strategy; in the second stage, any dropped tokens are greedily reassigned to the highest-affinity expert among those with remaining capacity. However, empirical results show that this approach does not lead to improvements in model quality.

SBASE (Clark et al., 2022) formulates MoE routing as an optimal transport problem: $\mathbf{c} = (c_1, c_2, ..., c_e)$ denotes the capacity of each expert, and $\mathbf{k} = (k_1, k_2, ..., k_n)$ specifies the number of experts each token should be assigned to. The matrix $\mathbf{A} \in \mathbb{R}^{n \times e}$ represents token-expert affinity coefficients. The feasible solution space is defined as

$$U(\mathbf{c}, \mathbf{k}) = \{\mathbf{P} \in \mathbb{R}_{\geq 0}^{n \times e} | \mathbf{P}^T \mathbf{1}_n = \mathbf{c}, \mathbf{P}\mathbf{1}_e = \mathbf{k}\}, \tag{4}$$

and the optimization objective is

$$d_{\mathbf{A}}(\mathbf{c}, \mathbf{k}) = \max_{\mathbf{P} \in U(\mathbf{c}, \mathbf{k})} \sum_{ij} \mathbf{P}_{ij} \mathbf{A}_{ij}. \tag{5}$$

To efficiently approximate the solution, SBASE employs the parallelizable Sinkhorn algorithm (Cuturi, 2013). Nonetheless, this formulation primarily contributes to improved training stability, offering limited gains beyond this benefit.

3 METHODOLOGY

We investigate the fundamental reasons why the iterative rerouting mechanism (**Iter**) and the optimal transport formulation (**Sinkhorn**) fail to improve model quality, and propose Maximum Score Routing (**MaxScore**), a novel mixture-of-experts routing strategy that integrates network flow modeling and a differentiable $\mathrm{SoftTopk}(\cdot)$ operator.

3.1 Limitations of Iter and Sinkhorn

Softmax operator. Both the iterative rerouting mechanism and the optimal transport formulation aim to achieve a globally improved allocation by replacing locally optimal assignment strategies. However, as discussed in Section 2.2, using the conventional $\mathrm{Softmax}(\cdot)$ to compute token-expert affinity scores results in the top-1 affinity being significantly higher than those of other token-expert pairs. We statistically analyze the probability distribution in a top-2 GShard MoE, as shown in Figure 2(b), where the top-1 token-expert affinities markedly exceeds that of the second-ranked expert. For example, if a token's top-2 affinities are 0.8 and 0.05 respectively, then when the first expert is saturated, substituting with any expert outside the top-2 (with affinity below 0.05) yields no meaningful benefit; similarly, if the second expert is saturated, replacing it has negligible impact on the model's gradient.
Limitation of optimal transport formulation. Modeling MoE routing using Equations (4) and (5) has inherent limitations: in MoE routing strategies, the actual gain of a token-expert pair appearing multiple times is equivalent to that of a single occurrence. This constraint cannot be enforced in the optimal transport formulation. As illustrated in Figure 3, high-probability token-expert pairs may be matched repeatedly, causing redundant reward accumulation and effectively degenerating to a top-1 routing scheme, which results in wasted computational resources.

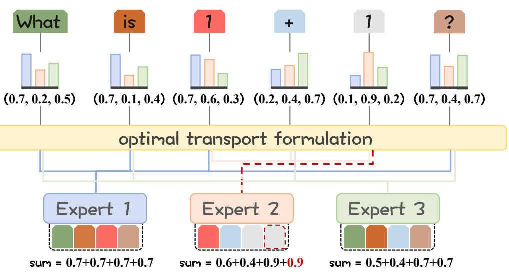

Figure 3: Limitation of optimal transport formulation. The fifth token and the second expert matched twice.

3.2 Maximum Score Routing

SoftTopk operator. We first tried different operators as shown in Table 1, but due to the potential damage caused by the increased computational complexity, we did not achieve better results than $\mathrm{Softmax}(\cdot)$. We propose a simple but highly effective $\mathrm{SoftTopk}(\cdot)$ operator for MoE routing:

Name	Expression
$\mathrm{Softmax}(x)$	$y = e^x / \sum_j^N e^{x_j}$
$\mathrm{Sigmoid}(x)$	$y = 1/(1+e^{-x})$
$\mathrm{SoftKmax}(x)^{(k)}$	$y^{(k)} = y^{(k-1)} + \mathrm{Softmax}(g^{(k-1)})$ $g^{(k-1)} = (1 - y^{(k-1)}) \otimes x$
$\mathrm{IterTopk}(x)^{(k)}$	$y^{(k)} = y^{(k-1)} + g(x; 1 - y^{(k-1)})$ $g(x; w) = w \cdot e^x / \sum_j^N w_j \cdot e^{x_j}$
$\mathrm{GradTopk}(x)^{(k)}$	$y^{(k)} = e^{g^{(k)} - z^{(k)}}$ $g^{(k)} = x + log(e^{z^{(k-1)}} - e^{g^{(k-1)}})$ $z^{(k)} = log(\sum_j^N e^{g_j^{(k)}}) - logk$

Table 1: Operators can be used for MoE routing. $\mathrm{SoftKmax}$, $\mathrm{IterTopk}$ and $\mathrm{GradTopk}$ are mentioned in Su (2024).

$$\mathrm{SoftTopk}(\mathbf{a})^{(k)} = \mathrm{SoftTopk}(\mathbf{a})^{(k-1)} + \mathrm{SE}(\mathbf{a}),$$
$$\mathrm{SE}(\mathbf{a})_i = \begin{cases} 0, a_i \in \mathrm{Topk}(\mathbf{a}) \\ t \cdot \mathrm{Softmax}(\mathbf{a})_i, \text{otherwise}, \end{cases} \tag{6}$$

where t is a constant that gradually decays from the initialization value t_0 to 0.
Network flow modeling. To better capture the characteristics of MoE routing, Equations (4) and (5) are revised as follows:

$$U'(\mathbf{c}, \mathbf{k}) = \{\mathbf{P} \in \mathbb{F}_2^{n \times e} | \mathbf{P}^T \mathbf{1}_n = \mathbf{c}, \mathbf{P}\mathbf{1}_e = \mathbf{k}\}, \tag{7}$$
$$d'_{\mathbf{A}}(\mathbf{c}, \mathbf{k}) = \max_{\mathbf{P} \in U'(\mathbf{c}, \mathbf{k})} \sum_{ij} \mathbf{P}_{ij} \mathbf{A}_{ij}, \tag{8}$$

where $\mathbb{F}_2$ denotes the finite field of $\{0, 1\}$ equipped with addition and multiplication operations. To address this problem, MoE routing can be formulated as a minimum-cost maximum-flow problem as shown in Figure 4. We model tokens and experts as nodes in a flow network graph. Edges from the super source to tokens have capacities representing that each token must be assigned to k experts, while edges from experts to the super sink enforce capacity constraints of c per expert. These edges carry zero cost. Edges between tokens and experts have unit capacity, allowing at most one match per token-expert pair, with costs defined as the negation of their affinity coefficients. A detailed summary of the graph edge properties is provided in Table 2.

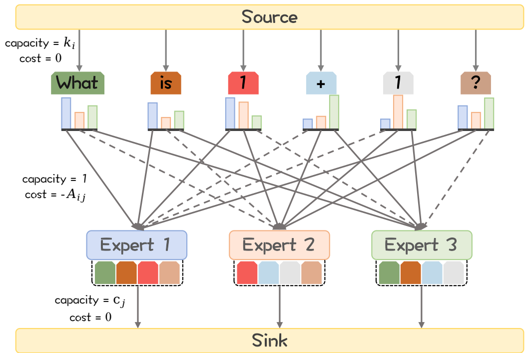

Figure 4: The minimum-cost maximum-flow modeling for MoE routing.

From	To	Capacity	Cost	Count
Source	Token_i	k_i	0	n
Expert_j	Sink	c_j	0	e
Token_i	Expert_j	1	$-A_{ij}$	$n * e$

Table 2: Edges in the graph of Figure 4. Source is the super source, Sink is the super sink, A_{ij} represents the affinity coefficient between Token_i and Expert_j.

Algorithm complexity optimization. TA commonly used and effective approach to solving the minimum-cost maximum-flow problem is the Shortest Path Faster Algorithm (SPFA) (Bellman, 1958; Ford, 1956), which iteratively searches for the lowest-cost augmenting path until no such path remains. However, this method is computationally expensive and inherently sequential, limiting its parallelizability. In top-2 MoE routing, given that the token drop rate in top-1 routing is relatively low (approximately 0 as shown in Figure 2) and that the Sinkhorn algorithm corresponds to the minimum-cost maximum-flow formulation under top-1 routing, we propose a two-stage strategy: first allocate tokens using top-1 routing, followed by applying the Sinkhorn algorithm to handle the residual routing problem. The complete algorithm process is shown in Algorithm 1. For top-k MoE routing with $k > 2$, a trade-off needs to be made between quality (SPFA) and speed (Iter).

4 EVALUATION

4.1 Experimental Setup

Model Architecture. We conduct our experiments using the Llama architecture (Touvron et al., 2023a,b; Grattafiori et al., 2024), incorporating grouped query attention (GQA) (Ainslie

Algorithm 1 Maximum Score Routing For Top-2 Mixture-of-Experts

Input: Weight matrix W_g in the routing function, the number of experts e, temperature $t \leftarrow t_0$, a batch of n tokens $\{x_i\}$

Initialization expert capacity $\mathbf{c}$: $c_j \leftarrow 2 * n/e$

Calculate the token-expert affinity coefficients: $a_{i,j} \leftarrow \text{SoftTopk}(x_i \cdot W_g)_j$

Update temperature: t

Calculate the mask matrix of top-1: $\text{mask}_{i,j} \leftarrow \text{onehot}(\text{Argmax}(a_i), e)_j$

Remove top-1: $a_{i,j} \leftarrow a_{i,j} \cdot \neg\text{mask}_{i,j}$

Update expert capacity $\mathbf{c}$: $c_j \leftarrow max(0, c_j - \sum_i \text{mask}_{i,j})$

Set $\mathbf{k}$: $k_i \leftarrow 1$

The feasible solution space: $U'(\mathbf{c}, \mathbf{k}) = \{\mathbf{P} \in \mathbb{F}_2^{n \times e} | \mathbf{P}^T \mathbf{1}_n = \mathbf{c}, \mathbf{P}\mathbf{1}_e = \mathbf{k}\}$,

Use **Sinkhorn** for an approximate solution: $d'_{\mathbf{A}}(\mathbf{c}, \mathbf{k}) = \max_{\mathbf{P} \in U'(\mathbf{c}, \mathbf{k})} \sum_{ij} \mathbf{P}_{ij} \mathbf{A}_{ij}$

Output: $\{\mathbf{P}_{ij}\}$

et al., 2023), SwiGLU activation function (Shazeer, 2020), RoPE position embedding (Su et al., 2023), and RMSNorm (Zhang and Sennrich, 2019). Our sparsely activated models are constructed by substituting the MLP layers of the dense baseline with MoE layers. We explore three different backbone sizes, as detailed in Table 8.

Baselines. We compared the dense model, GShard MoE (Lepikhin et al., 2020) and GShard-I MoE, the variant with iterative routing strategy (Fedus et al., 2022), SBASE MoE (Clark et al., 2022), ExpertChoice MoE (Zhou et al., 2022), DropLess MoE (Gale et al., 2022), DeepSeek-V2 MoE (Dai et al., 2024; DeepSeek-AI et al., 2024a) along with our proposed **MaxScore** MoE and **MaxScore-I** MoE, which replaces network flow modeling with the iterative rerouting mechanism. All MoEs except DeepSeek use the base configuration with $k = 2$ and $e = 16$, while DeepSeek MoE employs fine-grained experts with $k = 6$ and $e = 64$ and a double-sized shared expert.

Load Balance Loss. All MoE models employ the same auxiliary loss function, defined as

$$\mathcal{L}_{\text{aux}} = \lambda \cdot \frac{1}{e} \sum_{j=1}^{e} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{A}_{i,j} \right) \left(\frac{1}{n} \sum_{i=1}^{N} \mathbf{P}_{i,j} \right), \tag{9}$$

where the $\mathbf{A}_{i,j}$ and $\mathbf{P}_{i,j}$ correspond to the terms defined in Equations (7) and (8).

Training Settings. We adopt the tokenizer from

Model	ARC challenge	ARC easy	BoolQ	Hella-Swag	LAM-BADA	PIQA	RACE	SciQ	Record	OBQA	Avg.
Dense	18.69	40.19	57.06	28.91	16.28	63.71	25.65	64.2	56.05	15.0	38.57
GShard	18.86	**44.49**	61.90	31.74	21.54	66.38	**28.52**	69.4	62.08	16.2	42.11
GShard-I	19.80	44.36	59.94	32.54	21.52	67.03	28.23	68.7	62.84	16.0	42.10
SBASE	18.34	43.73	57.61	30.96	19.70	65.18	27.37	68.3	60.06	16.2	40.75
ExpertChoice	19.37	42.00	61.74	32.10	21.19	66.16	27.18	68.4	62.26	17.6	41.80
DropLess	19.28	44.07	61.16	32.03	21.35	67.14	27.08	67.9	61.55	16.0	41.76
DeepSeek	19.88	44.28	60.55	32.23	21.93	66.97	27.94	70.9	62.57	17.6	42.49
MaxScore-I	**20.90**	43.22	61.71	32.51	21.66	**67.41**	28.42	69.9	63.61	**18.4**	42.77
MaxScore	20.73	**44.49**	**62.23**	**32.85**	**23.27**	**67.41**	**28.52**	**72.5**	**64.00**	**18.4**	**43.44**

Table 3: Results for the base-sized models.

LLama (Touvron et al., 2023a,b; Grattafiori et al., 2024) and set the context length to 512. The batch size is 688, which is the largest setting that allows all baseline models to be trained on 8 NVIDIA A800 GPUs (this constraint arises primarily from the DeepSeek, as shown in Table 10). We can train all baselines with 8 NVIDIA A800 GPUs. All models are trained for $180k$ steps (approximately $65B$ tokens) on C4 dataset (Raffel et al., 2019). This exceeds the compute-optimal dataset size identified by Krajewski et al. (2024), ensuring convergence. For training, we leverage the HuggingFace Trainer (Wolf et al., 2020) integrated with DeepSpeed optimizations, including Zero Redundancy Optimizer (ZeRO) (Rajbhandari et al., 2020) and activation checkpointing (Chen et al., 2016), and we employ bfloat16 for numerical precision and efficiency. We adopt AdamW (Loshchilov and Hutter, 2019) as the optimizer with weight decay wd, adam betas (β_1, β_2) and adam epsilon ϵ. The learning rate is set to be lr following a WSD scheduler (Hu et al., 2024) with a warmup for $2k$ steps and decay over the last $6k$ steps.

Hyperparameters. We perform grid searchs over learning rate lr, weight decay wd, adam betas (β_1, β_2), and adam epsilon ϵ on the GShard baseline, and apply the selected hyperparameters uniformly across all other baselines, as summarized in Table 5. For the scaling factor λ of the auxiliary loss in Equation (9), we perform a grid search over the set $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ for each baseline. The final selected values are 10^{-3} for DeepSeek and 10^{-2} for all other baselines.

Evaluation Settings. We leverage the open source lm-evaluation-harness (Gao et al., 2024) for standardized evaluation on various types of tasks: ARC challenge, ARC easy (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), LAMBADA (Paperno et al., 2016), PIQA (Bisk et al., 2019), RACE (Lai et al., 2017), SciQ (Welbl et al., 2017), Record (Zhang et al., 2018) and OpenBookQA (OBQA) (Mihaylov et al., 2018).

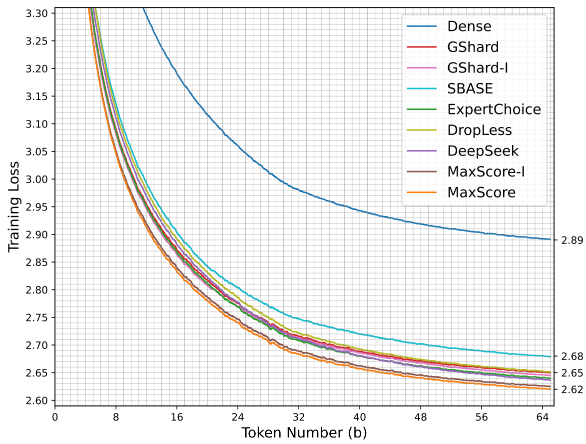

Figure 5: Training loss curve.

4.2 Main Results

Figure 5 presents the training loss curves for all evaluated base-sized models, and Table 3 summarizes the evaluation results of models after training on about $65B$ tokens.

Our proposed MaxScore and MaxScore-I consistently achieve lower training loss compared to all baseline methods throughout the training process and outperform existing baselines on the evaluation datasets. Notably, MaxScore attains the lowest final training loss of approximately 2.62, indicating more effective optimization and improved convergence behavior, and achieves the highest average accuracy of 43.44%, surpassing the best baseline

Model	ARC challenge	ARC easy	BoolQ	Hella-Swag	LAM-BADA	PIQA	RACE	SciQ	Record	OBQA	Avg.
GShard	18.86	**44.49**	61.90	31.74	21.54	66.38	**28.52**	69.4	62.08	16.2	42.11
GShard-I	19.80	44.36	59.94	32.54	21.52	67.03	28.23	68.7	62.84	16.0	42.10
GShard-M	20.14	43.74	59.38	32.27	22.30	66.63	27.61	68.7	62.59	18.2	42.16
GShard-S	20.52	44.30	59.13	32.34	22.54	66.74	28.19	69.4	63.94	**18.4**	42.55
GShard-SI (MaxScore-I)	**20.90**	43.22	61.71	32.51	21.66	**67.41**	28.42	69.9	63.61	**18.4**	42.77
GShard-SM (MaxScore)	20.73	**44.49**	**62.23**	**32.85**	**23.27**	**67.41**	**28.52**	**72.5**	**64.00**	**18.4**	**43.44**

Table 4: Ablation study results. We validate the contributions of the $\mathrm{SoftTopk}(\cdot)$ Operator (**S**), the Minimum-cost Maximum Flow Modeling (**M**), and the Iterative Routing Strategy (**I**).

(DeepSeek) by approximately 0.95%. It also attains state-of-the-art performance on almost all individual tasks. The iterative variant MaxScore-I demonstrates competitive results, particularly excelling on ARC challenge and PIQA.

These findings validate the superiority of our routing mechanisms in integrating the $\mathrm{SoftTopk}(\cdot)$ operator and the minimum cost maximum flow modeling in improving MoE routing quality.

Name	Gird Search	Result
lr	$\{\{1,3\} * \{10^{-4}, 10^{-5}, 10^{-6}\}\}$	$3 * 10^{-5}$
wd	$\{\{0,1,2,3,4\} * 0.05\}$	0.1
(β_1, β_2)	$(0.9, \{0.999, 0.99, 0.95, 0.9\})$	$(0.9, 0.95)$
ϵ	$\{10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$	10^{-6}

Table 5: Gird search and results for hyperparameters.

4.3 Ablation Evaluation

Table 4 presents the ablation study results, validating the individual contributions of the $\mathrm{SoftTopk}(\cdot)$ operator (**S**), the minimum-cost maximum flow modeling (**M**), and the iterative routing strategy (**I**). The variants GShard-S, GShard-M, and GShard-I correspond to incorporating SoftTopk, network flow modeling, and iterative routing respectively, while GShard-SI (MaxScore-I) and GShard-SM (MaxScore) combine these components.

GShard exhibits negligible improvements when employing either network flow modeling or the iterative strategy alone, consistent with observations reported in SwitchTransformer. However, incorporating the $\mathrm{SoftTopk}(\cdot)$ operator individually yields noticeable gains. Furthermore, combining the iterative strategy or network flow modeling with the $\mathrm{SoftTopk}(\cdot)$ operator results in substantial performance improvements. This demonstrates the necessity of the $\mathrm{SoftTopk}(\cdot)$ operator, revealing fundamental limitations in the iterative rerouting mechanism of Fedus et al. (2022) and the optimal transport-based routing of Clark et al. (2022).

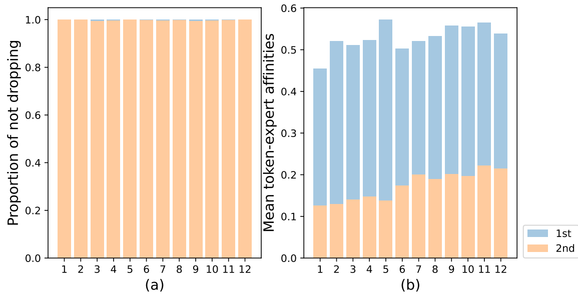

Figure 6: The proportion of not dropping and the mean token-expert affinities in top-2 routing are analyzed separately. The data is derived from our MaxScore MoE with $e = 16$ after training on 65 billion tokens.

By comparing Figure 2 and Figure 6, we observe that network flow modeling effectively eliminates token dropping, and the $\mathrm{SoftTopk}(\cdot)$ operator significantly improves the distribution of token-expert affinities.

Our full model, GShard-SM (MaxScore), consistently achieves the best average performance of 43.44%, outperforming all ablated variants. The synergistic combination of two methodological enhancements yields superadditive performance gains, with empirical results demonstrating that their integrated efficacy surpasses the linear summation of individual improvements.

4.4 Scalability

We perform scaling experiments along two dimensions: model size and sparsity. Detailed configurations are provided in Table 8 and Table 9.

As shown in Figure 7 and Tables 6 and 7, our MaxScore MoE consistently achieves a more significant reduction in training loss and superior evaluation performance compared to traditional MoE baselines such as GShard and DropLess across varying scales. In contrast, DropLess MoE suffers from increased expert load imbalance as sparsity increases, adversely affecting its scalability and overall performance. These results underscore

Size	Model	ARC challenge	ARC easy	BoolQ	Hella-Swag	LAM-BADA	PIQA	RACE	SciQ	Record	OBQA	Avg.
Base	GShard	18.86	**44.49**	61.90	31.74	21.54	66.38	**28.52**	69.4	62.08	16.2	42.11
	DropLess	19.28	44.07	61.16	32.03	21.35	67.14	27.08	67.9	61.55	16.0	41.76
	MaxScore	**20.73**	**44.49**	**62.23**	**32.85**	**23.27**	**67.41**	**28.52**	**72.5**	**64.00**	**18.4**	**43.44**
Large	GShard	19.88	45.58	62.16	33.34	23.69	67.74	29.28	70.0	64.99	19.2	43.59
	DropLess	20.05	45.24	61.19	33.97	23.60	67.63	27.66	69.7	63.95	17.2	43.02
	MaxScore	**20.90**	**45.92**	**62.39**	**34.00**	**24.96**	**68.28**	**29.67**	**74.3**	**66.12**	**19.8**	**44.63**
XL	GShard	20.05	46.68	63.09	35.14	25.05	69.31	29.19	72.7	67.50	20.0	44.87
	DropLess	20.14	46.60	61.69	35.11	24.34	68.34	29.19	72.4	67.55	20.2	44.56
	MaxScore	**21.22**	**47.60**	**63.60**	**35.57**	**25.93**	**69.95**	**29.90**	**75.2**	**67.90**	**21.6**	**45.85**

Table 6: Results of scaling in model size.

Sparsity	Model	ARC challenge	ARC easy	BoolQ	Hella-Swag	LAM-BADA	PIQA	RACE	SciQ	Record	OBQA	Avg.
2:16	GShard	18.86	**44.49**	61.90	31.74	21.54	66.38	**28.52**	69.4	62.08	16.2	42.11
	DropLess	19.28	44.07	61.16	32.03	21.35	67.14	27.08	67.9	61.55	16.0	41.76
	MaxScore	**20.73**	**44.49**	**62.23**	**32.85**	**23.27**	**67.41**	**28.52**	**72.5**	**64.00**	**18.4**	**43.44**
2:32	GShard	19.62	44.57	62.28	32.63	21.99	67.19	**29.04**	69.6	62.77	18.2	42.79
	DropLess	19.62	44.51	62.23	32.36	21.79	67.10	27.46	68.3	62.20	16.8	42.24
	MaxScore	**20.90**	**44.60**	**63.73**	**33.24**	**23.76**	**67.63**	**29.04**	**73.5**	**64.41**	**18.8**	**43.96**
2:64	GShard	19.80	44.86	62.26	33.05	22.20	67.30	28.46	69.4	63.17	17.6	42.81
	DropLess	19.60	44.69	62.40	32.79	21.65	67.27	27.56	69.5	63.05	17.6	42.61
	MaxScore	**21.11**	**46.17**	**64.24**	**33.38**	**23.41**	**67.95**	**28.90**	**73.3**	**64.60**	**19.0**	**44.21**

Table 7: Results of scaling in sparsity.

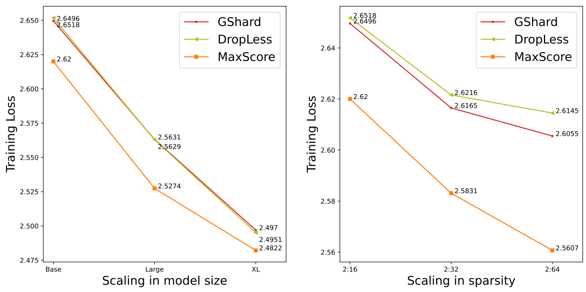

Figure 7: Scalability with respect to model size and sparsity. The Y-axis represents the training loss of each model after training on approximately 65 billion tokens.

MaxScore's effectiveness in harnessing both model capacity and sparsity to improve MoE routing and model accuracy.

4.5 Load Balancing Analysis

Figure 8 illustrates the sorted ratio between the number of tokens assigned to each expert and the capacity $c = \frac{k*n}{e}$ in the first MoE layer with $k = 2$ and $e = 16$ after training on about 65 billion tokens. For ExpertChoice MoE, this ratio remains strictly equal to 1, indicating perfect load balancing by design. MaxScore MoE achieves near-ideal load balance with a mean ratio of 0.9996, closely approximating ExpertChoice. In contrast, GShard exhibits notable load imbalance caused by token

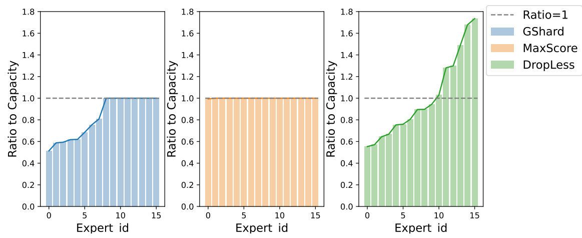

Figure 8: The sorted ratio between the number of tokens each expert allocated and Capacity $c = k * n/e$ in the first layer of MoE with $k = 2$ and $e = 16$. For ExpertChoice MoE, the ratio is always equal to 1. The mean ratios of GShard MoE, MaxScore MoE, and DropLess MoE are 0.8237, 0.9996, and 1, respectively.

dropping, resulting in a lower mean ratio of 0.8237 and uneven token distribution across experts. DropLess displays extreme variability, with ratio values ranging from 0.55 to 1.74, indicating significant disparity in expert loads. These findings demonstrate MaxScore's superior capability in mitigating load imbalance relative to traditional approaches.

4.6 Different SoftTopk Operators

We evaluate various $\mathrm{SoftTopk}(\cdot)$ operators listed in Table 1. As illustrated in Figure 9, none yield performance improvements except for our proposed operator defined in Equation (6). We hypothesize that the increased complexity of alternative operators may hinder effective model learning.

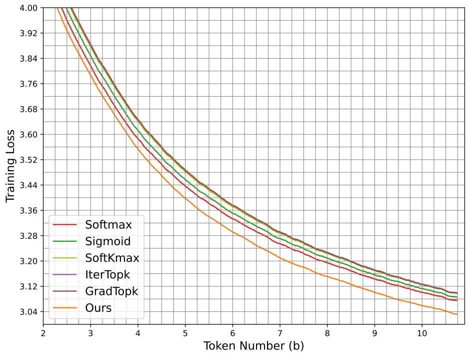

Figure 9: Results of different operators.

4.7 Hyperparameter t in SoftTopk Operator

We perform hyperparameter tuning for the parameter $\mathbf{t}$ in our $\mathrm{SoftTopk}(\cdot)$ operator defined in Equation (6), exploring two strategies: maintaining a constant value or decaying t to 1 over training on 10b tokens. As shown in Figure 10, the optimal approach initializes $\mathbf{t_0}$=4 and gradually decays it to 1.

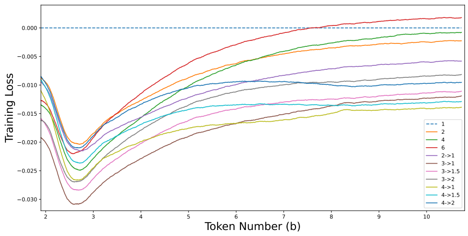

Figure 10: Hyperparameter tuning experiment.

5 Conclusion and Future Work

This work introduces MaxScore MoE, a novel mixture-of-experts routing paradigm formulated via minimum-cost maximum flow modeling and the integration of a differentiable $\mathrm{SoftTopk}(\cdot)$ operator. To our knowledge, this is the first successful integration of network flow modeling and SoftTopk within MoE routing. The synergistic combination of these components yields superadditive performance gains, with empirical evidence showing that their joint effect surpasses the linear sum of individual contributions. Future work will focus on evaluating the method at larger model scales and across more diverse benchmarks to validate its generality and robustness.

Limitations

Due to limited computational resources, our experiments are restricted to smaller-scale models, precluding direct comparison with larger, state-of-the-art models. Additionally, the training data volume is relatively modest; further experiments with substantially larger token budgets are necessary to fully assess the ultimate benefits and convergence properties of our approach.

Model	Base	Large	XL
Activated Params	$162M$	$317M$	$600M$
Total Params	$757M$	$1.6B$	$3.2B$
FLOPs	$302G$	$603G$	$1.2T$
hidden_size	768	1128	1608
num_heads	12	12	12
num_layers	12	12	12

Table 8: Configurations for different dense backbones. FLOPs are calculated with a single sequence of 512 tokens. The intermediate_size of the MLP layer in the dense model is four times that of the hidden_size, while for the top-k MoE, the intermediate_size in each expert is reduced to $1/k$, compared with the dense model.

Sparsity	$\mathbf{2:16}$	$\mathbf{2:32}$	$\mathbf{2:64}$
Activated Params	$162M$	$162M$	$162M$
Total Params	$757M$	$1475M$	$2867M$
FLOPs	$302G$	$302G$	$302G$

Table 9: Configurations of different sparsity.

Models	Peak GPU Memory Usage	Tokens processed Per Hour
GShard	$71.7GB$	$0.308b$
ExpertChoice	$71.7GB$	$0.301b$
DropLess	$73.4GB$	$0.296b$
DeepSeek	**78.8 GB**	**0.277b**
MaxScore-I	$71.7GB$	$0.305b$
GShard	$71.7GB$	$0.299b$

Table 10: The peak GPU memory usage and the speed of processing of MoE models during training. DeepSeek MoE's use of fine-grained experts leads to larger GPU memory and slower speed (Dai et al., 2024; DeepSeek-AI et al., 2024a).

Acknowledgments

This work was supported in part by National Key Research and Development Program of China under Grant No. 2020YFA0804503, National Natural Science Foundation of China under Grant No. 62272264, and ByteDance Doubao Large Model Fund Project under Grant No. CT20240909109354.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *Preprint*, arXiv:2305.13245.

Richard Bellman. 1958. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90.

Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. 2016. Conditional computation in neural networks for faster models. *Preprint*, arXiv:1511.06297.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *Preprint*, arXiv:1911.11641.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *Preprint*, arXiv:1604.06174.

Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, George van den Driessche, Eliza Rutherford, Tom Hennigan, Matthew Johnson, Katie Millican, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Jack Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. 2022. Unified scaling laws for routed language models. *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. *Preprint*, arXiv:1905.10044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

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

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao, Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei

An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2014. Learning factored representations in a deep mixture of experts. *Preprint*, arXiv:1312.4314.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

Lester Randolph Ford. 1956. Network flow theory. *Rand Corporation Paper, Santa Monica, 1956*.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. 2022. Megablocks: Efficient sparse training with mixture-of-experts. *Preprint*, arXiv:2211.15841.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. The language model evaluation harness.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *Preprint*, arXiv:2404.06395.

Changho Hwang, Wei Cui, Yifan Xiong, Ziyue Yang, Ze Liu, Han Hu, Zilong Wang, Rafael Salas, Jithin Jose, Prabhat Ram, Joe Chau, Peng Cheng, Fan Yang, Mao Yang, and Yongqiang Xiong. 2023. Tutel: Adaptive mixture-of-experts at scale. *Preprint*, arXiv:2206.03382.

Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. 2024. Scaling laws for fine-grained mixture of experts. *Preprint*, arXiv:2402.07871.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *Preprint*, arXiv:1704.04683.

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. *Preprint*, arXiv:2006.16668.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

André F. T. Martins and Ramón Fernandez Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. *Preprint*, arXiv:1602.02068.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *Preprint*, arXiv:1809.02789.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. 2024. Olmoe: Open mixture-of-experts language models. *Preprint*, arXiv:2409.02060.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *Preprint*, arXiv:1606.06031.

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. *Preprint*, arXiv:1905.05702.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. *Preprint*, arXiv:1910.02054.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

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. *Preprint*, arXiv:1701.06538.

Jianlin Su. 2024. After softmax: Finding a smooth approximation for top-k.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Gary R Waissi. 1994. Network flows: Theory, algorithms, and applications.

Ziteng Wang, Jun Zhu, and Jianfei Chen. 2025. Remoe: Fully differentiable mixture-of-experts with relu routing. *Preprint*, arXiv:2412.14711.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *Preprint*, arXiv:1707.06209.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *Preprint*, arXiv:1905.07830.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Preprint*, arXiv:1910.07467.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *Preprint*, arXiv:1810.12885.

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. *Preprint*, arXiv:2202.09368.

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. *Preprint*, arXiv:2202.08906.