MH-MoE: Multi-Head Mixture-of-Experts

Shaohan Huang Xun Wu Shuming Ma Furu Wei
Microsoft Research
https://aka.ms/GeneralAI

Abstract

Multi-Head Mixture-of-Experts (MH-MoE) [WHWW24] demonstrates superior performance by using the multi-head mechanism to collectively attend to information from various representation spaces within different experts. In this paper, we present a novel implementation of MH-MoE that maintains both FLOPs and parameter parity with sparse Mixture of Experts models. Experimental results on language models show that the new implementation yields quality improvements over both vanilla MoE and fine-grained MoE models. Additionally, our experiments demonstrate that MH-MoE is compatible with 1-bit Large Language Models (LLMs) such as BitNet [MWM+24].

1 Sparse Mixture-of-Experts

Sparse Mixture-of-Experts (SMoE) provides a highly efficient way to scale neural network training and achieves better performance than dense models in various tasks [SMM+17, LLX+20, DHD+21, KGS+21, CDH+22, CCG+22, ZCCC23, PKM+23]. SMoE dynamically selects which parameters to use for each input, rather than applying the same parameters uniformly. This approach allows the networks to significantly increase the number of parameters while maintaining a roughly constant number of FLOPs per token. Recent advancements in large language models employing Mixture of Experts (MoE) Transformers have demonstrated successful scaling to substantial sizes, accompanied by remarkable performance [JSR+24, DDZ+24]. For instance, the Mixtral 8×7B, an SMoE model consisting of 8 experts (with activated 12.9 billion parameters), has been shown to outperform models such as LLaMA-70B.

In MoE architectures, the traditional Feed-Forward Networks (FFNs) within a Transformer are replaced by MoE layers. These MoE layers consist of multiple experts, each functioning as a standard FFN. The model employs a gating mechanism to route tokens to one or two of these experts per layer, utilizing either a top-1 or top-2 gating method. The MoE layer consists of two components: $\mathbf{E}$ experts, each presented as $\text{Expert}_i : \mathbb{R}^d \to \mathbb{R}^d$, and a gate function, $G : \mathbb{R}^d \to \mathbb{R}^{\mathbf{E}}$. Given an input $\mathbf{x} \in \mathbb{R}^d$, the conditional output $\mathbf{y} \in \mathbb{R}^d$ is the weighted sum of gate function $G(\mathbf{x})$ and experts outputs $\{\text{Expert}_i(\mathbf{x})\}_{i=0}^{\mathbf{E}}$. The output $\mathbf{y}$ is computed by these activated experts, where $\Phi = \text{Top}_k\left(\text{Expert}_i\right)$ denote the set of activated experts and $|\Phi| = k$.

$$\mathbf{y} = \sum_{p \in \Phi} G\left(\mathbf{x}\right) \cdot \text{Expert}_p\left(\mathbf{x}\right). \tag{1}$$

2 Multi-Head Mixture-of-Experts

2.1 Review of Multi-Head Mixture-of-Experts

Wu et al., [WHWW24] introduced Multi-Head Mixture-of-Experts (MH-MoE), a novel approach that enhances the multi-head mechanism by enabling it to collectively attend to information from various representation spaces within different experts. MH-MoE incorporates two key modifications

Preprint.

compared to the standard Sparse Mixture-of-Experts: adding a "heads" dimension $\mathbf{h}$ to the token dimension and ingratiating two linear projection layers at both the beginning and the end of the MoE layer.

Given an input $\mathbf{x} \in \mathbb{R}^d$, d is the length of token dimension. First, $\mathbf{x}$ is projected by a linear layer with parameter matrices $\mathbf{W}_{\text{head}} \in \mathbb{R}^{d \times d}$,

$$\hat{\mathbf{x}} = \mathbf{x}\mathbf{W}_{\text{head}} \tag{2}$$

where $\hat{\mathbf{x}} \in \mathbb{R}^d$. After that, the token $\hat{\mathbf{x}}$ is split into h sub-tokens along the token dimensions, and these sub-tokens are arranged in parallel according to the original token sequence, forming a new feature space $[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ..., \tilde{\mathbf{x}}_h]$, where $\tilde{\mathbf{x}}_h \in \mathbb{R}^{\frac{d}{h}}$ and h denotes the number of heads.

Following the SMoE framework, the transformed input $\tilde{\mathbf{x}}$ is fed into a MoE layer. This layer consists of $\mathbf{E}$ experts, denoted as $\text{Expert}_i : \mathbb{R}^{\frac{d}{h}} \to \mathbb{R}^{\frac{d}{h}}$, and a gating function $G : \mathbb{R}^{\frac{d}{h}} \to \mathbb{R}^{\mathbf{E}}$. The output $\tilde{\mathbf{y}} \in \mathbb{R}^{\frac{d}{h}}$ is computed as following:

$$\tilde{\mathbf{y}} = \sum_{p \in \Phi} G\left(\tilde{\mathbf{x}}\right) \cdot \text{Expert}_p\left(\tilde{\mathbf{x}}\right). \tag{3}$$

where Φ is the set of activated experts. After processing through the MoE layer, all obtained outputs $\tilde{\mathbf{y}}$ are rearranged into the original order of sub-tokens and concatenated together to form $\hat{\mathbf{y}} \in \mathbb{R}^d$. This concatenated output $\hat{\mathbf{y}}$ is then projected using a merge layer with parameter matrices $\mathbf{W}_{\text{merge}} \in \mathbb{R}^{d \times d}$. This step ensures the effective integration of multiple features, capturing detailed information from different expert representation spaces.

$$\mathbf{y} = \hat{\mathbf{y}}\mathbf{W}_{\text{merge}} \tag{4}$$

where $\mathbf{y}$ is the final output of the MH-MoE layer.

2.2 Complexity Analysis

We use $\mathbf{B}$ to represent the number of tokens in batches, $\mathbf{d}$ as the token dimension, $\mathbf{d_{moe}}$ as the intermediate dimension in Expert$(\mathbf{x})$, and $\mathbf{h}$ as the number of multi-heads in MH-MoE. Assuming we use "position-wise feed-forward networks" (FFN) [VSP+17] in Expert$(\mathbf{x})$, and opting for a version with no bias, Expert$(\mathbf{x})$ can be computed as follows, where $\mathbf{X} \in \mathbb{R}^{\mathbf{B} \times \frac{\mathbf{d}}{\mathbf{h}}}$, $\mathbf{W_1} \in \mathbb{R}^{\frac{\mathbf{d}}{\mathbf{h}} \times \mathbf{d_{moe}}}$ and $\mathbf{W_2} \in \mathbb{R}^{\mathbf{d_{moe}} \times \frac{\mathbf{d}}{\mathbf{h}}}$:

$$\text{Expert}(\mathbf{x}) = \text{FFN}_{\text{ReLU}}(\mathbf{X}, \mathbf{W_1}, \mathbf{W_2}) = \max(\mathbf{X}\mathbf{W_1}^\top, 0)\mathbf{W_2} \tag{5}$$

The number of scalar multiplications in MH-MoE is:

$$\overbrace{2\mathbf{B}\mathbf{d}^2 - \mathbf{B}\mathbf{d}}^{\text{Head Layer}} + \overbrace{(4\mathbf{B}\mathbf{d}\mathbf{d_{moe}} - \mathbf{B}\mathbf{d} - \mathbf{B}\mathbf{d_{moe}}\mathbf{h}) \cdot k}^{\text{Activated Experts}} + \overbrace{2\mathbf{B}\mathbf{d}^2 - \mathbf{B}\mathbf{d}}^{\text{Merge Layer}} \tag{6}$$

Assuming we use top-1 gating (set $k = 1$) and the intermediate dimension $\mathbf{d_{moe}} = 4\mathbf{d}$, for sparse MoE, which does not include the head layer and merge layer, the number of scalar multiplications is $16\mathbf{B}\mathbf{d}^2 - 5\mathbf{B}\mathbf{d}$ and the leading term is $16\mathbf{B}\mathbf{d}^2$.

In MH-MoE [WHWW24], they set the intermediate dimension $\mathbf{d_{moe}} = 4\beta\mathbf{h}\mathbf{d}$, where β is a hyperparameter employed to scale the inner hidden dimension of FFNs. When the number of heads $\mathbf{h} = 4$ and β is $\frac{63}{64}$ in their experiment, the scalar multiplications in [WHWW24] is $67\mathbf{B}\mathbf{d}^2 - \frac{75}{4}\mathbf{B}\mathbf{d}$ and the leading term is $67\mathbf{B}\mathbf{d}^2$. Although the activated parameters and whole model parameters in [WHWW24] are on par with sparse MoE, the FLOPS of [WHWW24] is significantly higher than the baseline.

In our work, we will adjust the parameters in MH-MoE to maintain FLOPs parity with the vanilla method. Assuming the number of heads $\mathbf{h} = 2$, we aim to keep the leading term at $16\mathbf{B}\mathbf{d}^2$. To achieve this, we set the intermediate dimension $\mathbf{d_{moe}} = 3\mathbf{d}$ and increase the number of experts to match the model parameter count. Under this configuration, the number of scalar multiplications is $16\mathbf{B}\mathbf{d}^2 - 6\mathbf{B}\mathbf{d}$, ensuring that the leading term is on par with sparse MoE.

Alternatively, we can decrease the intermediate dimension to $\mathbf{d_{moe}} = \frac{3}{2}\mathbf{d}$ and switch from top-1 gating to top-2 gating. This adjustment allows us to match not only the model parameters but also achieve parity in the number of scalar multiplications.

2.3 Utilization Guidelines

In this section, we will explain how to set the intermediate dimension $\mathbf{d_{moe}}$ and the number of experts k in the mixture-of-experts layer. This process transforms a standard SMoE model into a MH-MoE model, ensuring that both the model parameters and the FLOPS are comparable to those of the standard SMoE model. The number of scalar multiplications in the sparse MoE is given by the following equation:

$$(4\mathbf{Bdd_{moe}} - \mathbf{Bd_{moe}} - \mathbf{Bd}) \cdot k \tag{7}$$

Our goal is to ensure that the FLOPS of the MH-MoE model are equal to those of the standard SMoE model. We only consider the leading term of the equation, which is $4\mathbf{Bdd_{moe}} \cdot k$. From the equation 6, we can obtain the leading term of the FLOPS of the MH-MoE model is $4\mathbf{Bd}^2 + 4\mathbf{Bdd_{mhmoe}} \cdot k$ The intermediate dimension $\mathbf{d_{mhmoe}}$ can be set using the following equation:

$$\mathbf{d_{mhmoe}} = \mathbf{d_{moe}} - \frac{\mathbf{d}}{k} \tag{8}$$

where $\mathbf{d_{moe}}$ is the intermediate dimension of the standard SMoE model, $\mathbf{d}$ is the input dimension, and k is the number of experts. By setting the intermediate dimension $\mathbf{d_{mhmoe}}$ using the equation 8, we can ensure that the FLOPs of the MH-MoE model are equal to those of the standard SMoE model.

As shown in equation 8, the MoE intermediate dimension of the MH-MoE model is smaller than that of the standard SMoE model. To maintain the same number of model parameters, we need to increase the number of experts $\mathbf{E}$ in the mixture-of-experts layer. The number of experts $\mathbf{E}$ in the mixture-of-experts layer can be set using the following equation:

$$\begin{aligned} \overbrace{2\mathbf{dd_{moe}} \cdot \mathbf{E_{moe}}}^{\text{\#parameter of standard MoE}} &= \overbrace{2\mathbf{d}^2 + 2\frac{\mathbf{d}}{\mathbf{h}}\mathbf{d_{mhmoe}} \cdot \mathbf{E_{mhmoe}}}^{\text{\#parameter of MH-MoE}} \\ &= 2\mathbf{d}^2 + 2\frac{\mathbf{d}}{\mathbf{h}}\left(\mathbf{d_{moe}} - \frac{\mathbf{d}}{k}\right) \cdot \mathbf{E_{mhmoe}} \end{aligned} \tag{9}$$

This equation ensures that the number of parameters in the MH-MoE model matches that of the standard MoE model by appropriately adjusting the number of experts.

To illustrate with an example, let's assume $\mathbf{d_{moe}} = 4\mathbf{d}$, we use top-1 gating (i.e., $k = 1$), and the number of heads is 3 (i.e., $\mathbf{h} = 3$). Using these values, we can derive the number of experts in the mixture-of-experts layer for the MH-MoE model.

First, recall the equation we derived for the intermediate dimension $\mathbf{d_{mhmoe}} = \mathbf{d_{moe}} - \frac{\mathbf{d}}{k}$. Substituting $\mathbf{d_{moe}} = 4\mathbf{d}$ and $k = 1$: $\mathbf{d_{mhmoe}} = 4\mathbf{d} - \frac{\mathbf{d}}{1} = 4\mathbf{d} - \mathbf{d} = 3\mathbf{d}$. Next, we use the equation for parameter parity:

$$\begin{aligned} 2\mathbf{dd_{moe}} \cdot \mathbf{E_{moe}} &= 2\mathbf{d}^2 + 2\frac{\mathbf{d}}{\mathbf{h}}\mathbf{d_{mhmoe}} \cdot \mathbf{E_{mhmoe}} \\ &= 2\mathbf{d}^2 + 2\frac{\mathbf{d}}{3}(3\mathbf{d}) \cdot \mathbf{E_{mhmoe}} \\ &= 2\mathbf{d}^2 + 2\mathbf{d}^2 \cdot \mathbf{E_{mhmoe}} \end{aligned} \tag{10}$$

Thus, the number of experts in the mixture-of-experts layer for the MH-MoE model is $\mathbf{E_{mhmoe}} = 4\mathbf{E_{moe}} - 1$.

3 Experiments Setup

We adopt a decoder-only Transformer [RNSS18, RWC+19] to evaluate the variants of MH-MoE and the baseline models on the RedPajama dataset [Com23]. We use the same code base, training parameters, and pre-training tasks across all experiments. The decoder architecture comprises 12 layers with a model dimension of 768.

For the SMoE configuration, we employ top-1 gating with 8 experts, integrating MoE Transformer layers every two layers. The feedforward network utilizes SwiGLU [Sha20], with the intermediate dimension $\mathbf{d_{moe}}$ set to 2048.

We also implement a fine-grained version of the sparse MoE. In this configuration, the intermediate dimension is reduced to 1024, while the number of experts is increased to 16.

For MH-MoE, we compare two variants based on the number of heads, either 2 or 3. When the head number is 2, we set the intermediate dimension $\mathbf{d_{mhmoe}}$ to 768 and use top-2 gating to maintain FLOPs parity, increasing the number of experts to 40. For the variant with 3 heads, we set the intermediate dimension to 512, employ top-3 gating, and increase the number of experts to 96.

Furthermore, as employing a residual MoE setting, i.e., using shared experts [DDZ+24], has been shown to be effective in MoE models, we also conduct experiments under this setting to comprehensively validate the effectiveness of our MH-MoE. Specifically, a shared expert with the same size (hidden dimension is set to 2048) is applied to all MoE models.

4 Experiments

5 Language Modeling Evaluation

For all experiments, we pre-train for 100,000 steps, with each training batch consisting of 0.5 million tokens. To evaluate the performance of different model architectures, we compute the perplexity on the validation set. Perplexity is reported at both 50,000 and 100,000 steps.

Table 1 reports the results for MoE models without a shared expert, while Table 2 summarizes the results for MoE models incorporating a shared expert. Notably, across both settings, our MH-MoE consistently achieve lower perplexities compared to both the standard sparse MoE and its fine-grained variant. Additionally, the configuration with three heads outperforms the two-head configuration, demonstrating superior performance.

Table 1: Validation set perplexity for the language modeling task. All models are matched in terms of parameters and computation.

Model	Training Steps	RedPajama	Wiki	C4
Dense	50,000	13.01	12.95	17.41
SMoE		11.87	10.51	15.63
Fine-grained SMoE		11.68	10.18	15.21
MH-MoE (head=2)		11.60	10.11	15.11
MH-MoE (head=3)		**11.45**	**10.00**	**14.90**
Dense	100,000	12.13	11.58	16.21
SMoE		10.90	9.68	14.35
Fine-grained SMoE		10.74	9.38	13.97
MH-MoE (head=2)		10.70	9.26	13.80
MH-MoE (head=3)		**10.51**	**9.18**	**13.63**

Table 2: Validation set perplexity for the language modeling task. All MoE models apply a shared expert [DDZ+24] with the same size and matched in terms of parameters and computation.

Model	Training Steps	RedPajama	Wiki	C4
SMoE	50,000	11.76	10.33	15.19
Fine-grained SMoE		11.51	10.06	15.01
MH-MoE (head=2)		11.48	9.91	14.87
MH-MoE (head=3)		**11.26**	**9.74**	**14.82**
SMoE	100,000	10.41	9.44	14.30
Fine-grained SMoE		10.66	9.15	13.78
MH-MoE (head=2)		10.36	8.79	13.66
MH-MoE (head=3)		**10.28**	**8.72**	**13.49**

6 1-bit MH-MoE

The recent impressive performance of BitNet [MWM+24] in quantizing and deploying large-scale models is heralding a new era for 1-bit Large Language Models (LLMs). Building on their impressive model performance, we conducted further experiments to explore whether our MH-MoE can effectively integrate with BitNet to achieve enhanced model optimization.

We employ the same experimental setting listed in Section 3, with the exception that all the models are quantized using BitNet. The corresponding experimental results are shown in Table 3. In the 1-bit training and validation setting, we observed that our MH-MoE consistently outperformed other models, e.g., SMoE and Fine-grained SMoE. This demonstrates that MH-MoE integrates effectively with BitNet, enabling more lightweight deployment of MoE models without compromising performance.

Besides, we observe a performance gap between the experimental results under the BitNet setting (shown in Table 3) and those under the non-BitNet setting (shown in Table 1). We attribute this discrepancy to the fact that when the model size is relatively small, BitNet tends to degrade performance, a finding that aligns with the conclusions reported in the original BitNet paper [MWM+24].

Table 3: Validation set perplexity for the language modeling task. All dense and MoE models are quantized and trained using BitNet [MWM+24], and matched in terms of parameters and computation.

Model	Training Steps	RedPajama	Wiki	C4
Dense	50,000	32.17	27.56	35.85
SMoE		29.18	24.70	32.34
Fine-grained SMoE		29.04	24.51	32.03
MH-MoE (head=2)		28.84	24.27	31.86
MH-MoE (head=3)		**28.77**	**24.13**	**31.81**
Dense	100,000	30.04	24.75	33.55
SMoE		26.78	21.54	29.73
Fine-grained SMoE		26.68	21.42	29.50
MH-MoE (head=2)		26.59	21.11	29.27
MH-MoE (head=3)		**26.47**	**21.06**	**29.14**

7 Ablations

In this section, we conduct a detailed ablation study focusing on the head layer and the merge layer, both of which are integral components of MH-MoE. The design of these layers draws inspiration from the multi-head attention mechanism [VSP+17]. Specifically, in our Multi-Head Mixture-of-Experts model, we conceptualize the head layer as constituting the query, key, and value projections. The merge layer, on the other hand, is considered the output projection. It is crucial to thoroughly investigate their contributions and understand their impact

We separately integrate head and merge layers into both our baseline SMoE and fine-grained SMoE models. Table 4 presents the validation set perplexity for various models with and without these layers. It is important to note that all models without the head and merge layers maintain the same number of scalar multiplications, and similarly, all models with the head and merge layers also maintain an equivalent number of scalar multiplications.

Our findings indicate that for both the SMoE and fine-grained SMoE models, the addition of head and merge layers—which inevitably increases the number of FLOPs in these layers—results in only marginal gains in performance. In contrast, for the MH-MoE model, the inclusion of the head and merge layers leads to significant improvements in performance. This underscores the critical role these layers play in enhancing the effectiveness of the MH-MoE model.

We further analyze the head and merge layers separately. As shown in Table 5, both of these layers contribute positively to model performance. Notably, the head layer provides a more substantial

Table 4: Validation set perplexity for different models with and without head and merge layers.

Model	w/ head & merge layer	RedPajama	Wiki	C4
SMoE	✗	11.87	10.51	15.63
SMoE	✓	11.84	10.48	15.61
Fine-grained SMoE	✗	11.68	10.18	15.21
Fine-grained SMoE	✓	11.67	10.18	15.19
MH-MoE (head=2)	✗	11.71	10.16	15.23
MH-MoE (head=2)	✓	11.46	9.98	14.89

gain compared to the merge layer. This suggests that while both layers are beneficial, the head layer plays a more critical role in enhancing model effectiveness.

Table 5: Validation set perplexity for ablation of head and merge layers.

w/ head layer	w/ merge layer	RedPajama	Wiki	C4
✗	✗	11.97	10.40	15.52
✓	✗	11.74	10.18	15.17
✗	✓	11.84	10.27	15.36
✓	✓	11.60	10.11	15.11

Through our ablation experiments, we aim to dissect the individual contributions of the head and merge layers. By systematically altering or removing components within these layers, we can gain insights into how each part influences the overall model performance. This analysis not only helps in validating our design choices but also provides guidance for potential improvements and optimizations in future iterations of the model.

8 Conclusion

In this work, we present a new implementation of MH-MoE to ensure FLOPs parity with sparse Mixture of Experts (MoE) models. Our experimental results show that the new variants both outperform both vanilla SMoE models and fine-grained MoE models under various experimenta settings. Additionally, we conducted ablation experiments to analyze the impact of head and merge layers. We demonstrate that both head and merge layers improve model performance, with the head layer yielding particularly substantial gains.

References

[CCG+22] Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. *arXiv preprint arXiv:2202.01169*, 2022.

[CDH+22] Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, et al. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems*, 35:34600–34613, 2022.

[Com23] Together Computer. Redpajama: An open source recipe to reproduce llama training dataset, 2023.

[DDZ+24] 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. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *CoRR*, abs/2401.06066, 2024.

[DHD+21] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. *arXiv preprint arXiv:2112.06905*, 2021.

[JSR+24] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[KGS+21] Kenichi Kumatani, Robert Gmyr, Felipe Cruz Salinas, Linquan Liu, Wei Zuo, Devang Patel, Eric Sun, and Yu Shi. Building a great multi-lingual teacher with sparsely-gated mixture of experts for speech recognition. *arXiv preprint arXiv:2112.05820*, 2021.

[LLX+20] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[MWM+24] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *CoRR*, abs/2402.17764, 2024.

[PKM+23] Hai Pham, Young Jin Kim, Subhabrata Mukherjee, David P Woodruff, Barnabas Poczos, and Hany Hassan Awadalla. Task-based moe for multitask multilingual machine translation. *arXiv preprint arXiv:2308.15772*, 2023.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[Sha20] Noam Shazeer. Glu variants improve transformer, 2020.

[SMM+17] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010, 2017.

[WHWW24] Xun Wu, Shaohan Huang, Wenhui Wang, and Furu Wei. Multi-head mixture-of-experts, 2024.

[ZCCC23] Xinyu Zhao, Xuxi Chen, Yu Cheng, and Tianlong Chen. Sparse moe with language guided routing for multilingual machine translation. In *Conference on Parsimony and Learning (Recent Spotlight Track)*, 2023.