HyperMoE: Towards Better Mixture of Experts via Transferring Among Experts

Anonymous ACL submission

Abstract

The Mixture of Experts (MoE) for language models has been proven effective in augmenting the capacity of models by dynamically routing each input token to a specific subset of experts for processing. Despite the success, most existing methods face a challenge for balance between sparsity and the availability of expert knowledge: enhancing performance through increased use of expert knowledge often results in diminishing sparsity during expert selection. To mitigate this contradiction, we propose HyperMoE, a novel MoE framework built upon Hypernetworks. This framework integrates the computational processes of MoE with the concept of knowledge transferring in multi-task learning. Specific modules generated based on the information of unselected experts serve as supplementary information, which allows the knowledge of experts not selected to be used while maintaining selection sparsity. Our comprehensive empirical evaluations across multiple datasets and backbones establish that HyperMoE significantly outperforms existing MoE methods under identical conditions concerning the number of experts. We have anonymized our code and uploaded it into the supplementary materials.

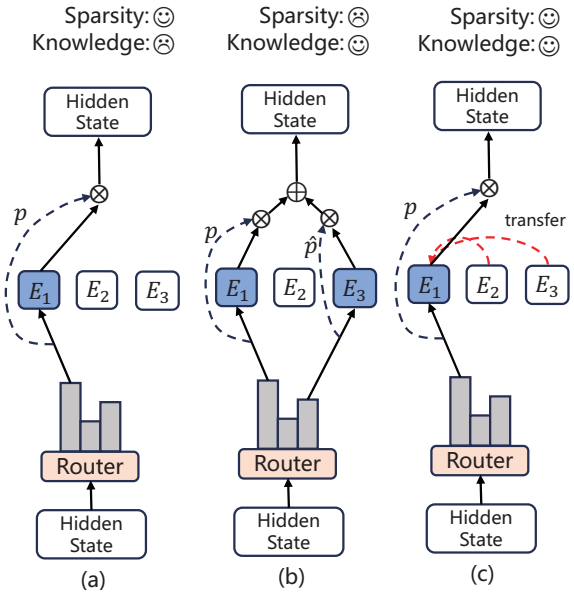

Figure 1: A trade-off in MoE: (a) A lower number of selectable experts can maintain sparsity but limits the availability of expert knowledge. (b) Increasing the number of selectable experts can improve performance but decrease sparsity. (c) Transferring partial knowledge from the unselected experts $E_{2,3}$ to the selected experts E_1 can improve the availability of expert knowledge while maintaining sparsity.

1 Introduction

The accelerated advancement of large language models (LLMs) has culminated in their widespread application across various domains, including healthcare, education, and social interactions (Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023). The remarkable capabilities of these models are attributed to the enhancements in their scale. Nevertheless, the scaling of dense models is often hampered by significant computational demands, posing a challenge to developing the Natural Language Processing (NLP) community. In response, sparse activation models have emerged as a solution (Artetxe et al., 2022; Du et al., 2022), activating only a subset of parameters for different inputs, thus mitigating computational costs. One of the most representative methods is the Mixture of Experts (MoE, Shazeer et al. (2017)), which routers different inputs to specific groups of experts, thereby enlarging the model's capacity without increasing computational burdens.

The key to effectively reducing computational costs lies in the sparsity of expert selection, with the number of experts selected for each token being kept at a lower level. In practical applications or experiments, existing works (Roller et al., 2021a; Fedus et al., 2022; Rajbhandari et al., 2022; Xue et al., 2023) usually select only one or two experts per input. However, increasing the number of selected

experts per token can enhance the availability of expert knowledge and improve the performance of downstream tasks (Yang et al., 2019; Shazeer et al., 2017; He et al., 2023). This scenario positions MoE model in a predicament akin to a zero-sum game: a choice between increasing the number of available experts to improve performance or preserving a lower level of available experts to ensure sparsity, as depicted in Figure 1.

To mitigate this contradiction, one solution would be to use the knowledge of other experts to assist the sparsely selected experts. This is similar to multi-task learning, which transfers knowledge among related tasks. Some works (Karimi Mahabadi et al., 2021; Ivison and Peters, 2022; Zhao et al., 2023) suggest using hypernetworks (Ha et al., 2017) to generate task-specific knowledge to enhance positive transfer between tasks. Inspired by this, we aim to increase the availability of expert knowledge by transferring the knowledge of unselected experts while sparsely selecting experts.

In this paper, we propose **HyperMoE**, a novel MoE framework built upon hypernetworks, which captures the information from every expert by leveraging expert-shared hypernetwork while achieving positive expert transfer by generating conditioned modules individually. We refer to the information as *cross-expert* information. Specifically, a HyperMoE consists of HyperExperts, which are generated based on the information of unselected experts and serve as supplementary information for selected experts while maintaining sparsity.

We further improve upon this by introducing the concept of *cross-layer* Hypernetworks: A hypernetwork is shared among all transformer layers, which enables information flow among MoEs in different layers. This brings additional efficiency in terms of parameters and computational costs: Despite the additional computation, our method only experienced a decrease[1] of approximately 15% in training speed and 10% in inference speed compared to the standard MoE.

We evaluate HyperMoE on 20 representative NLP datasets across diverse tasks: sequence classification, extractive question answering, summarization, and text generation. Extensive experimental results show that HyperMoE outperforms strong baselines, including Switch Transformer (Fedus et al., 2022) with MoE architecture. This demonstrates the effectiveness of our method in transferring knowledge to experts, which increases the utilization of expert knowledge while keeping the number of experts selected at a low level.

To summarise, our core contributions are:

- We propose a novel HyperMoE architecture with HyperExpert for MoE framework, which resolves the inherent tension between maintaining sparse expert selection and ensuring sufficient expert availability within MoE.
- HyperMoE outperforms strong baselines based on Switch Transformer across a diverse set of NLP tasks, confirming our approach's effectiveness.
- We show the relevance between selection embeddings, which are based on the context of unselected experts, and selected experts, indicating that the selection embeddings effectively encode the information of knowledge that the currently selected experts need.

2 Background

2.1 Mixture of Expert

A Mixture of Experts (MoE) typically consists of two parts: the gate model G and a set of expert models $E_1, E_2, \cdots, E_N$. The gate model is used to dynamically select and combine the outputs of the expert models based on the input x. As a result, each input will be determined by the collective participation of multiple expert models to obtain the output y:

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

The gate model $G(\cdot)$ is a Noisy Top-K Network (Shazeer et al., 2017) with parameters W_g and W_{noise}. This gating method introduces adjustable noise and then retains the top-k values as the final output:

$$\begin{aligned} G(x) = \mathrm{TopK}\left(\mathrm{Softmax}\left(xW_g\right.\right. \\ \left.\left. +\mathcal{N}(0,1)\,\mathrm{Softplus}\left(xW_{\text{noise}}\right)\right)\right), \end{aligned} \tag{2}$$

where $\mathrm{TopK}(\cdot)$ denotes selecting the largest K elements.

MoE allows for flexible adjustment of the contribution of expert models in different input scenarios, thereby improving the overall performance and adaptability of the model.

[1]The degree of decline in speed is related to the scale of the Hypernetworks and the bottleneck size in the generated HyperExpert (similar to r in LoRA). For various tasks, these hyperparameters can be dynamically adjusted to control the delay.

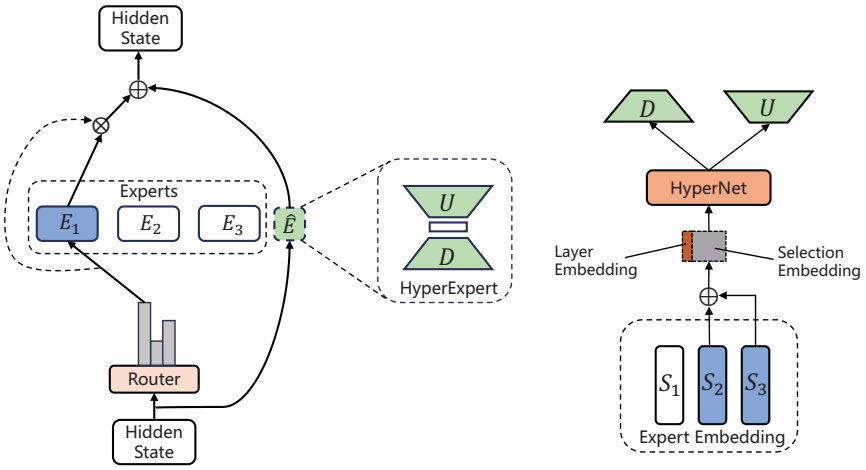

Figure 2: Overview of HyperMoE, with a case of one expert is selected. HyperExperts generated from the shared hypernetwork benefit from the cross-expert knowledge within it. Conditional inputs can enhance positive transfer between experts, generating independent modules containing knowledge relevant to the current expert. Taking the figure as an example, the selection embedding is obtained by aggregating unselected experts $S_{2,3}$'s embeddings. This selection embedding is input into a hypernetwork, which is shared across all experts and all layers, to generate a specific HyperExpert $\hat{E}$ that participates in the computation along with the selected experts E_1. The experts E_2 and E_3 are not activated throughout the process.

2.2 HyperNetworks

Hypernetwork (Ha et al., 2017) can generate parameters to be used by target networks or modules. Specifically, a hypernetwork with independent parameters ϕ denoted as h_ϕ, leverages an context information z to generate the target parameters θ for the primary network f_θ and the primary network with an input x is redefined as:

$$\text{output} = f_\theta(x) = f_{h_\phi(z)}(x). \tag{3}$$

This method of flexibly adjusting the parameters of the target network to adapt to different input scenarios is widely used in multi-task learning (Karimi Mahabadi et al., 2021; Üstün et al., 2022) and few-shot learning (Ponti et al., 2021). While generating condition-specific parameters, these parameters also benefit from shared knowledge (Pfeiffer et al., 2023).

3 Method

Overview. Taking inspiration from knowledge transferring between different tasks in multi-task learning, we propose HyperMoE. The key idea of HyperMoE is to enhance the availability of knowledge for the current input through positive knowledge transfer between experts. Through the condition input we designed, the relevant knowledge within the cross-expert information captured by the hypernetwork is encoded into HyperExpert, serving as supplementary information for the currently selected experts. In this work, we introduce conditional expert, in which we use shared hypernetworks to generate the expert weights based on the information of the unselected experts. The hypernetworks capture information across experts and transfer relevant knowledge to the selected experts by conditional generation.

3.1 Conditional Expert

In the transformer model based on the MoE structure, the experts $E_1, E_2, \cdots, E_N$ in MoE are typically denoted as a group of parallel FFNs. For an input $x \in \mathbb{R}^h$, the output $y \in \mathbb{R}^h$ can be calculated by the FFN layer as follows:

$$y = \text{FFN(x)} = \sigma(xW_1)W_2, \tag{4}$$

where $W_1 \in \mathbb{R}^{h\times b}$ and $W_2 \in \mathbb{R}^{b\times h}$ are weight matrices. $\sigma(\cdot)$ denotes a non-linear activation function.

In our approach, the matrices W_1 and W_2 are generated by a hypernetwork as described in Section 3.2. In addition, we adopt a bottleneck structure for the conditional expert to improve parameter efficiency inspired by the Adapter (Houlsby et al., 2019). Specifically, the bottleneck dimension b satisfies $b \ll h$ in our method.

3.2 HyperExpert

These works (Karimi Mahabadi et al., 2021; He et al., 2022; Phang et al., 2023; Ivison et al., 2023)

indicate that hypernetworks can learn the parameter information of the main neural network under different input scenarios and efficiently adjust the parameters of the target network to adapt to this information.

Consequently, we propose a novel design called HyperExpert, which captures beneficial knowledge from cross-expert through conditional generation to serve as auxiliary information for the selected experts involved in the computation, as shown in Figure 2. This also results in the extra parameters increasing sub-linearly with the number of layers, enhancing the parameter efficiency of the model.

Selection Embedding. We define the selection embedding to encode the information of experts not selected for each token. Let $p_i \in \mathbb{R}^t$ denote the selection embedding for i-th token and t denotes the dimension. To calculate the selection embedding efficiently and achieve better generalization, we introduce a group of expert embedding $\{S_m\}_{m=1}^N$, where $S_m \in \mathbb{R}^{t'}$ represents the m-th expert out of N experts. The computation process is as follows:

$$\hat{Z}_i = I - Z_i = I - G(x_i), \tag{5}$$

$$p_i = \mathrm{MLP}(\sum_{j=1}^{N} S_j \frac{z_{i,j}}{\sum_{j=1}^{N} z_{i,j}}), \tag{6}$$

where $G(\cdot)$ denotes Noisy Top-K Network as described in Section 2.1. The vector $Z_i \in \mathbb{R}^{|N|}$ represents token-expert allocations: each element $z_{i,j}$ is a binary scalar indicating if the expert embedding S_j is active for the input token x_i. I is an identity vector. $\mathrm{MLP}(\cdot)$ is consisting of two feed-forward layers and a ReLU non-linearity.

HyperExpert. We use a hypernetwork $H_e(\cdot)$ to construct HyperExpert $\hat{E}$ based on the conditional information of the unselected experts. To better share information across different layers and improve parameter efficiency, we **share the hypernetwork among all layers**. Additionally, we define the layer embeddings $l_\tau \in \mathbb{R}^{t'}$ for the τ-th Transformer layer. After that, we feed a concatenation of selection embedding and layer embedding to a project network to acquire final embedding $k_\tau^i = h(p_i, l_\tau)$, which is the input to hypernetwork $H_e(\cdot)$ to generates the weight matrices D_i^τ and W_i^τ for HyperExpert:

$$(D_i^\tau, U_i^\tau) = H_e(k_\tau^i) = (W^D, W^U)k_\tau^i. \tag{7}$$

The weight matrices of hypernetworks $W^{D/U}$ are used to generate the down-projection matrix $D_i^\tau \in \mathbb{R}^{h \times b}$ and the up-projection matrix $U_i^\tau \in \mathbb{R}^{b \times h}$ in the HyperExpert $\hat{E}_i$ for i-th token at τ-th transformer block.

Finally, we insert HyperExpert into the expert layer of MoE in parallel and calculate the output of i-th token as follows:

$$\hat{E}_i(x_i) = \mathrm{Relu}(D_i^\tau x)U_i^\tau, \tag{8}$$

$$y_i = \sum_{r=1}^{N} G(x_i)E_r(x_i) + \hat{E}_i(x_i). \tag{9}$$

In this way, the hypernetwork acts as an information capturer across experts, while the selection embeddings efficiently extract knowledge of experts suitable for the current token selection from the hypernetwork and generate HyperExpert to reduce the transfer of negative knowledge in cross-expert information.

4 Experiments

4.1 Datasets

We evaluate HyperMoE on 20 NLP datasets across diverse tasks including sequence classification, question answering, summarization, and text generation. GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benchmarks are widely used evaluation datasets for assessing natural language understanding capabilities. Both of them are a collection of text classification tasks: sentence similarity (STS-B; Cer et al., 2017), (MRPC; Dolan and Brockett, 2005), (QQP; Wang et al., 2018), question-answering (BoolQ; Clark et al., 2019), (MultiRC; Khashabi et al., 2018), (RECORD; Zhang et al., 2018), sentiment analysis (SST-2; Socher et al., 2013), sentence acceptability (CoLA; Warstadt et al., 2019), natural language inference (MNLI; Williams et al., 2018), (QNLI; Demszky et al., 2018), (RTE; Giampiccolo et al., 2007), (CB; De Marneffe et al., 2019), word sense disambiguation (WIC; Pilehvar and Camacho-Collados, 2019), coreference resolution (WSC; Levesque et al., 2012) and sentence completion (COPA; Roemmele et al., 2011). For the question-answering task, we consider SQuAD v1.1 (Rajpurkar et al., 2016), a collection of question-answer pairs derived from Wikipedia articles, with each answer being a text span from the corresponding reading passage. For the summarization task, we use Xsum (Narayan et al., 2018) and CNN/Daily Mail(CNNDM) (Hermann et al., 2015) to test the model's ability to summarize articles.

GLUE									
Method	**CoLA**	**SST-2**	**STS-B**	**MRPC**	**QQP**	**MNLI**	**QNLI**	**RTE**	**Avg**
MoE	54.24	93.81	**88.69**	87.90	**90.58**	87.93	91.68	67.35	82.77
MoE-Share	53.98	94.27	88.38	89.21	90.51	87.95	92.25	**67.52**	83.01
HyperMoE (ours)	**54.67**	**94.38**	88.68	**89.63**	90.52	**88.43**	**92.64**	67.01	**83.25**
SuperGLUE									
Method	**BoolQ**	**CB**	**MultiRC**	**COPA**	**ReCoRD**	**RTE**	**WIC**	**WSC**	**Avg**
MoE	72.69	69.64	66.38	45.00	71.26	67.15	63.63	56.58	64.04
MoE-Share	72.11	67.85	66.71	45.00	71.91	**67.87**	**65.36**	**56.84**	64.21
HyperMoE (ours)	**73.14**	**69.68**	**67.68**	45.00	**74.06**	67.67	65.31	56.53	**64.88**

Table 1: Overall comparison on GLUE and SuperGLUE. Switch Transformer-base-8 is used as the PLM backbone of all methods. For STS-B, we report Pearson Correlation. For MultiRC, we report F1. For ReCoRD, we report Exact Match. For CoLA, we report Matthews correlation. For other tasks, we report accuracy. The best result on each block is in **bold**.

And finally, the WikiText-2 dataset (Merity et al., 2017) is used to measure the ability of long-range dependencies generation.

4.2 Experiments Details

Following (He et al., 2023), we fine-tune pre-trained MoE models on downstream tasks and report results from the last checkpoint. Unless otherwise specified, Our base model primarily uses Switch Transformer-base-8, which is an MoE model built on T5-base (Raffel et al., 2020) with 8 available experts, having a total number of parameters of 620M. For the WikiText dataset, we employ GPT-2 small (Radford et al.) as the base model and expand it into the MoE structure by duplicating the weights of the feed-forward layer. In addition, we also use Switch Transformer-base-16/32 to explore the effect of expert numbers on our method. To achieve a fair comparison, all methods in our paper employ the same Top-1 routing and auxiliary loss. For different data scales, we grid-search the training epoch and batch size from {10, 15, 20}, and {8, 16, 32, 64}, respectively. The learning rate is grid-search from {1e-5, 5e-5, 1e-4, 5e-4} with Adam optimizer and the first 10% warm-up steps. We set the maximum token length to 1024 for WikiText datasets, 348 for SQuAD, and 256 for all other datasets except for the summarization task. For Xsum and CNNDM, we set the max length of source articles to be 1024 and the max length of the target summary to be 128. As for All experiments run for 3 times with different seeds and we report the average for each result.

4.3 Baselines

Our approach is built upon Switch Transformer (Fedus et al., 2022), a well-known MoE model using Top-1 routing. Consequently, we primarily compare our approach with the following baselines: (1) **MoE**, fully finetuning switch transformer model. (2) **MoE-Share**, as it is a relevant baseline that does not exploit the inductive bias of the relationship between selected and unselected experts in the process of computation: add an MLP network that is shared among all experts in the MoE layer of a switch transformer, which has the same size as the experts in MoE.

4.4 Results and Analysis

4.4.1 Main Results

GLUE and SuperGLUE. Table 1 shows the results of various methods applied to the tasks within GLUE and SuperGLUE. Overall, our method improves significantly compared to both MoE and MoE-Share. Specifically, compared to MoE, our method shows a +0.48% and +0.84% increase on the GLUE and SuperGLUE benchmarks, respectively. This enhancement underscores the advantage of adopting expert knowledge transfer in improving the performance of MoE models. It's noteworthy that MoE-Share is relevant to ours, but performs worse than MoE on certain datasets such as STS-B, CoLA, and BoolQ. A potential reason is that the cross-expert information captured through a shared network cannot achieve effective positive transfer, adversely impacting MoE-Share's effectiveness on these datasets. In contrast, our method maintains a lead on these datasets while also performing well on most other datasets. This under-

Method	Sum.Task		QA.Task	Modeling.Task
	XSum	CNNDM	SQuAD	WikiText
MoE	19.35	19.75	83.01	21.71
MoE-Share	19.41	19.80	82.87	21.63
HyperMoE	**19.67**	**20.12**	**83.51**	**21.49**

Table 2: Overall comparison on Xsum, CNNDM, SQuAD, WikiText. For Xsum and CNNDM, we report the Rouge-2 metric (↑). For SQuAD, we report the Exact Match metric (↑). For WikiText, we report the Perplexity metric (↓). All tasks are conducted on the Switch Transformer, except for WikiText, which is carried out on Bert with an MoE structure, as detailed in Section 4.2.

scores the effectiveness of our conditional generation strategy: selectively transferring knowledge by leveraging expert selection information during the computation process.

Other Tasks. Table 2 displays the performance of various methods across question-answering tasks, summarization tasks, and text-generation tasks. In addition to achieving outstanding performance on Natural Language Understanding (NLU) tasks represented by GLUE and SuperGLUE, our method also excels in Natural Language Generation (NLG) tasks. Experimental results show that our method outperforms baseline methods across all NLG tasks. Specifically, in extractive question-answering tasks, our method shows improvements of 0.50% and 0.64% over MoE and MoE-Share, respectively. Like the NLU tasks, MoE-Share again underperforms, indicating that the extra networks may not effectively learn information useful to experts without the expert selection inductive bias. Furthermore, our method still performs well in summarization tasks involving long-text inputs. This demonstrates that our method can still effectively enhance the availability of expert knowledge through knowledge transfer under complex input conditions, suitable for tasks of various text lengths. Lastly, our method also achieves considerable improvement on Wikitext. These results demonstrate the effectiveness of HyperMoE in various tasks.

4.4.2 Ablation Study

We conduct an ablation study on the SQuAD to evaluate the effectiveness of the proposed modules. The embedding design is removed to verify the effect of using external information as embeddings. As shown in Table 3 (row 1), when the embedding and hypernet are removed, our method is equivalent to MoE. Table 3 (row 2) omits the embedding

Embedding	Hypernet	Exact Match
✗	✗	83.01
✗	✓	82.92
W	✓	83.33
P	✓	**83.51**

Table 3: Ablation study on SQuAD. **W** represents the use of expert weights as embeddings. **P** denotes the use of our proposed selection embedding.

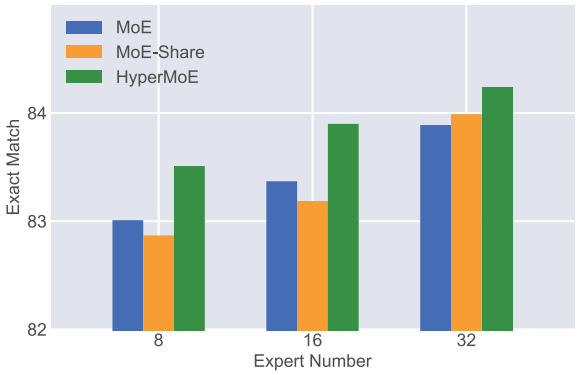

Figure 3: Performance comparison of MoE methods on the SQuAD dataset with the increase in the number of experts.

design, directly using the sample's hidden state as input to the hypernetwork. This results in a marked decrease in performance, even falling below that of MoE. This suggests that conditioning the hypernetwork on the sample enlarges the parameter search space and is difficult to optimize. In an additional experiment, we use a depthwise separable convolutional network (Howard et al., 2017) with kernels of sizes 5×5 and 3×3 to compress and reduce the dimensions of the experts' weights, obtaining expert embeddings. More details are in Appendix A. The selection embeddings are then computed and input into the hypernetwork as described in Section 3.2. Empirically, expert weights can better represent the information of experts. However, as shown in Table 3 (row 3), this strategy leads to a slight drop in performance, defying expectations. A potential explanation is the substantial information loss associated with compressing expert weights, resulting in a loss of specific information details. We leave the exploration of this strategy to future work.

4.4.3 Performance in Scaling the Number of Experts

To explore the impact of the variation in the number of experts on our method, we fine-tuned on the

SQuAD dataset using Switch Transformer-base-16/32 as pre-trained models. These models possess 16 and 32 experts in each MoE layer, respectively. As demonstrated in Figure 3, every method achieves performance enhancement across models featuring a diverse number of experts. Notably, our method exhibits consistent superior growth and outperforms the others. This indicates that the proposed conditional generation strategy can still effectively benefit from knowledge transfer as the number of experts increases.

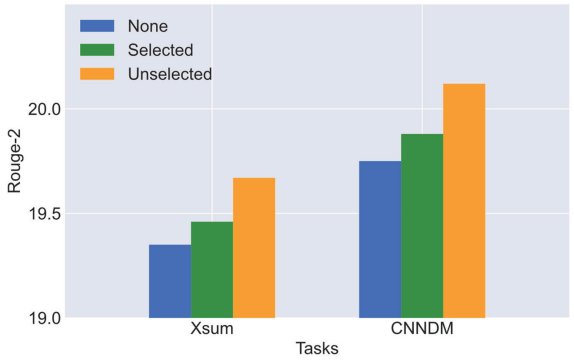

Figure 4: Compare the performance of our method when calculating selection embedding using the selected expert embeddings or the unselected expert embeddings, respectively.

4.4.4 Investigating of Selection Embedding.

The unselected expert embeddings are more informative than selected expert embeddings. Empirically, by conditioning on the information of unselected experts, specific knowledge can be extracted from cross-expert knowledge, which selected experts do not possess, thereby aiding the selected experts. To verify this idea, we input embeddings of both selected and unselected experts into a hypernetwork, comparing their performance on the Xsum and CNNDM datasets. As shown in Figure 4, using unselected expert information as conditional input can achieve comparable results. This implies that the conditional information of unselected experts can generate more beneficial knowledge for the selected experts through a shared hypernetwork.

Expert embeddings and the selection embeddings have a corresponding relationship. In addition, to explore whether the embeddings encode the information in our proposed method, we provide visualizations of the expert embeddings and computed selection embeddings within the final MoE layer of Switch Transformer-base-8 learned on CNNDM. Figure 5 reveals that both sets of embeddings exhibit sparse distributions, suggesting that the embeddings encode some specific non-relevant information. We also observe a correlation between the distances among selection embeddings and the distances among expert embeddings, such as between 4-5-6, 1-2-8, 7-3. This correlation implies that the information of the unselected experts encoded by the selection embeddings depends on the information of the selected experts, further illustrating that the selection embeddings effectively capture the information of the knowledge the currently selected experts need.

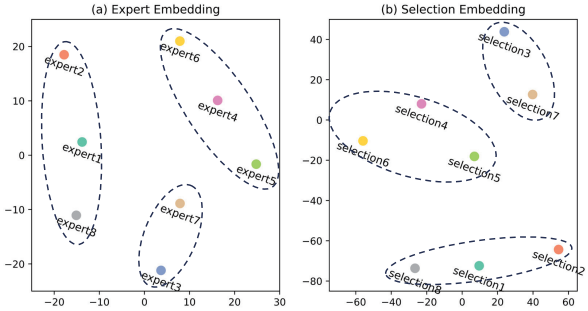

Figure 5: t-SNE visualizations for expert embeddings (right) and selection embeddings (left). selection i denotes calculated using all expert embeddings except for the i-th expert embedding.

4.4.5 Impact of Additional Computation

Although our method achieves significant performance improvements compared to the original MoE structure, it introduces additional networks, which inevitably slightly reduces the inference speed of HyperMoE. We evaluate the number of samples per second that our method can train/infer based on Switch Transformer-base-8. The methods in each task employ the same batch size. As shown in Table 4, our method's training/inference speed is only reduced by about 15% and 10% compared to MoE, respectively. This suggests that our approach can enhance the availability of expert knowledge more effectively without significantly increasing computational costs while maintaining sparsity during expert selection.

5 Related Work

5.1 Mixture of Expert

Shazeer et al. (2017) introduces Mixture-of-Expert layers for LSTM language modeling and machine translation. These layers are inserted between the standard layers of the LSTM model. Subsequent

Method	Sum.Task		QA.Task
	XSum	**CNNDM**	**SQuAD**
MoE_{train}	76.31	77.20	89.56
HyperMoE_{train}	65.69	65.95	75.38
MoE_{eval}	5.42	5.99	78.43
HyperMoE_{eval}	4.78	5.32	67.73

Table 4: The number of samples trained/evaluated per second.

work primarily builds on Transformers, where expert layers often replace dense layers. Lepikhin et al. (2021) first introduces MoE layers into Transformers and studies them in the context of machine translation. With the release of Gshard (Lepikhin et al., 2021) and Switch Transformer (Fedus et al., 2022), MoE models are scaled up to new heights by introducing thousands of small-scale experts. In terms of routing, Shazeer et al. (2017) use routing to the top k experts out of k > 1. Hazimeh et al. (2021) propose DSelect-k, a smoothed version of the top-k routing algorithm that improves upon standard top-k routing. Fedus et al. (2022), Clark et al. (2022) and Xue et al. (2023) demonstrate that top-1 routing can also achieve competitive results. Hash Layer (Roller et al., 2021b) and StableMoE (Dai et al., 2022) employ fixed routing strategies for more stable routing and training. Zhou et al. (2022) propose an expert selection routing strategy where each token can be assigned to a different number of experts. Rajbhandari et al. (2022) and Dai et al. (2024) isolate general knowledge from experts using shared experts from engineering and algorithm perspectives, respectively, to promote expert specialization.

In contrast to previous work, our work mainly focuses on the knowledge transfer between experts in MoE. This provides a solution for improving the availability of expert knowledge in MoE while maintaining sparsity.

5.2 HyperNetwork

Hypernetworks (Ha et al., 2017) are widely used in multi-task learning due to their ability to avoid negative interference of corresponding modules by soft parameter sharing and generating module parameters conditioned on the shared parameters. The most common approach usually takes task (Karimi Mahabadi et al., 2021; Zhao et al., 2023) or language embeddings (Üstün et al., 2020; Baziotis et al., 2022) as contextual information to generate corresponding module parameters, such as adapter layers (Üstün et al., 2020; Ansell et al., 2021; Karimi Mahabadi et al., 2021), classifier heads (Ponti et al., 2021), and continuous prompts (He et al., 2022). In addition, hypernetwork-based approaches have also been very successful in zero-shot and few-shot scenarios (Deb et al., 2022; Phang et al., 2023; Ivison et al., 2023).

In the field of NLP, hypernetworks are mainly used to improve the generalization (Volk et al., 2022; Zhang et al., 2023) and applicability (Wullach et al., 2022; He et al., 2022; Tan et al., 2023) of dense models. Our work explores the integration of hypernetworks with sparse MoE. We propose to input the expert selection status of tokens as information into the hypernetwork and generate module parameters that correspond to the respective tokens. To the best of our knowledge, this is the first time that hypernetworks have been introduced in the MoE structure, which extends the application scope of hypernetworks and provides new insights for knowledge transferring in MoE.

6 Conclusion

In this work, we introduce HyperMoE, a novel Mixture of Experts (MoE) architecture. Inspired by the concept of knowledge transfer in multi-task learning, we propose a method to facilitate knowledge transfer between experts through conditional generation. Our method enhances expert knowledge availability while maintaining expert selection's sparsity. We show the effectiveness of our approach across a wide range of NLP tasks. Experimental results demonstrate that our method exhibits excellent performance compared to the conventional MoE. Furthermore, our analysis shows that without any measures, there could be negative knowledge transfer across experts when transferring knowledge to specific experts. Our approach mitigates this issue by capturing the contextual information of experts. We explore the feasibility of knowledge transfer between experts in MoE, providing a new perspective for future improvements in MoE architectures.

Limitations

Despite our work has demonstrated strong experimental results, there are several limitations: (1) In this work, we utilize end-to-end training to learn expert embeddings. Incorporating prior knowledge,

such as expert weights, into the embedding learning process may improve efficiency and performance. We will improve upon this in future work. (2) We insert HyperExpert into the expert layer of MoE in parallel. This incurs additional computational overhead. Mitigating this issue could be achieved by employing some parameter-efficient methods (such as LoRA (Hu et al., 2022) and $(IA)^3$ (Liu et al., 2022)) to insert HyperExpert into MoE. (3) Current experiments mainly focus on fine-tuning the pre-trained MoE model. Utilizing our proposed method to train a large-scale MoE from scratch will be the emphasis of our future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. Mad-g: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giridharan Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeffrey Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Veselin Stoyanov. 2022. Efficient large scale language modeling with mixtures of experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11699–11732, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Christos Baziotis, Mikel Artetxe, James Cross, and Shruti Bhosale. 2022. Multilingual machine translation with hyper-adapters. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1170–1185.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Aidan Clark, Diego De Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, George Bm Van Den Driessche, Eliza Rutherford, Tom Hennigan, Matthew J Johnson, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Marc'Aurelio Ranzato, Jack Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. 2022. Unified scaling laws for routed language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4057–4086. PMLR.

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. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Stablemoe: Stable routing strategy for mixture of experts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7085–7095.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Budhaditya Deb, Ahmed Hassan, and Guoqing Zheng. 2022. Boosting natural language generation from instructions with meta-learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6792–6808.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

David Ha, Andrew M. Dai, and Quoc V. Le. 2017. Hypernetworks. In *International Conference on Learning Representations*.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. 2021. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34:29335–29347.

Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. 2023. Merging experts into one: Improving computational efficiency of mixture of experts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14685–14691, Singapore. Association for Computational Linguistics.

Yun He, Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, et al. 2022. Hyperprompt: Prompt-based task-conditioning of transformers. In *International Conference on Machine Learning*, pages 8678–8690. PMLR.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hamish Ivison, Akshita Bhagia, Yizhong Wang, Hannaneh Hajishirzi, and Matthew E Peters. 2023. Hint: Hypernetwork instruction tuning for efficient zero- and few-shot generalisation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11272–11288.

Hamish Ivison and Matthew Peters. 2022. Hyperdecoders: Instance-specific decoders for multi-task NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1715–1730, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576, Online. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. 2023. Modular deep learning. *arXiv preprint arXiv:2302.11529*.

Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. 2023. HyperTuning: Toward adapting large language models without back-propagation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27854–27875. PMLR.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Edoardo M. Ponti, Ivan Vulić, Ryan Cotterell, Marinela Parovic, Roi Reichart, and Anna Korhonen. 2021. Parameter space factorization for zero-shot learning across tasks and languages. *Transactions of the Association for Computational Linguistics*, 9:410–428.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18332–18346. PMLR.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

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, arthur szlam, and Jason Weston. 2021a. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems*, volume 34, pages 17555–17566. Curran Associates, Inc.

Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, et al. 2021b. 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. In *International Conference on Learning Representations*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. Udapter: Language adaptation for truly universal dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, Gertjan van Noord, and Sebastian Ruder. 2022. Hyper-X: A unified hypernetwork for multi-task multilingual transfer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7934–7949, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tomer Volk, Eyal Ben-David, Ohad Amosy, Gal Chechik, and Roi Reichart. 2022. Example-based hypernetworks for out-of-distribution generalization. *arXiv preprint arXiv:2203.14276*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier

benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Tomer Wullach, Amir Adler, and Einat Minkov. 2022. Character-level hypernetworks for hate speech detection. *Expert Systems with Applications*, 205:117571.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. 2023. Openmoe: An early effort on open mixture-of-experts language models. *preprint*.

Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. 2019. Condconv: Conditionally parameterized convolutions for efficient inference. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Liang Zhang, Chulun Zhou, Fandong Meng, Jinsong Su, Yidong Chen, and Jie Zhou. 2023. Hypernetwork-based decoupling to improve model generalization for few-shot relation extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6213–6223.

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. *arXiv preprint 1810.12885*.

Hao Zhao, Jie Fu, and Zhaofeng He. 2023. Prototype-based HyperAdapter for sample-efficient multi-task tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4603–4615, Singapore. Association for Computational Linguistics.

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

A Depthwise Separable Convolutional Networks Details

For every expert weight in each MoE layer, we use the same convolutional network to reduce its dimensionality. First, we stack them so that their dimensional form is three-dimensional, similar to images. Then, we perform convolution on them. Our experiments used depthwise separable convolutions, with specific parameters as shown in Table 5.

Type/Stride	Filter Shape	Input Size
Conv dw / s5	$5 \times 5 \times 2$ dw	$8 \times 2 \times 3072 \times 768$
Conv / s1	$1 \times 1 \times 2 \times 32$	$8 \times 2 \times 614 \times 153$
Avg Pool / s(16, 6)	Pool(16,6)	$8 \times 32 \times 614 \times 153$
Conv dw / s3	$3 \times 3 \times 32$ dw	$8 \times 32 \times 38 \times 25$
Conv / s1	$1 \times 1 \times 32 \times 128$	$8 \times 32 \times 12 \times 8$
Avg Pool / s8	Pool(8,8)	$8 \times 128 \times 12 \times 8$
Output	–	$8 \times 128 \times 1 \times 1$

Table 5: Specific parameters and structure of depthwise separable convolutions.

The compressed expert weights are used as expert embeddings in subsequent computations as described in Section 3.2.