FROM DENSE TO DYNAMIC: TOKEN-DIFFICULTY DRIVEN MOEFICATION OF PRE-TRAINED LLMS

Kumari Nishu, Sachin Mehta,[*] Samira Abnar, Mehrdad Farajtabar,

Maxwell Horton, Mahyar Najibi, Moin Nabi, Minsik Cho, Devang Naik
Apple

ABSTRACT

Training large language models (LLMs) for different inference constraints is computationally expensive, limiting control over efficiency-accuracy trade-offs. Moreover, once trained, these models typically process tokens uniformly, regardless of their complexity, leading to static and inflexible behavior. In this paper, we introduce a post-training optimization framework, $\mathrm{DynaMoE}$, that adapts a pre-trained dense LLM to a token-difficulty-driven Mixture-of-Experts model with minimal fine-tuning cost. This adaptation makes the model dynamic, with sensitivity control to customize the balance between efficiency and accuracy. $\mathrm{DynaMoE}$ features a token-difficulty-aware router that predicts the difficulty of tokens and directs them to the appropriate sub-networks or experts, enabling larger experts to handle more complex tokens and smaller experts to process simpler ones. Our experiments demonstrate that $\mathrm{DynaMoE}$ can generate a range of adaptive model variants with a single fine-tuning step, utilizing only $5B$ tokens, a minimal cost compared to the base model's training. Each variant offers distinct trade-offs between accuracy and performance.

1 INTRODUCTION

Large language models (LLMs) have significantly advanced the field of natural language processing, showcasing strong capabilities in addressing complex tasks Brown et al. (2020); Touvron et al. (2023a); Wei et al. (2022). However, their large size presents challenges, particularly in terms of high memory and computational demands, which can limit their deployment in resource-constrained settings. To address this, LLMs must be optimized for specific memory and computational constraints Touvron et al. (2023b). However, designing multi-billion-parameter models for every use case is not cost-effective, as it demands substantial training time, data, and resources.

Mixture-of-Experts (MoE) models (Shazeer et al., 2017; Du et al., 2021; Fedus et al., 2022; Zoph et al., 2022; He, 2024) have emerged as a promising alternative to dense models, offering improved efficiency by sparsely activating select sub-modules or experts. However, training MoEs from scratch remains resource-intensive and each expert becomes static, often requiring fixed compute budget irrespective of the input complexity.

Flextron (Cai et al., 2024) explored a post-training methodology by integrating the MoE concept into a nested elastic structure within the MLP layers, creating heterogeneous experts of different sizes, selected by a router conditioned on the input data. However, the lack of supervision in the router training leads to sub-optimal input complexity adaptation. Salehi et al. (2023) proposed an input-adaptive approach that predicts the difficulty of input data and dynamically adjusts the network's width accordingly. In the absence of ground-truth difficulty labels, they relied on heuristic methods for label generation, which may limit precision and consistency in difficulty estimation.

To address their shortcomings, we introduce $\mathrm{DynaMoE}$, a post-training optimization framework designed to transform a dense LLM into a token-difficulty-driven MoE model. $\mathrm{DynaMoE}$ leverages the insight that not all tokens require the full capacity of a model's weights. For example, in the

[*]Contributed when employed by Apple.

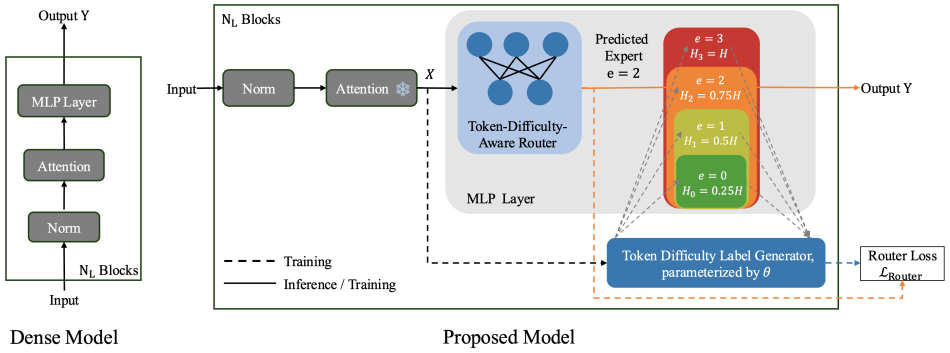


Figure 1: Overview of our proposed post-training optimization framework, DynaMoE. The left part represents the base pre-trained LLM, while the right part shows the adapted DynaMoE model.

sentence "Geoffrey did his PhD at the university of Edinburgh", simpler tokens like "at the university of" are predictable using prior context, while more complex tokens like "Edinburgh" demand broader contextual understanding. To maximize efficiency, DynaMoE selectively activates nested sub-components of the MLP, referred as experts, based on the predicted difficulty of each token. To this end, we make the following contributions:

- The framework includes a novel token-difficulty-aware router, trained to predict token hardness and assign it to the appropriate expert dynamically.
- Due to the lack of ground truth notion of hardness, we introduce a method to derive token difficulty labels which serve as supervision signals for training the router. This approach allows a token to have varying difficulty labels across different layers.
- A post-training optimization framework, DynaMoE, to easily adapt a pre-trained dense LLM to a token-difficulty-driven MoE model, featuring a sensitivity parameter to customize the efficiency vs accuracy trade-off.

2 METHOD

In this section, we describe our proposed post-training optimization framework, DynaMoE, which transforms a dense LLM into an MoE model for adaptive inference based on token difficulty. The process involves three key steps, detailed in the below sub-sections.

2.1 DEFINING HETEROGENEOUS EXPERTS

In this work, we focus on defining experts into the MLP layers of the LLM Devvrit et al. (2023), as these layers account for the majority of the compute and operate on a token-by-token basis. The overview of DynaMoE is depicted in Fig. 1. The left part of the figure denotes the base pre-trained model and the right part shows the adapted DynaMoE model, where the original single MLP layer is transformed into multiple nested FFN blocks or experts. Such expert formation introduces no additional parameters to the base model, aside from the router. This design draws inspiration from adaptive width reduction in transformer Salehi et al. (2023) and recent works like Matformer Devvrit et al. (2023) and Flextron Cai et al. (2024).

Let D and H denote the embedding and the hidden dimensions of the MLP layer respectively. The input to the MLP layer is $\boldsymbol{X} \in \mathbb{R}^{B\times D}$ and the output is $\boldsymbol{Y} \in \mathbb{R}^{B\times D}$, where B is the batch dimension. The MLP layer with two fully connected layers is represented by weight matrices $\boldsymbol{W}^{(IN)} \in \mathbb{R}^{H\times D}$ and $\boldsymbol{W}^{(OUT)} \in \mathbb{R}^{D\times H}$. In order to get best results, we first rearrange these fully-connected layers, $\boldsymbol{W}^{(IN)}$ and $\boldsymbol{W}^{(OUT)}$, to have the most important rows/columns in the beginning of the matrix so that they can be included in all of the experts Samragh et al. (2023). There are a total of E experts indexed using $e \in \{0, 1, \ldots, E-1\}$. Each expert gets a portion H_e of the weight matrices $\boldsymbol{W}^{(IN)}$ and $\boldsymbol{W}^{(OUT)}$, sliced over the hidden dimension H. The value H_e is obtained as a fraction of H as, $H_e = \left\lfloor \left(\frac{e+1}{E}\right) \cdot H \right\rfloor$, consequently, $H_0 < H_1 < \cdots < H_{E-1}$ and $H_{E-1} = H$. The

	Cost (#Tokens)	Params	ARC-e	LAMBADA	PIQA	WinoGrande	Avg4	SciQ	HellaSwag	ARC-c	Avg7
Base Mistral 7B	-	7B	80.2	75.1	80.8	75.5	**77.8**	96.4	61.4	50.5	74.2
DynaMoE $\theta = 0.9$	5B	6B	76.2	71.0	78.8	72.1	**74.5**	95.7	56.9	43.7	70.6
DynaMoE $\theta = 0.8$	5B	5B	68.3	66.5	76.6	66.2	**69.4**	94.4	53.4	34.4	65.7
Base Llama2-7B †	-	6.5B	75.1	71.5	77.5	69.1	73.3				
Flextron †	93.57B	4.1B	68.6	65.1	76.1	63.7	68.3				

Table 1: Evaluation of DynaMoE models with different sensitivity factor θ on downstream tasks, using zero-shot accuracy metric. Our base model is Mistral 7B (Jiang et al., 2023). (†): results from Flextron (Cai et al., 2024) used as our baseline. *Params* denotes the average number of total activated parameters, aggregated over the downstream tasks. *Avg4* averages over *ARC-e, LAMBDA, PIQA, WinoGrande*, while *Avg7* averages over all tasks.

restriction of the matrices $\boldsymbol{W}^{(IN)}$ and $\boldsymbol{W}^{(OUT)}$ to the expert width H_e is obtained using the slicing operator that selects the first H_e rows and columns respectively as $\boldsymbol{W}_e^{(IN)} = \boldsymbol{W}^{(IN)}[0 : H_e, :]$ and $\boldsymbol{W}_e^{(OUT)} = \boldsymbol{W}^{(OUT)}[:, 0 : H_e]$. With σ as the activation function, the output $\boldsymbol{Y}_e$ of the MLP layer corresponding to the expert e can thus be obtained as, $\boldsymbol{Y}_e = \sigma\left(\boldsymbol{X} \cdot \left(\boldsymbol{W}_e^{(IN)}\right)^T\right) \cdot \left(\boldsymbol{W}_e^{(OUT)}\right)^T$.

2.2 Generating Token Difficulty Label

We aim to train a token-difficulty-aware router to dynamically assign tokens to an appropriate expert. But there is no ground-truth label denoting token difficulty to train such a router. To this end, we propose a method to estimate the token difficulty and generate a derived-ground-truth difficulty label during training. First, we pass the input to all experts and generate the output $\boldsymbol{Y}_e$ for each $e \in [E]$. Then, for each token $b \in [B]$ and each expert $e \in [E]$, we compute a similarity score $S_{b,e}$ that measures how similar is the output of the expert e compared to the output of the full MLP layer $e = E - 1$ for that token. We calculate this similarity as, $S_{b,e} = \frac{\langle \boldsymbol{Y}_e[b,:], \boldsymbol{Y}_{E-1}[b,:] \rangle}{\langle \boldsymbol{Y}_{E-1}[b,:], \boldsymbol{Y}_{E-1}[b,:] \rangle}$.

Finally, we generate a derived ground-truth hardness label l_b, representing the target expert index for token b. Given a threshold θ, we assign l_b as the smallest expert index e satisfying $S_{b,e} > \theta$, that is, $l_b = \min\{e \in [E] \mid S_{b,e} > \theta\}$. We say that a token is easier if it has a smaller label l_b.

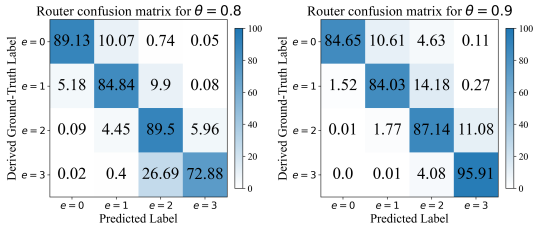
2.3 Training a Token-Difficulty-Aware Router

The output of a router is in $\mathbb{R}^{B \times E}$, denoting logits over the E experts. Each router is parameterized by two linear layers, projecting the token embedding from dimension D to U and subsequently to E. In our experiments, we use $U = 256$. We train the router using the derived labels from Section 2.2 with the cross-entropy loss. The overall objective function of DynaMoE is: $\mathcal{L} = \lambda_{LLM} \cdot \mathcal{L}_{LLM} + \lambda_{Router} \cdot \mathcal{L}_{Router}$. Here, $\mathcal{L}_{LLM}$ is the main LLM Cross-entropy loss and $\mathcal{L}_{Router}$ is the router loss. λ_{LLM} and λ_{Router} are the weights of the respective losses.

3 Experiments and Results

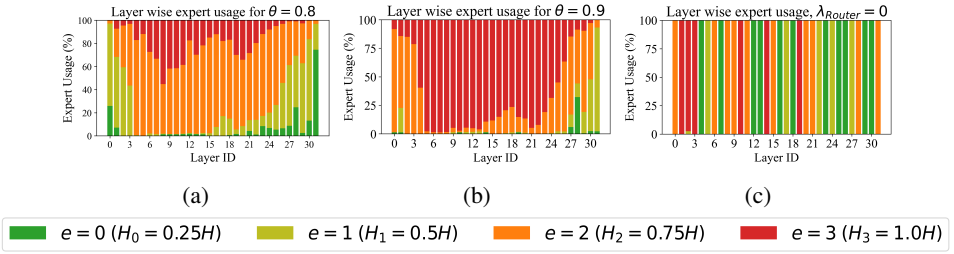
3.1 Training Details

DynaMoE integrates seamlessly with any transformer model, regardless of the architecture. We use Mistral 7B model Jiang et al. (2023), a widely-used open-source pre-trained language model. For DynaMoE fine-tuning, we use a small subset (5B tokens) of the Falcon RefinedWeb dataset Penedo et al. (2023). This minimal fine-tuning overhead enables a cost-effective conversion of any pre-trained LLM into an MoE variant for faster inference. We begin by reordering the pre-trained MLP neurons Samragh et al. (2023) based on their importance, measured by absolute activations aggregated over a small portion of the training dataset, 0.004%. We then fine-tune the DynaMoE model with 4 experts of sizes $0.25H$, $0.5H$, $0.75H$, and H respectively. The details of other hyper-parameters are given in Appendix A.



(a) (b)

Figure 2: Confusion matrix for the router's classification task in DynaMoE.



(a) (b) (c)

Figure 3: Layer-wise expert usage in DynaMoE with varying θ.

3.2 Evaluation

We evaluate the DynaMoE models on 7 downstream tasks (Appendix B) using LM Evaluation Harness Gao et al. (2024) and report the 0-shot accuracy metric in Table 1. DynaMoE is compared to two baselines, the base Mistral 7B model and the Flextron model (Cai et al., 2024) using Avg4 metric. Compared to Mistral 7B, DynaMoE improves efficiency by activating only $5B$ of $7B$ parameters on an average, with an 8.4 point accuracy drop after fine-tuning on only $5B$ tokens at $\theta = 0.8$. The number of activated parameters adapts dynamically to token difficulty. For reference, Flextron fine-tunes on $93.57B$ tokens, activating $4.1B$ of $6.5B$ parameters, with a 5 point accuracy drop from its base model, Llama2-7B Touvron et al. (2023a). We emphasize that with only $\frac{1}{18}$th of the Flextron's fine-tuning cost, our results are comparable to Flextron. Accuracy improves with increase in fine-tuning cost, but to keep the adaption lightweight, we opt for a smaller cost.

Analysis of Token-Difficulty-Aware Router: To evaluate the router's accuracy, we compare predictions from all layers against the derived ground truth labels. Fig. 2 shows the confusion matrices, which display a strong diagonal pattern, indicating high accuracy. Misclassifications mainly occurred within neighboring expert classes, highlighting the router's understanding of token difficulty.

Expert usage analysis: We visualize the expert usage patterns across all layers in Fig. 3. The parameter θ affects expert usage in DynaMoE models by controlling how quickly tokens are routed to larger experts based on difficulty. At lower θ values (e.g., $\theta = 0.8$), smaller experts ($e = 2$) dominate across layers. In contrast, at higher θ values (e.g., $\theta = 0.9$), larger experts ($e = 3$) are utilized more frequently, prioritizing accuracy over efficiency. In the absence of router loss, Fig. 3c, the model converges to using specific experts per layer instead of dynamically allocating experts based on token difficulty.

4 Conclusion

We present DynaMoE, a post-training optimization framework that converts a standard pre-trained dense LLM into a token-difficulty-driven MoE model. DynaMoE incorporates a lightweight router to predict the token difficulty and routes them to an appropriate expert. To train this router, we propose a novel method to derive the token difficulty labels, which act as supervision signals. DynaMoE generates adaptive model variants with sensitivity control, allowing customization of the trade-off between efficiency and accuracy.

REFERENCES

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:208290939.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783.

Ruisi Cai, Saurav Muralidharan, Greg Heinrich, Hongxu Yin, Zhangyang Wang, Jan Kautz, and Pavlo Molchanov. Flextron: Many-in-one flexible large language model. *ArXiv*, abs/2406.10260, 2024. URL https://api.semanticscholar.org/CorpusID:270560556.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018. URL https://api.semanticscholar.org/CorpusID:3922816.

Devvrit, Sneha Kudugunta, Aditya Kusupati, Tim Dettmers, Kaifeng Chen, Inderjit S. Dhillon, Yulia Tsvetkov, Hannaneh Hajishirzi, Sham M. Kakade, Ali Farhadi, and Prateek Jain. Matformer: Nested transformer for elastic inference. *ArXiv*, abs/2310.07707, 2023. URL https://api.semanticscholar.org/CorpusID:263834773.

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 Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Z. Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts. *ArXiv*, abs/2112.06905, 2021. URL https://api.semanticscholar.org/CorpusID:245124124.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), jan 2022. ISSN 1532-4435.

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. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Xu Owen He. Mixture of a million experts. *arXiv preprint arXiv:2407.04153*, 2024.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL https://api.semanticscholar.org/CorpusID:263830494.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. URL https://api.semanticscholar.org/CorpusID:53592270.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and R. Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *ArXiv*, abs/1606.06031, 2016. URL https://api.semanticscholar.org/CorpusID:2381275.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023. URL https://arxiv.org/abs/2306.01116.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. An adversarial winograd schema challenge at scale. 2019. URL https://api.semanticscholar.org/CorpusID:199370376.

Mohammadreza Salehi, Sachin Mehta, Aditya Kusupati, Ali Farhadi, and Hannaneh Hajishirzi. Sharcs: Efficient transformers through routing with dynamic width sub-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://api.semanticscholar.org/CorpusID:264289348.

Mohammad Samragh, Mehrdad Farajtabar, Sachin Mehta, Raviteja Vemulapalli, Fartash Faghri, Devang Naik, Oncel Tuzel, and Mohammad Rastegari. Weight subcloning: direct initialization of transformers using larger pretrained ones. *ArXiv*, abs/2312.09299, 2023. URL https://api.semanticscholar.org/CorpusID:266335424.

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. URL https://openreview.net/forum?id=B1ckMDqlg.

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. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023a. URL https://api.semanticscholar.org/CorpusID:257219404.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. 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, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023b. URL https://api.semanticscholar.org/CorpusID:259950998.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. URL https://api.semanticscholar.org/CorpusID:246411621.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. *ArXiv*, abs/1707.06209, 2017. URL https://api.semanticscholar.org/CorpusID:1553193.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:159041722.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

APPENDIX

A TRAINING HYPER-PARAMETERS

We finetune the $\mathrm{DynaMoE}$ model using only $5B$ tokens with AdamW optimizer Loshchilov & Hutter (2017) and a fixed learning rate of 10^{-5}. We keep the attention layers frozen. We set λ_{LLM} to 0.2 and λ_{Router} to 1 in the objective function for fine-tuning. We experiment with different values of threshold $\theta \in \{0.4, 0.8, 0.9\}$ to build various $\mathrm{DynaMoE}$ family of models with varying sensitivity parameter. A low sensitivity parameter, that is a smaller value of θ, makes the system less reactive, favoring smaller experts for most tokens and only escalating to bigger experts for significantly complex tokens. And a high sensitivity parameter makes the system more reactive, escalating to bigger experts even for moderately complex tokens. We use 4 experts ($E = 4$) with sizes $0.25H$, $0.5H$, $0.75H$, and H respectively. We denote the size of expert with index e as H_e.

B DOWNSTREAM TASKS

The selected evaluation tasks include ARC (Easy and Challenge) Clark et al. (2018), HellaSwag Zellers et al. (2019), PIQA Bisk et al. (2019), SciQ Welbl et al. (2017), WinoGrande Sakaguchi et al. (2019), and LAMBADA Paperno et al. (2016). Performance is measured using the zero-shot accuracy metric.