# Zoo of Experts

**Max Zuo**
Department of Computer Science
Brown University
Providence, RI 02906
zuo@brown.edu

## Abstract

Mixture of Experts (MoE) is a promising approach for scaling language models to billions or trillions of parameters in a computationally efficient manner [4, 9, 17]. More recently, MoEs have also been explored as a method for post-hoc model merging [1, 14] and finetuning [10, 15]. However, most prior work on MoEs has used homogeneous experts, with identical architectures [10, 15]. In this work, we propose a post-hoc method for mixing heterogeneous experts into a zoo of experts.

## 1 Introduction

Large language models (LLMs) have achieved impressive performance on a wide range of natural language tasks. As LLMs scale to billions or trillions of parameters, training and serving them becomes increasingly challenging. Deep Mixture of Experts (MoE) models [5] offer a promising path towards more efficient and modular LLMs, by providing a mechanism for increasing model scale while keeping the inference costs low. This is achieved by only activating a subset of the total number of parameters during inference, so while the entire model must be kept in memory, only some of an MoE's weights are used for computation.

MoE LLMs replace the standard feedforward network in the Transformer [16, 12] block with a sparsely-gated mixture of experts layer [13]. This creates a model that can support models comparable in size to dense LLMs, but routes its input into one or several smaller "experts", resulting in a lower "active" parameter count – the number of parameters used during any one pass through the model. This often leads to a slightly smaller memory footprint and faster inference while retaining performance when compared to their monolithic counterpart.

## 2 Methods

Recently, Mixture of Experts has been used as the inspiration for post-hoc model-merging in a computationally efficient manner. PHATGOOSE [10] creates MoEs by treating LoRA [7] finetunes as experts, and routing between them using a learned sigmoid gate for each expert. PHATGOOSE is however limited in that it only allows for the merging of identical expert architectures, where each expert spans the entire transformer model. We posit that such a restrictive is not necessary.

Similar to PHATGOOSE [10], we propose a mechanism for merging experts. Our proposed method, however, allows the integration of individual expert modules with distinct, specialized architectures. We divide this goal into four subgoals:

1. **Finetuned MLP Expert**: As a litmus test for the capabilities of our approach, we start with adding a small MLP expert fine-tuned on a downstream task.
2. **Diffusion Transformer Expert**: By increasing the granularity and allowing individual modules to be added, we provide the freedom of merging much smaller, or much larger

modules post-hoc. As a test for how large and computationally complex our expert can be, we propose adding and training a diffusion expert [6], concretely, a Diffusion Transformer [11].

3. **Expert Stacking**: To test scalability, we propose adding a large number of various experts fine-tuned on a variety of different downstream tasks, all onto a single base model.
4. **Routing Mechanism**: Deciding when to route between experts can be a difficult task to learn [10, 8]. We will experiment with a variety of different routing mechanisms. Co-LLM leverages weak supervision at the token-level [14], while the sigmoid gating learned by PHATGOOSE can be used at either the token-level or module-level granularities [10]. Other works cluster average embeddings for each expert's training dataset to create router weights [2, 8], which often results in poor routing when compared to an oracle router. We propose repurposing Random Network Distillation [3], a mechanism for building intrinsic exploration rewards in reinforcement learning, to measure expert familiarity. We believe a router based on RND will be better suited for scaling to a larger number of modules than both the learned sigmoid gating and embedding cluster average routers.

## 3 Evaluation

A critical consideration is preserving the core competencies and performance characteristics of the base language model after integrating the expert modules. Consequently, our evaluation will encompass not only assessing gains on targeted downstream tasks but also quantifying any potential regressions on the broad generalized capabilities inherent to the original model. We will conduct extensive ablation studies to analyze how each of the proposed subcomponents impacts the overall model performance profile. Specifically, we will measure:

1. How does the type of expert module affect downstream performance and base model performance?
2. How does the number of additional expert modules affect downstream performance and base model performance?
3. What kinds of tradeoffs do each routing mechanism provide?

## 4 Conclusion

In this work, we propose a method for mixing heterogeneous experts into a zoo of experts. We believe that this approach will allow for the creation of more flexible and efficient models, capable of scaling to even larger sizes than previously possible. We will evaluate our method on a variety of downstream tasks and compare it to existing methods for model merging and finetuning, such as PHATGOOSE [10], Co-LLM [14], and Branch-Train-Mix [15].

## References


[1] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *arXiv preprint arXiv:2403.13187*, 2024.

[2] Joshua Belofsky. Token-level adaptation of lora adapters for downstream task generalization. In *2023 6th Artificial Intelligence and Cloud Computing Conference (AICCC)*, pages 168–172, 2023.

[3] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[4] 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. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022.

[5] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[8] Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Exploring the benefits of training expert language models over instruction tuning. In *International Conference on Machine Learning*, pages 14702–14729. PMLR, 2023.

[9] 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.

[10] Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. Learning to route among specialized experts for zero-shot generalization. *arXiv preprint arXiv:2402.05859*, 2024.

[11] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

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

[13] 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. *arXiv preprint arXiv:1701.06538*, 2017.

[14] Shannon Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. Learning to decode collaboratively with multiple language models. *arXiv preprint arXiv:2403.03870*, 2024.

[15] Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Rozière, Jacob Kahn, Daniel Li, Wen-tau Yih, Jason Weston, et al. Branch-train-mix: Mixing expert llms into a mixture-of-experts llm. *arXiv preprint arXiv:2403.07816*, 2024.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[17] Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*, 2024.