Component Transfer Can Exceed Full Model Performance: Investigating Post-Trained Mixture-of-Experts

Rabin Tiwari
Independent Researcher
rabintiwari45@gmail.com

Abstract

Post-training methods such as supervised fine-tuning and preference optimization are widely used to align large language models, yet how their benefits distribute across architectural components and transfer across tasks and prompts remains unclear. In this work, we analyze component-level transfer in a Mixture-of-Experts language model by selectively replacing routers, attention modules, and expert networks between two post-trained Mixture of Experts models trained with different post-training recipes and dataset mixtures. Starting from a SFT+DPO checkpoint, we systematically replace its components (routers, attention, experts) with those from a Tulu3 checkpoint and evaluate the impact of each replacement and their combinations on mathematical and scientific reasoning and a general-purpose classification task under zero-shot, few-shot and Chain of Thought prompting. We find strong component-specific specialization: expert networks account for most gains on mathematical and scientific reasoning, while attention mechanisms consistently outweigh expert transfer on general tasks and router transfer alone provides minimal benefit or harms performance. Prompting strategy further modulates these effects, with expert transfer degrading zero-shot science performance but improving few-shot reasoning. Strategically combining components from different model versions can in some cases match or exceed the performance of the best available model, motivating principled techniques for composing post-trained models into task- and prompt-specific systems without additional training.

1 Introduction

Post-training methods such as supervised fine-tuning (SFT), reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), direct preference optimization (DPO) (Rafailov et al., 2024), and reinforcement learning from verifiable rewards (RLVR) (Lambert et al., 2025) have become standard approaches for aligning large language models with human intent.

Recent advances in post-training recipes—such as Tulu 3 (Lambert et al., 2025)—combine improved data curation, preference optimization, and verifier-based reward modeling to achieve substantial performance gains on benchmark aggregates. These improvements are typically evaluated by comparing full model performance across tasks, with higher aggregate scores interpreted as evidence of universal capability enhancement.

Mixture-of-Experts (MoE) (Shazeer et al., 2017; Fedus et al., 2022) architectures have emerged as a scalable approach to language modeling, routing each token to a subset of specialized expert networks while maintaining computational efficiency. Recent work has shown that MoE models can achieve competitive performance than dense models while being more parameter-efficient (Jiang et al., 2024), and that post-training can further enhance their capabilities through expert specialization (Shen et al., 2023). However, how post-training improvements distribute across different architectural components—routers, attention mechanisms, and expert networks and whether these improvements transfer uniformly across tasks and prompting strategies remains unclear.

While recent research has investigated layer importance in language models, as well as the functional role of attention modules, revealing that different layers and attention modules contribute differently to model capabilities (Zhang et al., 2024; Shim et al., 2022; Gromov et al., 2024). Component-level analysis of post-training effects remains underexplored. Prior work on model editing (Meng et al., 2022) and mechanistic interpretability (Elhage et al., 2021; Olah et al., 2020) has primarily focused on dense models or specific capabilities, leaving questions about how modular architectures like MoE respond to post-training at

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 77–83

the component level. Furthermore, existing evaluation practices focus on end-to-end comparisons on aggregate benchmarks, offering limited visibility into the extent to which individual components contribute to overall model accuracy. Our work additionally contributes to GEM's broader goal of understanding evaluation beyond aggregate model-level metrics by demonstrating that post-training improvements are highly component-, task-, and prompt-dependent. Component-level evaluation provides a complementary perspective to conventional benchmark aggregation and may help practitioners better characterize model capabilities under different generation settings.

We address this gap through systematic component-level analysis of post-training improvements in Mixture-of-Experts models. We compare two instruction-tuned OLMoE-1B-7B (Muennighoff et al., 2024) models that differ in their post-training recipes and data distributions. We systematically transfer routers, attention mechanisms, and expert networks between the two post-trained models—individually and in combination—to isolate the contribution of each component across mathematical reasoning, reading comprehension, and science reasoning tasks under zero-shot, few-shot, and chain-of-thought prompting.

Our work makes three key contributions:

- **Component-level transfer methodology for MoE analysis.** We introduce a systematic approach to isolate architectural components in post-trained Mixture-of-Experts models, enabling controlled evaluation of how different components contribute to performance.

- **Multi-task, multi-prompt evaluation of component effects.** We conduct a comprehensive evaluation across mathematical reasoning, reading comprehension, and science tasks under different prompting strategies, providing fine-grained insights into both the effects of post-training and the component-level contributions of routers, attention mechanisms, and expert networks.

- **Guidance for component-level model merging.** We show how component-level analysis can inform model merging decisions, helping practitioners identify which components contribute most to performance and should be prioritized during merging.

Collectively, these contributions demonstrate that standard full-model evaluation practices are insufficient for understanding post-training effects. By revealing specialization patterns invisible to aggregate benchmarks, our component-level evaluation framework offers practitioners a more principled basis for model assessment and deployment.

2 Related Work

There is a significant body of research focused on understanding language model components and their roles. This section provides an overview of key approaches.

Mixture-of-Experts architectures: Shazeer et al. (2017) introduced sparse MoE layers that activate only a subset of expert networks per token, enabling larger capacity at fixed computational cost. Fedus et al. (2022) scaled MoE models to trillions of parameters, demonstrating competitive performance with dense models while requiring significantly less computation. Shen et al. (2023) observed that instruction tuning induces expert specialization in MoE models, with different experts activating preferentially for different task types.

Analyzing components of LLMs: Research on attention mechanisms has examined their redundancy and functional roles. Michel et al. (2019) demonstrated that multi-head attention exhibits substantial redundancy, with significant portions removable without degrading performance. Layer-wise analysis has revealed asymmetric importance across model depth. Gromov et al. (2024) through systematic pruning showed that later layers have disproportionate impact on complex reasoning capabilities. Zhang et al. (2024) identified the existence of cornerstone layers in LLMs, finding that certain early layers exhibit dominant contributions, with their removal causing drastic performance collapse to near-random guessing levels.

Mechanistic interpretability: Some research views the inner workings of LLMs as computational circuits. Elhage et al. (2021); Olah et al. (2020) conceptualize neural networks as circuits, mapping out how information flows through the network. Meng et al. (2022) focus on locating and understanding functional circuits within LLMs, providing insights into how factual knowledge is stored and retrieved.

3 Methodology

3.1 Models

We compare two instruction-tuned OLMoE-1B-7B models that differ in their post-training procedures and dataset mixes. The first model, OLMoE-1B-7B-0924 (September 2024 release), was post-trained using an earlier recipe combining supervised fine-tuning (SFT) and direct preference optimization (DPO). The second model, OLMoE-1B-7B-0125 (January 2025 release), employs the improved Tulu 3 post-training recipe, which features enhanced SFT data mixtures, refined DPO sampling strategies, and an additional proximal policy optimization (PPO) (Schulman et al., 2017) phase with verifier-based rewards. Throughout this paper, we refer to these checkpoints as the SFT+DPO checkpoint (OLMoE-1B-7B-0924) and the Tulu3 (Lambert et al., 2025) checkpoint (OLMoE-1B-7B-0125). Although the Tulu3 checkpoint achieves stronger aggregate benchmark performance, our results reveal task-dependent complementary strengths between the two checkpoints.

3.2 Component Transfer

We isolate three component types within each transformer layer of the MoE architecture:

- **Router**: The gating network that determines expert selection for each token.
- **Attention**: The self-attention mechanism, including query, key, and value projections and output projection.
- **Experts**: All 64 MLP expert networks within each layer.

Starting from the SFT+DPO checkpoint, we perform systematic component transfer by replacing corresponding parameters with those from the Tulu3 checkpoint while keeping all other parameters frozen. This procedure yields eight transfer configurations: (1) baseline (no transfer), (2-4) individual component transfers (router only, attention only, or experts only), (5-7) pairwise combinations (router + attention, router + experts, or attention + experts), and (8) full transfer (router + attention + experts). We additionally evaluate the Tulu3 checkpoint (9) as the target performance ceiling.

3.3 Evaluation

Tasks. We evaluate component transfer across three diverse tasks: (1) **GSM8K** (Cobbe et al., 2021), a mathematical reasoning dataset requiring multi-step arithmetic problem solving; (2) **BoolQ** (Clark et al., 2019), a reading comprehension task requiring yes/no answers to questions about passages; and (3) **ARC-Challenge** (Clark et al., 2018), a science reasoning benchmark containing grade-school level science questions. These tasks span different reasoning complexities, from simple classification to complex multi-step reasoning.

Prompting strategies. We employ chain-of-thought prompting (Wei et al., 2022) for GSM8K, where we prepend "Let's think step by step" to elicit explicit reasoning traces. For BoolQ and ARC-Challenge, we use zero-shot prompting with direct question-answer formatting. Additionally, we evaluate ARC-Challenge under 5-shot prompting, where we provide five task examples before the test question. This design enables us to assess how prompting strategy modulates component effectiveness.

Evaluation protocol. We use exact match evaluation for all tasks. To balance computational constraints with reliable measurement, we evaluate all nine component transfer configurations on 500 randomly sampled test instances per task. To validate this approach, we perform full evaluation on three key configurations per task—the SFT+DPO checkpoint, the Tulu3 checkpoint, and the highest-performing component combination—comparing 500-sample results. Statistical significance is assessed using two-proportion z-tests, and we report 95% confidence intervals computed via bootstrap resampling with 10,000 samples.

4 Empirical Results

4.1 Component Transfer Can Match or Exceed Source Models

Table 1 presents full evaluation results on key configurations across all tasks. Strikingly, component transfer can exceed both source models: on BoolQ, transferring attention and experts achieves 79.2% ± 1.4%, significantly outperforming both the Tulu3 checkpoint (75.4% ± 1.5%, $p < 0.001$) and the SFT+DPO checkpoint (78.0% ± 1.4%, $p = 0.013$). Similarly, on ARC-Challenge few-shot, transferring experts alone achieves 62.5% ± 2.8%, outperforming both the Tulu3 checkpoint (60.9% ± 2.9%) and the SFT+DPO checkpoint (60.4% ± 2.8%). These results suggest that principled component composition can in some cases match or outperform source models without additional training.

Configuration	BoolQ Zero shot	GSM8K CoT	ARC-C Zero shot	ARC-C 5 shot
SFT+DPO checkpoint	78.0 ± 1.4*	38.3 ± 2.6***	**61.3 ± 2.8**	60.4 ± 2.8
Tulu3 checkpoint	75.4 ± 1.5	**68.5 ± 2.5**	58.7 ± 2.9	60.9 ± 2.9
Attention only	–	–	60.9 ± 2.8	–
Experts only	–	–	–	**62.5 ± 2.8**
Attn + Exp	**79.2 ± 1.4*** **	67.6 ± 2.5	–	–

Table 1: Full evaluation results on key configurations. Accuracies shown with 95% confidence intervals (bootstrap, 10K resamples). Significance vs Tulu3 checkpoint (Tulu 3): *p<0.05, **p<0.01, ***p<0.001.

Configuration	BoolQ Zero shot	GSM8K CoT	ARC-C Zero shot	ARC-C 5 shot
SFT+DPO checkpoint	76.6 ± 3.7	40.0 ± 4.3***	63.4 ± 4.2	60.0 ± 4.3
Tulu3 checkpoint	74.6 ± 3.8	68.0 ± 4.1	58.2 ± 4.3	63.6 ± 4.3
Router only	76.6 ± 3.7	36.6 ± 4.2***	63.2 ± 4.2	60.2 ± 4.3
Attention only	78.2 ± 3.6	43.0 ± 4.3***	63.4 ± 4.2	58.4 ± 4.4
Experts only	73.8 ± 3.8	60.2 ± 4.3*	56.0 ± 4.4	64.6 ± 4.2
Router + Attn	78.4 ± 3.5	44.4 ± 4.4***	61.8 ± 4.2	63.6 ± 4.1
Router + Exp	70.6 ± 4.0	60.6 ± 4.3*	56.0 ± 4.4	64.0 ± 4.2
Attn + Exp	78.6 ± 3.7	68.6 ± 4.1	55.0 ± 4.4	59.4 ± 4.4
Full transfer	77.2 ± 3.7	69.2 ± 4.1	57.6 ± 4.3	63.6 ± 4.2

Table 2: Component transfer results on 500-sample evaluation with 95% confidence intervals. Significance vs Tulu 3: *p<0.05, **p<0.01, ***p<0.001. Full evaluation on selected configurations validates these patterns (Table 1).

4.2 Component Effects Vary by Task and Prompting Strategy

Table 2 presents comprehensive component transfer analysis on 500-sample evaluation. We observe striking task- and prompt-dependent patterns that challenge the assumption of universal post-training improvements.

Mathematical reasoning (GSM8K). On GSM8K, Tulu3 checkpoint substantially outperforms the earlier recipe (68% vs 40%). Component transfer reveals that experts drive this improvement: transferring only experts recovers 72% of the 28-point gap (+20 points on 500 samples), while attention contributes only 10% (+3.0 points). Router transfer actively harms performance (-3.4 points), indicating router-expert co-adaptation during post-training. Full evaluation confirms that combining attention and experts (67.6% ± 2.5%) performs equivalently to Tulu3 checkpoint (68.5% ± 2.5%, p = 0.59), demonstrating that selective component transfer can match the full model without complete replacement.

Reading comprehension (BoolQ). The pattern reverses on BoolQ: the earlier model outperforms Tulu 3 on both 500-sample (76.6% vs 74.6%) and full evaluation (78.0% vs 75.4%). Transferring Tulu3 checkpoint experts harms performance (-2.8 points), while attention provides modest benefits (+1.6 points). Critically, full evaluation reveals that attention-expert transfer achieves 79.2% ± 1.4%, significantly exceeding both Tulu3 checkpoint ($p < 0.001$) and the SFT+DPO checkpoint baseline ($p = 0.013$). This 3.8-point improvement over Tulu 3 demonstrates that component composition can outperform either source model by combining complementary strengths.

Science reasoning depends on prompting. ARC-Challenge exhibits prompting-dependent reversal. Under zero-shot prompting, the SFT+DPO checkpoint excels on both 500-sample (63.4% vs 58.2%) and full evaluation (61.3% vs 58.7%), with expert transfer causing severe degradation (-7.4 points on 500 samples). Attention-only transfer achieves 60.9% ± 2.8% on full evaluation, recovering 85% of the baseline advantage without significance ($p = 0.27$). However, with 5-shot prompting, expert transfer now helps (+4.6 points on 500 samples), achieving 62.5% ± 2.8% on full evaluation compared to Tulu3 checkpoint's 60.9% ± 2.9% ($p = 0.42$). This 12-point swing in expert effectiveness (+4.6 vs -7.4) demonstrates that post-training specialization depends critically on prompting strategy.

4.3 Validation of 500-Sample Methodology

To validate our 500-sample evaluation approach, we performed full evaluation on the three strongest configurations per task. Results show strong agreement: on BoolQ, attention-expert transfer achieved 78.6% on 500 samples versus 79.2% on full evaluation (difference of 0.6 percentage points); on GSM8K, 68.6% versus 67.6% (difference of 1.0 points). This consistency confirms that our 500-sample evaluations provide reliable relative comparisons across the 36 architectural configurations tested (9 per task × 4 conditions), despite reduced precision compared to full benchmarks. Confidence intervals appropriately reflect this precision difference: ±3.6-4.4% margins on 500 samples versus ±1.4-2.9% on full evaluation.

4.4 Post-Training Creates Task-Prompt Specialization

Our results suggest that the Tulu3 post-training recipe induces task- and prompt-specific component specialization rather than uniform improvements across tasks. (1) **Expert specialization**: Tulu3 checkpoint experts excel at math reasoning (+21.9 points) and few-shot science (+4.6 points) but fail at simple classification (-2.8 points) and zero-shot science (-7.4 points). (2) **Attention gen-**

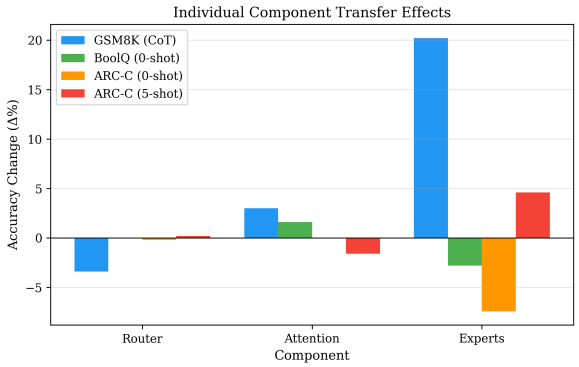

Figure 1: Individual component transfer effects from SFT+DPO checkpoint baseline across tasks and prompting strategies. Expert networks exhibit dramatic task-dependent variation (range: -7.4 to +20 points), while attention shows more consistent but modest effects.

eralization: Unlike experts, attention improvements are more consistent: positive on math (+3.0) and reading (+1.6), neutral on zero-shot science (0.0), with only modest harm on few-shot science (-1.6). (3) **Component superiority**: Selective composition can exceed both source models, as demonstrated on BoolQ where attention-expert transfer outperforms Tulu 3 by 3.8 percentage points ($p < 0.001$).

Figure 1 visualizes these patterns, highlighting the stark contrast between expert specialization (which varies dramatically by task) and attention consistency (which shows modest but stable effects across conditions).

These patterns suggest that Tulu3 post-training optimized experts specifically for complex reasoning with chain-of-thought prompting, creating specialized modules that over-complicate simple tasks but excel at multi-step reasoning. These findings suggest that post-trained models with stronger aggregate benchmark performance may not be uniformly superior across all tasks and prompting settings.

5 Implications for Model Merging and Composition

Our results suggest that component-level transfer offers a principled approach to model merging that goes beyond existing techniques. While prior work on model merging (Wortsman et al., 2022) typically operates on full model weights, our findings demonstrate that selective component merging can exploit task-specific specialization to exceed either source model.

Task-adaptive merging. Our results reveal a clear pattern for task-adaptive composition: expert networks from the Tulu3 checkpoint should be preferred for complex reasoning tasks requiring chain-of-thought prompting, while the SFT+DPO checkpoint's experts are better suited for simple classification tasks. Attention mechanisms from the Tulu3 checkpoint consistently provide modest but reliable improvements across most conditions, suggesting they serve as a robust default component for merging strategies. Router transfer, however, introduces instability due to router-expert co-adaptation, and should be retained from the source model when merging expert networks.

Towards principled component selection. Based on our findings, we propose a simple heuristic for practitioners merging post-trained MoE checkpoints:

Task Type	Experts	Attention
Complex reasoning	Tulu3	Tulu3
Simple classification	SFT+DPO	Tulu3
Science (zero-shot)	SFT+DPO	Either
Science (few-shot)	Tulu3	SFT+DPO

Table 3: Proposed component selection heuristic based on empirical results.

This heuristic is directly motivated by our empirical observations and validated by our results: following these guidelines selects the best-performing configuration on 3 out of 4 evaluated conditions. These findings motivate future work on automated component selection methods that generalize beyond the specific checkpoints studied here.

6 Conclusion and Future Work

We present the first systematic component-level analysis of post-training in Mixture-of-Experts models, revealing that post-training creates task- and prompt-specific specialization rather than universal improvements. Through controlled transfer experiments, we demonstrate that expert networks drive gains on complex reasoning but harm simple classification, prompting strategy critically modulates component effectiveness, and selective component composition can in some cases match or exceed source model performance.

These findings challenge the assumption that post-trained models with stronger aggregate benchmark performance are universally superior across tasks and prompting settings. Our work establishes component-level evaluation as a valuable

methodology for understanding post-training effects and motivates future research on automated component selection, mechanistic analysis of specialization patterns, and principled techniques for task-adaptive model composition. In particular, future work should investigate the extent to which these findings generalize across different MoE architectures, model scales, and post-training recipes. Another important direction is developing mechanistic understanding of component specialization through analyses of routing behavior, expert activation patterns, and router-expert co-adaptation during post-training.

Limitations

Our study has two main limitations. First, due to computational constraints, we evaluate most configurations on 500 randomly sampled test instances, with full evaluation reserved for key configurations. While strong agreement between sampling approaches validates our methodology (e.g., BoolQ: 78.6% vs 79.2%), this limits precision of absolute accuracy estimates. Second, our experiments are limited to one model family (OLMoE-1B-7B) and focus on behavioral transfer rather than mechanistic understanding of how post-training affects individual components. Future work should validate these findings across different MoE architectures, model scales, and investigate the underlying mechanisms driving component specialization.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback and valuable suggestions.

References

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 *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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

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

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2024. The unreasonable ineffectiveness of the deeper layers. *ArXiv*, abs/2403.17887.

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, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Neural Information Processing Systems*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *ArXiv*, abs/1905.10650.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Daniel Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, and 5 others. 2024. Olmoe: Open mixture-of-experts language models. *ArXiv*, abs/2409.02060.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347.

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.

Sheng Shen, Le Hou, Yan-Quan Zhou, Nan Du, S. Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Y. Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. 2023. Mixture-of-experts meets instruction tuning: A winning combination for large language models. In *International Conference on Learning Representations*.

Kyuhong Shim, Jungwook Choi, and Wonyong Sung. 2022. Understanding the role of self attention for efficient speech recognition. In *International Conference on Learning Representations*.

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

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *ArXiv*, abs/2203.05482.

Yang Zhang, Yanfei Dong, and Kenji Kawaguchi. 2024. Investigating layer importance in large language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 469–479, Miami, Florida, US. Association for Computational Linguistics.