CONGRESSIONAL AI: A FRAMEWORK FOR TASK GENERALIZATION AND ALIGNMENT WITH EXPERT LANGUAGE MODELS

By

Keshav Ramji

An Undergraduate Thesis submitted as part of the

WHARTON RESEARCH SCHOLARS

Faculty Advisor:

Weijie Su

Associate Professor of Statistics and Data Science

THE WHARTON SCHOOL, UNIVERSITY OF PENNSYLVANIA

APRIL 2024

Congressional AI: A Framework for Task Generalization and Alignment with Expert Language Models

Keshav Ramji*

April 8, 2024

Contents

*Keshav Ramji is affiliated with the Department of Statistics and Data Science, The Wharton School, and the Department of Computer and Information Sciences at the University of Pennsylvania, Philadelphia, PA 19104; Email: keshavr@wharton.upenn.edu. I would like to sincerely thank my advisor, Weijie Su, for his guidance, feedback, and mentorship. I thank Shayne Longpre, Surbhi Goel, Sang Michael Xie, Hamish Ivison, and Alon Albalak, among many others for helpful discussions and feedback

Abstract

As foundation models have facilitated rapid adaptation to downstream tasks, a challenge remains in efficiently and flexibly improving their instruction-following capabilities and alignment to human preference distributions. We propose a novel modular architecture, *Congressional AI*, consisting of parallel trained "experts", such that the top-k relevant experts can be activated during inference. These experts are obtained by fine-tuning LoRA adapters on interpretable data mixtures; for instruction-tuning, each dataset corresponds to a task cluster, while for preference alignment to improve steerability, each dataset represents a group or persona. Our experiments show that instruction-tuning with *Congressional AI* through low-rank adapter merging is effective via evaluation of cluster-specific adapters across various domains on the MMLU benchmark. These findings demonstrate that Congressional AI is a hardware-efficient and interpretable mixture-of-experts (MoE)-style framework for adapting language models to new tasks and domains, and can be used to further improve both pre-trained and fine-tuned LLMs.

1 Introduction and Background

The last several years have led to an explosion of interest in the development of foundation models; as per the Center for Research on Foundation Models at Stanford University, these are defined as "a model trained on a huge amount of data and [which may be] adapt[able] to many applications." The notion of statistical language models have been proposed since the 1980s, and N-gram models were the dominant approach for much of the 1990s and 2000s, prior to the renewed interest in neural networks and the deep learning renaissance of the 2010s. During this era, architectures appropriate for sequence modeling emerged in the context of language modeling, such as recurrent neural networks (RNNs); while in 2017, the advent of the Transformer (Vaswani et al. [86]) re-shaped much of the community's view on approaching language modeling at scale.

Examples of foundation models that have generated significant public interest are large language models (LLMs) such as GPT-3 (Brown et al. [8]), BERT (Devlin et al. [16]), T5 (Raffel et al. [63]), image generation models such as DALL-E (Ramesh et al. [65]) and Stable Diffusion (Rombach et al. [67]), vision-language models such as CLIP (Radford et al. [61]), and multimodal models such as GPT-4 (OpenAI et al. [54]). The diversity of models available through various means – open-source (OLMo (Groeneveld et al. [22]), open-weights (Llama 2 (Touvron et al. [84]), DBRX, Mixtral (Jiang et al. [33]), closed-source (GPT-4 (OpenAI et al. [54]), Claude 3) – has rapidly grown from 2023 to 2024, and further accelerated the pace of research into the abilities of such models. The two key facets of research in foundation models are the study of the opportunities and risks presented by current foundation models (Bommasani et al. [7]), which includes enabling the responsible use of foundation models for downstream applications as well as the pros and cons of various training approaches, and building new foundation models. Building new foundation models presents several challenges to developers: which pre-training method and which fine-tuning scheme is most effective? Are there any modifications on learning representation (e.g. positional embeddings) that can be applied? How large is the model (in terms of parameter count)? In our discussion of prior literature, we discuss works that have explored several of these aspects (and more), which have shed some light on improvements to be made for future approaches.

The problem of **multi-task learning**[1] can be defined as learning multiple machine learning tasks at

[1] https://www.cs.cornell.edu/ caruana/mlj97.pdf

the same time, often leveraging learned information / knowledge about particular tasks as well as the similarities and differences between the many tasks given in order to learn new tasks. In the case of foundation models, where the goal is to design the model to be applicable to many downstream tasks, this requires the pre-trained representation to generalize well across many possible tasks that may be of interest to a user, so hence understanding the multi-task generalization capabilities of existing foundation models, as well as newly designed models, is crucial.

As artificial intelligence systems become more prominent as tools in the everyday lives of humans, it is imperative to consider the safety and trustworthiness implications of their use. The field of AI safety is largely concerned with the prevention of harmful consequences of AI systems, and the goal of making them more reliable. One key component of this research area is that of *AI alignment*: steering AI systems toward the intended goals, interests, preferences, and principles / values of humans. However, it is worth noting that from a sociocultural standpoint, there is often a large degree of heterogeneity among the values, preferences, etc. of human subgroups, which may be divided in many ways, and there has been little to no prior effort attempting to capture such dynamics. Furthermore, while the intent to enable alignment of new foundation models to human preferences has been present in several works, there has been little study aiming to quantify the alignment quality of such models, and in turn, comparing the models against one another. Hence, developing a framework through which to do so can be crucial for furthering the field of safety and alignment research, and mitigating many risks of foundation models as discussed by the public, governmental agencies, etc. regarding the responsible use of AI.

In this thesis, we study the problem of improving model adaptation; in improving their foundational capabilities while enhancing performance on a wide range of downstream tasks of interest. We propose a new framework, termed ***Congressional AI***, to achieve multi-task generalization across a variety of domains, which furthermore can be extended to improve alignment and steerability to human preferences. Notably, we suggest parallel training of domain "experts" defined by interpretable data mixtures (subdivisions), as a viable means of lightweight instruction tuning, as well as preference optimization at the group level. We empirically seek to suggest evidence of the potential efficacy of this approach through lightweight instruction tuning, with experts being defined over clusters of tasks. We note that defining experts at the level of subpopulations transforms Congressional AI into a means to aid steerability to group preferences at inference time.

1.1 Potential Impact

We anticipate that this work will ultimately have great implications for the alignment, FM risks, and AI ethics communities – one major critique of the deployment of LLMs in many industries is the lack of steerability. That is, the model can possibly prove to be a poor representation of human preferences, and generate toxic or malicious output. Furthermore, there could be differing toxicity standards across subpopulations – one subgroup might take offense to something that another would not, ultimately re-addressing the value of studying the heterogeneity across these subgroups. To the knowledge of the broader AI community, most conversational AI agents (e.g. ChatGPT, Bard) have had some (varying degrees of) manual filtration for toxic content, to avoid producing offensive material – but this does little to mitigate risks long-term on the deployment of FMs. A framework which captures both the population-level preferences through the meta-learner (merged model) as well as the individual subgroup-level preferences would enhance model explainability (understanding which model has learned which pieces of information), and improve our understanding of alignment (RLHF vs RLAIF) and the process of fine-tuning pre-trained models.

This could also spark new discussion along a more qualitative aspect – regarding our selection of subpopulations (what is the "correct" criterion for doing so, and how granular should the groups

be in training the reward model?), leading to new research in computational social science. Is it actually a fair assumption that each homogenous subpopulation generally has the same preference ratio? Furthermore, the OpinionQA paper suggests further that quantifying the alignment properties of FMs (from the original human preference set to the model output probabilities), whether through information theoretic lens (Go et al. [21]) or via another learning algorithm (Yu et al. [104];(Liu et al. [44]), is a hot topic in the field. This area is especially saturated with researchers in academia, given that there is a much lower bar to entry with respect to computing resources, a commodity that has left the development of most new foundation models to industry researchers.

Given that the proposed method is independent of the particular selection of a foundation model (in theory, any foundation model (by definition, a pre-trained model), of any modality, size, etc. can be used as the starting point in the process prior to fine-tuning), this can have far-reaching implications in sparking further research building on this work. For example, with recent works such as LLaMA (Touvron et al. [83]) sparking interest in defying the principles of scaling laws, the open-source AI community has developed many models building on the principles proposed in LLaMA, such as Alpaca[2] and Vicuna[3], with modifications leading to improved performance in dialogue applications (with Vicuna even rivaling ChatGPT). We hope that the framework proposed in this thesis could also spark similar potential lines of study by open-source contributors.

This thesis has the potential to significantly impact the community interested in developing new algorithms for large language models. Having a new lens with which to translate findings from federated learning / distributed training to individually fine-tuned LLMs would advance the development of new methods studying mixtures of LLMs – an area which had been relatively unexplored as of the conception of this thesis in March 2023[4]. We hope that this work can renew interest in Mixture-of-Experts (MoE) style frameworks – albeit, wherein an MoE is constructed subsequent to parallel training of experts, rather than being jointly pre-trained. Furthermore, we believe that this may bridge to theoretical directions, in connection with work such as Chen et al. 2022 (Chen et al. [11]) that build a formal understanding of the properties induced by introducing the MoE layer and the value of different routing methods, different MoE initializations, etc.

Finally, this work will have potential implications for downstream consumers and users of foundation models, conversational AI agents, etc. As mentioned before, the proposed thesis work can lead to increased explainability for a set of entirely black-box models, which is an incredibly valuable outcome, if achievable. This will help in improving the trustworthiness and increasing confidence in the safety of these models, possibly leading to faster adoption in commercial settings. Furthermore, given the widespread attention on AI research at the moment, and the U.S. government establishing an AI task force, we hope that this work could possibly have such far-reaching consequences as to influence the court of public opinion, provided that our framework enables us to steer foundation models with improvements in alignment quality. It is also worth noting, however, that this will likely also lead to further studies on bias, as it is possible that as preferences / behaviors / opinions within a population evolve, the model is not reflective of that; this may highlight the need for more continuous model updates. Fortunately, the examination of the preference ratio from both the human input and the model's outputs in this work will prove instrumental in identifying such sources of bias, and may also provide a glimpse into solving the memorization vs learning problem, an open problem in the field.

[2] https://crfm.stanford.edu/2023/03/13/alpaca.html

[3] https://lmsys.org/blog/2023-03-30-vicuna/

[4] We would like to acknowledge the rapid explosion of concurrent Mixture-of-Experts (MoE) research since June 2023. As such, we have included further related work on recent MoE explorations in Section 5.

2 Background

2.1 Introduction

The body of work examined in this review first discusses traditional methods in natural language generation (NLG) and text classification, prior to the advent of the Transformer (Vaswani et al. [86]). Natural language generation refers to, as implied, AI-generated text, with the intention of capturing a similar semantic structure as natural language (human) text. This line of work examine methods in sequence-to-sequence language modeling, as well as decoder-style language models (more on this later); one major component of this is the semantic parsing literature. Semantic parsing is the process of converting (hence, parsing) natural language text or speech into a different representation; some fundamental tasks that rely on semantic parsing include neural machine translation (NMT), question answering (QA) from a knowledge base / knowledge graph, and code generation from natural language. Given that the primary focus of this thesis work pertains to QA (i.e. from training data, the model has compiled an internal representation of a "knowledge base", from which to answer questions, which is the premise of LLMs fine-tuned for dialogue applications such as ChatGPT), the works referenced will largely pertain to the modern QA literally, while taking some inspiration from NMT (which uses neural networks to predict the likelihood of a sequence of words / tokens) and semantic parsing from text.

Then, we shall discuss recent (2017-present) advances in language modeling and in turn, natural language generation, as the Transformer ushered in a series of new, powerful training strategies and learning representations, leading to the advent of the modern "large language models" (LLMs). Examining this time period provides the necessary ties to the models explored in this thesis, most of which have come to the forefront of AI and NLP research in the last four years.

2.2 A Brief Overview of Architectures for Pre-trained Models

Most deep learning-based modeling methods for NLP tasks are somewhat related to encoder, decoder, or encoder-decoder models. The encoder model is often used to parse natural language into some intermediate representation, depending on the application, and "encode" the input tokens (word). Encoder-only models are most useful for sentence classification, semantic role labeling, part-of-speech classification, and QA with information extraction (i.e. some part of the answer has to be extracted from the question itself).

The decoder is considered often as auto-regressive, in line with the definition of the terminology in statistics – that is, it uses a probability distribution over the tokens / words to predict the next token to be generated. Decoder-only models are most often used in NLG (generative) applications for this reason. These two components, the encoder, and the decoder, have been shown to form an effective learning representation in prior work, as evidenced by autoencoders (origin disputed, but referenced in (Rumelhart, Hinton, and Williams, 1985[5])).

Finally, there are the encoder-decoder models (often referred to as sequence-to-sequence models), which contain both of these components – and hence, the advantages of both. The challenge with integrating these two frameworks is often posed with regards to the choice of pre-training objective (in other words, should it be trained based on loss minimization for the encoder, or for the decoder?). Encoder-decoder models can be particularly useful for semantic parsing tasks and machine translation, as an input sequence (e.g. English text) can first be parsed into some shared intermediate representation, which in turn can be parsed into an output format (e.g. French text, in the context of machine translation).

[5] https://apps.dtic.mil/dtic/tr/fulltext/u2/a164453.pdf

2.3 Classical Methods in Sequence-to-Sequence Language Modeling and Natural Language Generation using Deep Learning

An early work in the field of Neural Machine Translation and sequence-to-sequence language modeling is (Sutskever et al. [81]). This work has an early approach to an encoder-decoder style framework, using a Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997[6]) model to parse from the input sequence to an intermediate vector representation (akin to an encoder) and another LSTM to parse / decode from the intermediate vector encoding to natural language (akin to a decoder model). (Bahdanau et al. [3]) approaches the same task with a similar encoder-decoder framework, but replaces the encoder with a BiRNN (Schuster and Paliwal [71]), and then uses an alignment score between the tokens of the input sequence compared to the target sequence to induce a probability distribution to determine the most likely token, and generate it accordingly. BiRNNs simply introduce connections between two hidden layers moving in opposite directions (e.g. along the input text sequence) relative to a current position, so as to capture the "context" of the text on either side of the current cursor.

LSTMs were the primary model for Sequence-to-Sequence language modeling with deep learning up until the advent of the Transformer in 2017. LSTMs expand on the Recurrent Neural Network (RNN) model class (Rumelhart, Hinton, and Williams, 1985), by introducing gated units - in particular, a "forget gate" - which leverages both the long-term and short-term memory capabilities of the RNN. By introducing this unit that allows selective "forgetting", along with feedback connections between the units and the layers, this partially solves the vanishing gradient problem faced by RNNs. The vanishing gradient problem (Hochreiter, 1997[7]) is when, during backpropagation to train the network using gradient-based optimization methods (e.g. gradient descent, stochastic gradient descent, etc.), some gradients go to 0, preventing the model from learning in that part of the network. Because of these properties, LSTMs rose to prominence, particularly as they could be used for both sequence-to-sequence (text-to-text, including multilingual translation), and natural language generation (leveraging its decoder component).

2.4 Transformer-based Pre-trained Language Models

2.4.1 Transformer Architecture

With the introduction of the Transformer architecture in 2017 (Vaswani et al. [86]), this became the dominant learning representation with which to train language models for natural language processing (NLP) tasks. The Transformer leverages a multi-head attention mechanism – attention is essentially a function (hence, a metric) computed using a query and a key-value part, with regards to an output. The Transformer essentially uses what is known as a scaled dot-product attention function, and computes it in parallel (for different inputs) using the multiple "attention heads", and then concatenates them. Due to greater connectivity between the network components, and with an improved runtime (training) complexity due to the parallel computations, the Transformer presented key advantages over the RNNs that were predominantly used at the time.

Furthermore, the Transformer architecture continues to exploit the prior encoder-decoder style framework, but has opened new doors to pre-training strategies due to its self-attention mechanism, which computes attention between positions in the input sequence after a positional encoding has been applied. It is worth noting that in the Transformer encoder, each of the attention layers can attend bidirectionally to "reach back" and retrieve the tokens of the input sequence, while in the Transformer decoder, the only tokens that can be retrieved are those already generated (i.e. to the left of the current token cursor location).

[6] https://www.bioinf.jku.at/publications/older/2604.pdf
[7] https://www.worldscientific.com/doi/10.1142/S0218488598000094

2.4.2 Pre-training Strategies

It is important to precisely define pre-training in the context of LLMs: pre-training is the process of training a model on (a large quantity of) some training data, possibly pertaining to a particular task, prior to training the model on another dataset / data from another task using its weights. The former is referred to as pre-training, while the latter is considered fine-tuning. It is clear that this scheme poses some benefits: leveraging a pre-trained model is likely more efficient, especially when the new domain has already been seen by the model, or the dataset involved for fine-tuning is relatively small; and the model often converges (close) to its optimal performance faster, as it has already undergone extensive training on data from a (similar, but perhaps not the same) distribution.

The following are some examples of tasks / strategies that have been commonly employed for pre-training LLMs, whether they be encoder-only, decoder-only, or encoder-decoder, adapted from (Qiu et al. [58]). Masked Language Modeling (MLM) (Devlin et al. [16]; Song et al. [76]) involves masking (essentially, hiding) some of the tokens in a given input sequence, and tasking the model with predicting the token that should have been there, using the remaining tokens in the sequence – in other words, filling in the blanks. The two works here are generally considered as some of the first works to apply MLM to Encoder models and Encoder-Decoder models, respectively; the latter particularly applies MLM with text generation (NLG) as the task of interest. (Liu et al. [45]) presents improvements on (Devlin et al. [16]) via dynamic masking – essentially, duplicating the data and applying each masking strategy on one copy of the data, and obtained significant improvements on major benchmark QA datasets.

Denoising methods (akin to Denoising Autoencoders (Vincent et al. [88])) can be broadly categorized as performing some data augmentation and/or input perturbation, with the aim to reconstruct the original sequence from the perturbed input. This idea was proved to be effective in model training, particularly for sequence-to-sequence LMs, through (Lewis et al. [39]), which combined the ideas of (Devlin et al. [16]) with a Decoder model (Radford and Narasimhan [59]), while experimenting with various noising methods, particularly for QA, summarization, text generation, and machine translation applications, after fine-tuning.

One additional style of pre-training objectives involves Contrastive Learning (van den Oord et al. [85]), which uses semantic similarity between pairs of tokens in an input sequence to compute a score heuristic function. One such example of a contrastive learning pre-training objective is Deep Infomax (Hjelm et al. [27]), and an extension for graph learning in NLP through Deep Graph Infomax (Veličković et al. [87]), using an information theoretic framework known as mutual information, and iteratively computing the mutual information between the input and output. Another contrastive learning approach is next sentence prediction (NSP) (Devlin et al. [16]; (Sun et al. [80]), where the objective is essentially to determine whether a pair of sentences follow contiguously within a paragraph – this is more useful for encoder-models, as it pertains to classifying the sentences conditional on their location in the paragraph.

2.4.3 Encoder-only Models

To begin our examination of encoder-only models, we must start with BERT (Devlin et al. [16]), which stands for Bidirectional Encoder Representations from Transformers. This work considers the context of the text on both sides of the current position, hence capturing bidirectional context, and is a simple extension on the Transformer architecture – particularly, only the encoder component. BERT was groundbreaking in introducing pre-training on two aforementioned objectives: masked language modeling (MLM) and next sentence prediction (NSP). Then, DistilBert (Sanh et al. [68]) combined the idea of knowledge distillation (Hinton et al. [26]) during pre-training with the BERT model, allowing for a significantly smaller model but while retaining similar NL (natural language)

reasoning capabilities. RoBERTa (Liu et al. [45]), perhaps the simplest extension on BERT, introduces dynamic masking for its pre-training objective, while the rest of the model remains largely the same. XLM-RoBERTa (Conneau et al. [15]) present an extension of the RoBERTa model to multilingual cross-language transfer and machine translation applications, a major finding given many prior works had considered either reasoning capabilities on English-only corpora or solely considering "high resource languages" (i.e. those that have a significant amount of labeled data). The ERNIE model (Zhang et al. [108]) builds on BERT for knowledge graph (KG) applications, so pre-train the representation using masked language modeling with multiple BERT-style models for text and the KG entities.

2.4.4 Decoder-only Models

The key work of decoder-only models starts with the GPT model (Radford and Narasimhan [59]), which stands for Generative Pre-trained Transformer. This approach combines the paradigms of generative modeling (i.e. estimating a probability distribution over all of the possible inputs and outputs) with discriminative modeling (estimating the probability of producing a particular output, conditioned on the previous tokens generated), and demonstrated strong NLG capabiltiies even with a simple pre-training objective of next token prediction. GPT-2 (Radford et al. [60]) expanded on the GPT framework with a significant increase in model parameter size (from 117M to 1.5B) and minor changes to the architecture, which presented some further improvements on the results in original result. GPT-3 (Brown et al. [8]) presented major improvements on GPT-2, and increased the versatility of the model to various tasks such as code generation, image generation, and other generative modeling tasks. It does this using sparse attention patterns, and by scaling it up to 175B parameters (from 1.5B parameters), were able to make claims on the few-shot reasoning capabilities of such models. Few-shot learning refers to the ability for pre-trained models to generalize to unseen data domains, using only a few labeled examples from that domain.

More recently, larger models (of comparable and larger size compared to GPT-3) have been trained for similar applications in natural language generation, which can also be fine-tuned for general purpose language applications including question answering. One such model is InstructGPT (Ouyang et al. [55]), which introduced an idea known as reinforcement learning from human feedback (RLHF) in the context of language modeling. The core premise involves pre-training a model as usual (perhaps a smaller model; they use a 1.3B parameter model as compared to the 175B parameter GPT-3 model), but after doing supervised fine-tuning, adds reward shaping based on human feedback (i.e. human evaluation of model correctness) in the reinforcement learning (RL) training process via proximal policy optimization (PPO). The idea of RLHF is not new - it was originally proposed in a different form in (Christiano et al. [13]) - but the application of RLHF in large language modeling was novel. Another similar work (Stiennon et al. [77]) by OpenAI (the group responsible for the GPT model series) used an RLHF idea (reward shaping with RL with a human-in-the-loop) for document summarization. Furthermore, (Wei et al. [92]) shows that with an instruction fine-tuned model (i.e. tuning with many different tasks simultaneously), it is possible to achieve zero-shot learning, a significant finding.

In the interrim, OpenAI has developed several models that build on the fundamental ideas of GPT-3, but with various slight modifications and domain-specific fine-tuning for a task in mind. This includes the Codex model (Chen et al. [9]), which fine-tuned GPT-3 on publicly available data from Github for Python code synthesis / generation, generally marking one of the biggest advantages in the AI for Code research community, while also being deployed practically in powering the GitHub Copilot tool. Other models include GPT-J (Wang and Komatsuzaki [89]), a 6B parameter model which uses rotary position encodings (Su et al. [78]) and is trained on the 800GB dataset known as The Pile (Gao et al. [20]); and GPT-NeoX-20B (Black et al. [6]), which was also trained on the Pile,

but has also made model weights publicly available for replication experiments, further fine-tuning, and deployment in various applications, while also outperforming GPT-3 on five-shot (few-shot) reasoning tasks.

Another model, PaLM (Chowdhery et al. [12]), is a 540B parameter model with general purpose NL reasoning capabilities, having achieved state-of-the-art performance on hundreds of tasks with various datasets. While this model is decoder-only, it has several improvements on other models being developed in parallel, including leveraging multi-query attention (Shazeer [72]) and rotary position embeddings (RoPE), as implemented in the RoFormer (Su et al. [78]). Another model that is 175B parameters (same as GPT-3) is OPT, which stands for Open Pre-trained Transformer (Zhang et al. [106]) – the model leverages training improvements via parallel computing that were introduced in Megatron-LM (Shoeybi et al. [75]), which allow for extensive reductions in computational expenses while also involving extensive discussions about toxic and hate speech detection and ethical implications of natural language generation. Finally, there is Megatron-Turing NLG, a 530B parameter model which uses advances in parallel training / computing from Megatron-LM and DeepSpeed-MoE (Rajbhandari et al. [64]), and has an extensive evaluation on many downstream tasks. MoE in the DeepSpeed-MoE work stands for Mixture of Experts (Shazeer et al. [73]) which uses the idea of gating from LSTMs to more efficiently train massive neural networks (at the time, LSTMs), up to 137B parameters. While MoE incorporates multiple expert networks, they are still part of a single model – another approach when working with multiple models is model merging, such as ColD Fusion (Don-Yehiya et al. [17]), which takes a single foundation model, fine-tunes it several ways, fuses them back into a single model by taking an average over the parameters of the fine-tuned models, and then can continue in a loop by using the fused model as the new starting foundation model.

Given that there is extensive interest in the natural language generation capabilities of AI models in general, it is clear that the most extensive work has been aimed at improving decoder-only models. We shall also see that some fundamental progress has also been made in encoder-decoder models.

2.4.5 Encoder-Decoder Models

The first example of a notable encoder-decoder model is BART (Lewis et al. [39]), which stands for Bidirectional Autoregressive Transformers, which combines the encoder model representation from BERT (Devlin et al. [16]) with the decoder model representation from GPT (Radford and Narasimhan [59]), along with a denoising autoencoder pre-training objective. BART has been used for both text generation and semantic parsing tasks to great success, as the denoising process has been particularly effective in capturing semantic patterns and meaning. DQ-BART (Li et al. [42]) makes an improvement on BART through knowledge distillation (Hinton et al. [26]) and quantization (Zhang et al. [107]), which makes the model more efficient by reducing the floating point precision in the stored model weights, enabling for a smaller model (30x smaller than BART) with similar, if not better performance on most tasks. mBART (Liu et al. [46]) extends BART to a multilingual setting, particularly for machine translation tasks, and does the pre-training objective of denoising across multilingual corpora, the first to do so.

Another interesting encoder-decoder model is the T5 model (Raffel et al. [63]), known as the Unified Text-to-Text Transformer, in a transfer learning setting (i.e. pre-training and fine-tuning, across domains). As with BART, there is a multilingual extension on the T5 model, namely, the mT5 model (Xue et al. [100]), which uses a grounding approach to ensure translation into the correct language, and attempts to prevent "hallucination" (making up terminology; in a multilingual setting, generating text in the wrong language). An extension on this is the Switch Transformer (Fedus et al. [19]). This work pairs the T5 model with sparse attention, quantization, and MoE to produce the first LLM with 1T parameters, and still achieves a speed improvement on the largest T5 model

architectures through parallel training and TPU (tensor processing unit) utilization.

2.4.6 Neural Scaling Laws

With these many model architectures in mind, it is important to consider the body of work entailed with identifying optimal training set size and number of model parameters for improvement on out-of-domain (OOD) generalization – that is, achieving strong performance on unseen data. Thus, works on "neural scaling laws" (Kaplan et al. [34]; Hernandez et al. [25]), attempt to identify the optimal tradeoff ratios for a fixed budget of compute power, suggesting that larger models (i.e. more parameters) are sample-efficient (make better use of the samples, even if there are fewer of them), and hence that there is scaling process involved with determining that tradeoff. (Bahri et al. [4]) examine some information theoretic connections to support the presence of neural scaling laws, through an examination of high-dimensional data settings and kernel methods. Another work that touches upon the idea of neural scaling laws and emergent properties (i.e. tasks where a larger model outperforms a smaller model) is (Wei et al. [94]). However, there is still much debate in the community as to whether neural scaling laws should be treated as given, particularly for theoretical analysis; especially since some training methods such as in-context learning (Brown et al. [8]; (Wei et al. [95];Dong et al. [18]) appear to contradict this notion, as on some tasks, a smaller model is easily able to outperform larger models. To this effect, the Inverse Scaling Prize (McKenzie et al. [51]) was launched to identify tasks for which a larger model inherently performs worse than smaller models (where model size is quantified in terms of number of parameters).

2.4.7 Training Language Models with Feedback and Fine-tuning Strategies

In discussing the idea of incorporating feedback with language models, we must start with human-in-the-loop approaches, which has been an active area of NLP research for the past few decades. The progress that has been made in that area has been well documented in a 2021 survey (Wang et al. [91]), which discusses the idea of online and offline feedback loops which exist in question answering (in other words, whether feedback is instantaneous / dynamic or if it requires a separate collection process), as well as connections to the human-computer interaction (HCI) field.

More recently, this area has been taken by storm with the InstructGPT work ([55]) and the rise in popularity of RLHF, as aforementioned. In the last year, several other paradigms for training foundation models with feedback have emerged, often involving different approaches for refining the quality of generated outputs. The Constitutional AI / RLAIF (Reinforcement Learning from AI Feedback; (Bai et al. [5]) framework aims to reduce the amount of harmful outputs generated by foundation models by starting with principles outlined in a human-provided "constitution", prompting a pre-trained LM with this principle, and then comparing the outputs between the models to determine which is more harmless – this yields a fine-tuned preference model, which is then used in the same way as RLHF. Imitation Learning from Language Feedback (ILF; (Scheurer et al. [70]) is a new framework in which natural language human feedback is provided as opposed to a ranking / comparison of outputs, enabling the model to revise accordingly, choosing the best revision given the feedback, and performing imitation learning (learning a policy to achieve this refinement, conditional on the input). Self-Refine (Madaan et al. [49]) has the model provide feedback on its own generations, and then refine its output from its feedback, in an iterative process until convergence.

Instruction-tuning (Wei et al. [93]) has also been shown to be a powerful framework for fine-tuning language models – by fine-tuning LLMs on many NLP tasks together by expressing them as instructions, this has been shown to enable strong zero-shot generalization. FLAN-T5 and FLAN-PaLM were released in 2022 (Chung et al. [14]), which scaled the size of the instruction-tuning approach, both in parameter count (up to 540B for FLAN-PaLM which is decoder-only, and 11B parameters for FLAN-T5-XXL which is encoder-decoder) and number of tasks (1.8k for FLAN-PaLM). More recent

works such as Self-Instruct (Wang et al. [90]) have sought to induce models into generating their own high-quality instructions during instruction-tuning, based on their outputs, in a similar vein to the self-refine work, but with instructions themselves. The Alpaca work and its predecessor, LLaMA (Touvron et al. [83]) both showed the efficacy of instruction fine-tuning on model performance on benchmarks such as MMLU (Hendrycks et al. [24]) – in particular, Alpaca fine-tunes LLaMA on 52k instruction-following samples, and with a 7B parameter model, is able to outperform the 175B parameter GPT-3 model. Instruction-tuning has led to much reconsideration regarding the principle of neural scaling laws, and is a highly promising approach to explore further.

Recent works have sought to derive theoretical foundations corresponding to the alignment problem; in particular, using optimization-based frameworks. For a single model, it has been shown that RLHF (RL with KL-divergence-based penalties), applied in a reward maximization (RM) problem setting, is equivalent to minimizing the reverse KL divergence of the learned generative model with the (implicit) target distribution p (Korbak et al. [36]). The KL-divergence can be written as follows, for samples $\mathcal{X}$, and probability distributions $P(x)$ and $Q(x)$: $D_{KL}(P||Q) = \int_{x \in \mathcal{X}} p(x) * \log(\frac{p(x)}{q(x)}) * \mu(dx)$ for a measure μ which P and Q are absolutely continuous with respect to. This is equal to $\sum_{x \in \mathcal{X}} P(x) * \log(\frac{P(x)}{Q(x)})$ when P and Q are discrete distributions. The reverse KL-divergence is then computed by switching the positions of $Q(x)$ and $P(x)$ in the above definition. Another recent finding is the generative distributional control (GDC) framework proposed in Khalifa et al. [35], wherein the target distribution, an aligned language model, is explicitly defined, and hence yields a forward KL-divergence objective through distributional policy gradient (DPG) as initially proposed in Parshakova et al. [56]. Finally, Go et al. [21] connects these two approaches and further generalizes them: given KL-divergence is a special case of an f-divergence, the incorporation of the GDC framework enables the use of any f-divergence measure, aiming to approximate any target distribution. The f-divergence is defined as follows: for a space Ω and P and Q being probability distributions defined over this space such that P is absolutely continuous with respect to Q, $D_f(P||Q) = \int_{\Omega} f(\frac{dP}{dQ}) * dQ$ for a convex function f.

3 The *Congressional AI* Framework

We propose a novel framework, *Congresional AI*, to improve downstream adaptation. Given a foundation model $\mathcal{M}$ and n data mixtures, we fine-tune $\mathcal{M}$ on these data mixtures in a parallel fashion. For a memory-efficient approach[8], we turn to parameter efficient fine-tuning (PEFT) – in particular, the low-rank adaptation (LoRA; Hu et al. [28]) technique. Fine-tuning with LoRA yields an adapter of weights, which may be flexibly merged and unmerged from the base model, and set to perform inference; the parallel training process yields n *expert* adapters of this nature. Then, we define an MoE layer to introduce a router network with a top-k gating function (applying softmax), by averaging the parameters defining the feedforward layer parameters as a function of the gating function and the expert's frozen feedforward layer parameters. Note that once the MoE layer has been attached, we now have a single MoE model; however, since the router has not been trained jointly with the rest of the network as in traditional MoE pre-training, we must fine-tune the MoE. While in the interest of computational resources and engineering challenges, one could freeze the expert weights on solely train the router network, this would likely require a longer duration of training. Finally, this yields a single model which routes to the top-k *representative experts* at inference time, and produces a response conditioned on the query.

Data mixtures in the Congressional AI framework are intended to chosen such that they correspond to a well defined group – for example, summarization or question answering task categories,

[8]We note that with further computational resources, one could instead perform full fine-tuning to yield the parallel trained experts. However, then one would be required to use an MoE layer to build a router network, and cannot perform Retrieval-of-Experts – see Section 3.1 for a discussion on routing.

or particular demographics for response steerability. As such, the adapters are *expert LLMs* which *represent* said groups – considering these representatives on aggregate yields a form of a Congress, inspiring the name of this framework. We claim that this method is interpretable by construction. Those training the models in a parallel fashion have complete control over data selection (in this case, how the experts are defined with respect to the data mixtures), while at inference time, one could determine the top-k experts which are routed to for each query with minimal changes. As such, one can identify which expert are responsible for providing an answer to a query – this could be valuable for model debugging and other safety explorations, or even data attribution based on the mixture the expert was trained on.

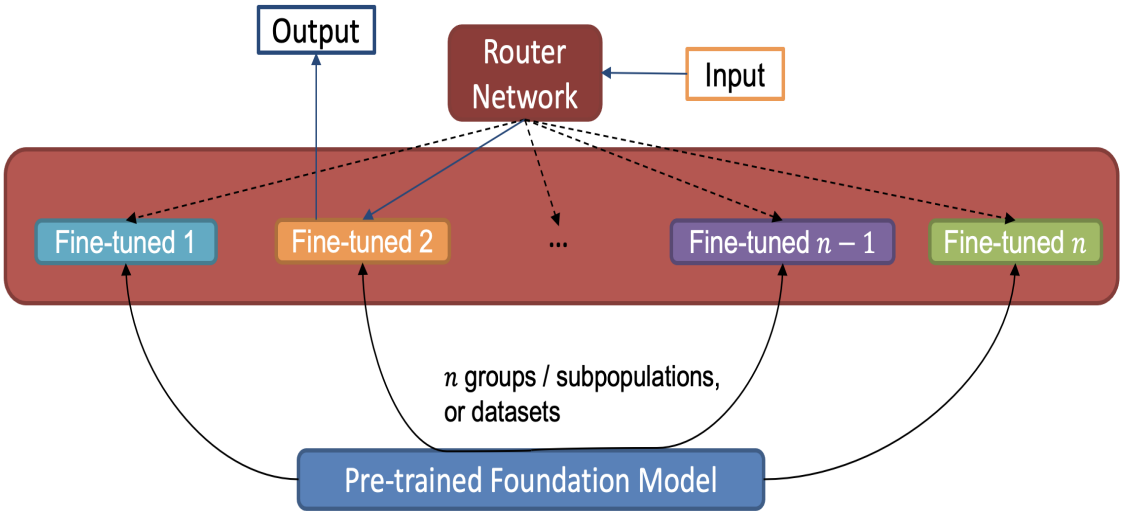

Figure 1: The architecture proposed is a modification on the Switch Transformer ([19]) by replacing the FFN layers with fine-tuned foundation models, each trained independently on a dataset / data subset disjointly, and a top-k token-level routing function. This may be instead viewed from a bottom up perspective, via parallel training of experts and the introduction of a router network to perform top-k routing. The goal is to be able to route into the individual expert network for a "expertise area" (e.g. representing a subpopulation, a mixture, or even a fine-tuning strategy), enabling a more interpretable framework. In the above diagram, the dashed lines indicate candidate expert models, while the 2^{nd} adapter is the one chosen to route to (in a top-1 routing scheme).

3.1 Methods of Router Construction

We note that the router network in the Congressional AI architecture is designed by introduction of a Mixture-of-Experts (MoE) layer, thus yielding an ad-hoc Switch Transformer-like architecture that has been constructed. However, under computational resource constraints, we propose other means of inference that may be used in its place, under the parallel adapter paradigm. One such method is Retrieval-of-Experts (RoE), as used in Jang et al. [32] and Ye et al. [102]; in this technique, one constructs an *expert library* consisting of randomly sampled training instances from the data mixture for the expert and using a SentenceBERT (Reimers and Gurevych [66]) model to store embeddings in said library, and then performing dense retrieval within the *expert library* via maximum inner product search (MIPS). We note that in our case, retrieval yields pointers to the top-k expert adapters corresponding to the task clusters which were most similar. To perform inference and output a response, one could choose to ensemble the k responses from the respective experts or perform majority voting (e.g. for reasoning problem in a Chain-of-Thought setup). Another alternative is to merge the top-k adapters at inference time, before then merging the new adapter into the base model to perform inference; one could use a strategy such as linear task addition (Ilharco et al.

[30]), TIES merging (Yadav et al. [101]) to avoid interference when merging task-specific models, or DARE merging (Yu et al. [103]) to perform this efficiently with sparse deltas (either set to zero or scaled). In either case, the cost incurred with this approach (i.e. without an MoE layer) is storing the adapters (each of which we configure to constitute $\sim 0.1\%$ of the base model's parameters), and more challenging evaluation without a pointer to a single model.

3.2 Instruction Tuning with Parallel Training of Expert LLMs

Instructon tuning of large language models was popularized in Wei et al. [93], wherein the model is fine-tuned on data containing some instruction for the task to be performed. The standardized nature of the data format helps enable a multi-task learning setup, where the resulting FLAN model can adeptly perform zero-shot inference having been trained on 62 tasks. This was scaled further to 1836 tasks in Chung et al. [14], spanning 146 task categories, and yielding 2 instruction tuned models with strong performance: Flan-T5 and Flan-PaLM. We note that to extend Congressional AI to the instruction tuning setting, we could select (and the compare the performance) of experts at different levels of granularity (task level, dataset level, task cluster level, varied cluster definitions, etc.). An example of such a structure could be a question answering subunit, which consists of specialized adapters for different tasks under the question answering cluster (e.g. open-domain QA, content-grounded QA, etc.).

In the Flan Collection, there does not exist a 1-1 mapping between datasets and tasks (i.e. multiple datasets may encompass one task, and one dataset could be used for multiple tasks). Furthermore, it could be possible to construct a hierarchy of Congressional AI "units", wherein routing to an expert (which would usually mean to the expert adapter) would instead route to the router of the subunit, which in turn determines which sub-adapter to use. While the data mixtures for the Flan Collection (Longpre et al. [47]) are made publicly available, there currently do not exist task cluster labels for the dataset, which could have been used to define experts[9]. We instead determine expert clusters through the Data Provenance Collection [48], which has been curated through a large-scale volunteer effort, towards more thorough curation of data licensing and attribution. We select data from task clusters solely with an origin present in the Flan Collection: that is, from Flan 2021, Flan SNI (SuperNaturalInstructions), Flan P3 (Public Pool of Prompts), and Flan CoT (Chain-of-Thought); this yields 18 task clusters, details of which are included in Table 1.

We individually fine-tune LoRA adapters on the chosen base model for each data mixture in a parallel fashion, while transforming the data format to reflect an instruction, input, output format, similar to Alpaca. We suggest that this method enables effective and efficient multi-task training even across dissimilar tasks, thus minimizing potential for negative task transfer. Notably, the flexible nature of our architecture enables new experts to be introduced with little entropy to the rest of the architecture as a result of its modularity; only the method for expert routing must be changed.

3.3 Subpopulation Alignment and Steerability

When applying Congressional AI to the problem of aligning foundation models to preferences expressed by particular subpopulations, we suggest that there are meaningful differences between population-level preferences and the preferences of individual subpopulations (as can be evidenced by opinion polls in political survey research, reflected in the OpinionQA dataset (Santurkar et al. [69])). A consequence of this is that we must determine how best to take these into account while developing (or in this case, improving) the foundation model; we suggest that leveraging group (i.e. demographic) information in such data can be a valuable way to yield data mixtures for parallel training.

[9]This was verified through discussion with the corresponding author.

Task Cluster	Flan Collections	Total Dataset Count	Samples
Bias and Toxicity Detection	Flan SNI	17	116,063
Chain-of-Thought	Flan CoT	9	183,848
Code	Flan SNI	6	51,365
Commonsense Reasoning	Flan SNI	9	61,567
Creative Writing	Flan SNI	5	40,151
Creativity	Flan SNI	1	401
Dialogue Generation	Flan SNI	11	72,793
Language Style Analysis	Flan SNI	10	59,136
Logical and Mathematical Reasoning	Flan SNI	1	6,349
Miscallaneous	Flan SNI	18	39,444
Natural Language Inference	Flan SNI	15	84,266
Open-form Text Generation	Flan SNI, P3, 2021	21	524,931
Question Answering	Flan SNI, P3, 2021	61	904,989
Sequence Tagging	Flan SNI, 2021	7	161,938
Short Text Generation	Flan SNI	56	293,300
Summarization	Flan SNI	9	32,663
Text Classification	Flan SNI, P3, 2021	95	969,924
Translation	Flan SNI, 2021	35	1,064,826

Table 1: Dataset statistics for each of the 18 task clusters from the Flan Collection, as derived from the Data Provenance Collection. The cluster data sources are one of: Flan SNI, Flan P3, Flan 2021, or Flan CoT. The collection we use contains a total of 4,667,954 samples.

While many frameworks, such as Constitutional AI (Bai et al. [5], use self-instructions (Wang et al. [90]) or self-critiques to claim a broader understanding of the vast possible set of human opinions on a matter, our procedure thus considers n individually "homogeneous" subpopulations, fine-tuning the FM n times in parallel (once for the preferences of each subpopulation). Then, as proposed above, we may assign an MoE layer and fine-tune the model accordingly to learn a router, or instead perform retrieval for a less computationally expensive approach.

From a preference modeling perspective, the core idea is that on a given issue (for simplicity, assume that preferences are binary, either A or B on that issue), each homogeneous subpopulation i has a preference ratio $A_i : B_i$, and the population-wide preferences are characterized by $A' : B'$. By training our reward model with the respective ratio $A_i : B_i$ for the particular subgroup, we accurately align this particular fine-tuned model (call this M_i) to the beliefs and intents of subgroup i. Then, by taking an ensemble of the models $M_1, M_2, \ldots, M_n$ – this is analogous to a mixture model over homogeneous populations to capture heterogeneity in the population – we learn the population-wide dynamics and preferences. Then, when the model is prompted generally (subgroup-agnostic), it would reflect the population-wide human preferences, generally akin to current model behavior. However, if given a group-specific prompt (i.e. the prompt addresses a particular subgroup j), the meta-learner will route (send the prompt) to the correct M_j and ensure that the outputs reflect the preference set of subgroup j. These desired dynamics of preference steering are exemplified by Figure 2.

We suggest that to directly learn these preference distributions in a parallel manner as proposed, we can employ the Direct Preference Optimization (DPO; Rafailov et al. [62]) technique, which leverages a reparametrization on the reward formulation (maximizing expected rewards with respect to a KL-divergence penalty) via the introduction of a pairwise preference model, of some flavor of Plackett-Luce for ranking (e.g. Bradley-Terry). Methods such as the DPOTrainer on HuggingFace allow for DPO to be used in tandem with PEFT strategies such as LoRA – thus, we can still yield interchangeable lightweight adapters.

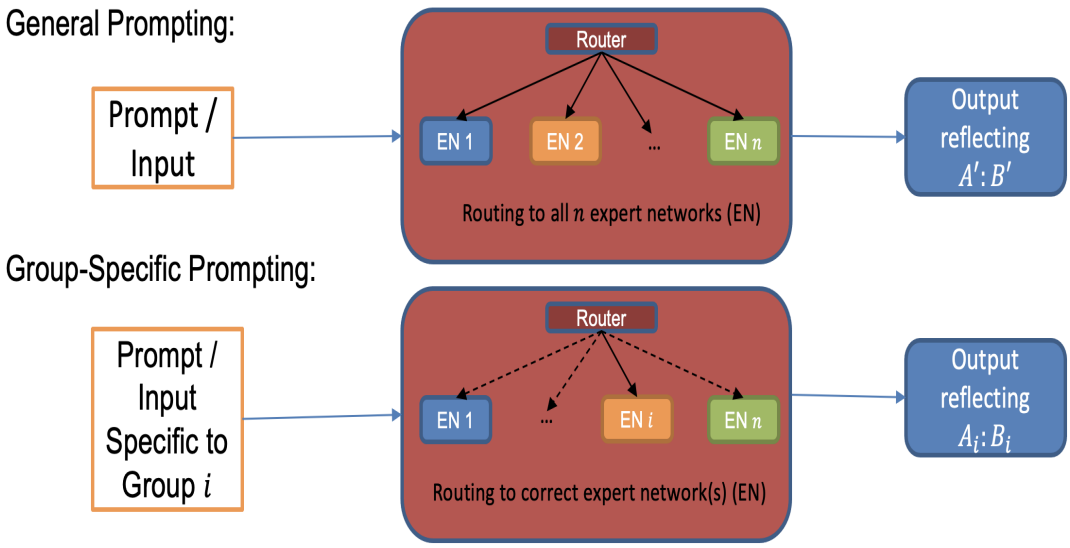

Figure 2: Desired prompting behavior under generic prompting (i.e. relevant to population-wide preferences, and hence routing to all experts) and group-specific prompting (relevant only to a subset of the subpopulations, and routing to those experts accordingly to reflect their preference ratios). In the latter, we take $k = 1$, where EN_i is the i^{th} expert, and is the one routed to for the group-specific prompt, in accordance with the personas expressed.

3.4 *Expert-Instruct* Algorithm and Synthetic Data Generation

Motivated by the Self-Instruct (Wang et al. [90]) technique for generating synthetic instruction tuning data via bootstrapping from the seed instructions and a small subset of the model-generated instructions, we introduce a new algorithm, *Expert-Instruct*. For the *Expert-Instruct* method, for a task cluster, we randomly sample 20 instructions as seeds, and run Self-Instruct with key distinctions. We modify the conditions for inclusion of a new instruction as such: if the ROUGE-L between the new instruction and the existing set exceeds 0.7 for at least one member of the existing set, we proceed to an *instruction refinement* phase, rather than discard the instruction. In this phase, we provide hand-crafted in-context exemplars of instruction refinement, motivated by Self-Refine (Madaan et al. [49]); the forms of refinement are inspired by WizardLM (Xu et al. [98]), and are namely: increasing specificity, changing topic, increasing reasoning steps, and adding constraints. Then, we compute the ROUGE-L of each of these responses with respect to the initial candidate, with a threshold of 0.85 (i.e. if the refined candidate has above a 0.85 similarity with the original candidate, we discard it). For those that pass this initial diversity check, we then compare its diversity with respect to the complete set of instructions generated this far, and use the ROUGE-L check from before (with a threshold of 0.7) to determine membership in the final instruction set.

We describe this notationally as follows: given a data mixture D_t corresponding to task cluster t, we sample 20 seed instructions, denoted as E_t. We first seek to generate k instructions, given E_t; however, given the nature of our *instruction refinement* procedure, where it is possible that both the original candidate, as well as all refinement attempts fail to achieve the diversity threshold, we may have fewer than k instructions. Conversely, if the instructions generated by multiple refinement types are deeemed sufficiently diverse, we may have more than k instructions – therefore, we denote k'_t to be the number of instructions which are generated during this phase of the algorithm, for task t. Then, we generate m_t samples per instruction, thus producing $m_t * k'_t$ samples for task t, where we denote this data S_t. In deploying this algorithm, we suggest re-defining the task cluster data mixture as $D_t \cup S_t$, and fine-tuning the LoRA adapter on this new mixture, for each of the n clusters.

The decision to introduce the refinement phase, with up to two attempts for each refinement type, is largely guided by the notion of task clusters possibly having very similar instruction data (as in, the instructions may be largely the same while the data differs). This leads us to seek greater diversity in formulation; works such as Self-Refine (Madaan et al. [49]) demonstrate that in-context demonstrations are beneficial for models such as GPT-4 to perform multi-aspect refinement. In our setting, the challenge in performing refinement would likely be due to vagueness of the instruction, and as such, we have selected refinement types which create (relatively) clear and explicit differences to be made relative to the previous candidate.

4 Experimental Results

We examine the performance of parallel instruction tuning with 10 out of the 18 task clusters mentioned in Section 3.2. These are: Summarization, Question Answering, Dialogue Generation, Language Style Analysis, Text Classification, Code Generation, Logic and Mathematical Reasoning, Commonsense Reasoning, Creativity, and Miscellaneous. We use the encoder-decoder model, T5-3B (Wei et al. [93]) from HuggingFace to be the base model for all experiments. All fine-tuning was performed using the LoRA technique with parameters of $r = 32$ and $\alpha = 16$.We evaluate our expert adapters on the MMLU benchmark (Hendrycks et al. [24]), a leading evaluation tool for multi-task performance on natural language understanding (NLU) tasks; complete results are included in the Appendix. In Tables 2+3 (where Table 3 serves as a continuation of Table 2), we consider each of the 58 domains in the MMLU benchmark, and note the top 3 adapters (i.e. task clusters) for that domain, as well as the average percentage improvement relative to the baseline T5-3B model for the 3 adapters.

Through this analysis, we observe that most adapters appear at least once across the list of MMLU domains. This hints at the generalization-specialization tradeoff that we are interested in – which the primary purpose of instruction tuning is to imbibe foundational skills for the model, we find that certain experts specialize more heavily than others, resulting in strong performance on a few domains and poor performance on others. Notably, merging all adapters directly (i.e. not in a query-conditional manner) would, based on intuition, result in interference and hinder performance relative to only using the best few experts. In particular, we find that the average percentage change relative to the T5-3B benchmark results across the three experts is strongly positive. The dialogue generation, commonsense reasoning, and logical and mathematical reasoning appear to improve performance especially well on the high school and college-level benchmarks. Using the average change across the top-3 adapters, we find that this would outperform most expert's scores on those domains, and thus would increase the average MMLU score for the model yielded with an optimal routing strategy. Thus, we suggest that routing to the top-3 expert adapters is likely to strongly boost the performance of T5-3B, by way of an interpretable set of data mixtures, with further flexibility define experts across degrees of granularity.

Analysis at the adapter level can help us also identify the extent of variance present across domains. While some domains have multiple experts with roughly the same improvement relative to T5-3B (e.g. High School Computer Science), such that it would be unlikely to see degradation by switching from top-3 to top-1 routing, others like Econometrics only have a single expert which actually improves (namely, that corresponding to the Language Style Analysis cluster), where we may prefer top-1 routing instead. While the framework has not been designed for changing the value of k in a query-conditional manner (and we foresee this to be challenging), enabling such a mechanism in MoE broadly would likely improve performance further. In a domain such as Jurisprudence, where only one expert yielded a +3% improvement, while the others saw degradation of at least 9%, top-1 routing would help ensure that our Congressional AI model does indeed improve on that domain. In the case of domains for which the top-3 experts all saw no improve-

Domain Benchmark	Top-3 Adapters	Avg. % Change
Abstract Algebra	Summarization, QA, Logic & Math	-6 (-6, -6, -6)
Anatomy	Logic & Math, Summarization, QA	+1.33 (4, 0, 0)
Astronomy	Dialog. Gen, Misc. Text Classification	+43.67 (49, 44, 38)
Business Ethics	Creativity, QA, Language Style Analysis	-12 (-3, -15, -18)
Clinical Knowledge	Commonsense, Dialog Gen., Text Classif.	+29.67 (31, 29, 29)
College Biology	Creativity, Logic & Math, Dialog Gen.	-1 (2, 0, -5)
College Chemistry	Dialog Gen., Commonsense, Text Classif.	+133 (141, 141, 117)
College Computer Science	Commonsense, Dialog Gen., Logic & Math	+29.33 (36, 28, 24)
College Mathematics	Dialog Gen., Logic & Math, Commonsense	+40 (40, 40, 40)
College Medicine	Dialog Gen., Commonsense, Text Classif.	+26 (28, 26, 24)
College Physics	Dialog Gen., Commonsense, Text Classif.,	+37.33 (58, 58, -4)
Computer Security	Summarization, Code, Creativity	+4.67 (7, 7, 0)
Conceptual Physics	Language Style, Misc., Dialog Gen.,	+38.67 (58, 56, 2)
Econometrics	Language Style, QA, Summarization	+4.67 (14, 0, 0)
Electrical Engineering	Summarization, QA, Code	+16 (16, 16, 16)
Elementary Mathematics	Language Style, Summarization, QA	+3.33 (4, 3, 3)
Formal Logic	Dialog Gen., Commonsense, Logic & Math	+89.67 (91. 91, 87)
Global Facts	Misc., Summarization, Code	+7 (9, 6, 6)
High School Biology	Dialog. Gen, Commonsense, Logic & Math	+27 (30, 30, 21)
High School Chemistry	QA, Text Classif., Summarization	+4.67 (6, 5, 3)
High School Computer Science	Summarization, Code, Misc.	+65 (65, 65, 65)
High School European History	QA, Text Classif., Summarization	+17.33 (21, 17, 14)
High School Geography	Dialog Gen., Commonsense, Text Classif.	+ 43 (52, 52, 28)
High School Gov. And Pol.	Dialog Gen., Misc., Commonsense	+16 (16, 16, 16)
High School Macroeconomics	Dialog Gen., Commonsense, Logic & Math	+68.33 (84, 82, 39)

Table 2: We analyze the the top-3 adapters for each domain in the MMLU benchmark, based on results comparing individual fine-tuned adapters against the base model, T5-3B. The order of the adapters is in descending order of their % change relative to the T5-3B score for that domain (reported in the parantheses. Full results are contained in the Appendix. Ties are broken arbitrarily.

ment or degradation (e.g. Logical Fallacies, Marketing), routing would help avoid degradation altogether, an important outcome as prior studies have found that multi-task task training can cause degradation on certain tasks as a result of negative task transfer (Aghajanyan et al. [1]; Asai et al. [2]).

We call attention to domains on which little or no improvement is seen with our instruction tuning approach. We take these to be any domain with a less than 5% improvement relative to the T5-3B baseline, with regards to average change over the top-3 adapters. These include: Abstract Algebra, Anatomy, Business Ethics, College Biology, Computer Security, Econometrics, Elementary Mathematics, High School US History, High School World History, Jurisprudence, Logical Fallacies, Marketing, Philosophy, Prehistory, Professional Law, Professional Psychology, Public Relations, and Virology. We observe that a number of these domains require more subjective (or less concrete) responses, and as such, would require more abstract thinking – notably in strategic communications (Marketing, PR), as well as law and ethics (Professional Law, Business Ethics, Jurisprudence, Logical Fallacies, Philosophy). We suggest that perhaps synthetic data efforts could be focused toward such domains, and including the Chain-of-Thought Reasoning cluster could help improve performance.

We observe that these are a number of combinations (e.g. pairings of task clusters) that are especially common across the 58 domains, in some order: 22 domains had Dialogue Generation and Commonsense Reasoning, 16 had Summarization and Question Answering, and 7 had Language Style Analysis and Creativity. In fact, 10 domains had the top-3 combination of Dialogue Generation, Commonsense Reasoning, and Logical and Mathematical Reasoning. We believe that studying such combinations could be useful to determine ordering and/or data selection in instruction tuning.

Domain Benchmark	Top-3 Adapters	Avg. % Change
High School Mathematics	QA, Text Classif., Misc.	+10 (10, 10, 10)
High School Microeconomics	Misc., Dialog Gen., Commonsense	+44 (46, 43, 43)
High School Physics	Dialog Gen., Commonsense, Logic & Math	+20 (24, 24, 12)
High School Psychology	Dialog Gen., Commonsense, Logic & Math	+48.67 (61, 61, 24)
High School Statistics	Dialog Gen., Commonsense, Logic & Math	+173.33 (175, 175, 170)
High School US History	Creativity, Summarization, Code	+1.67 (3, 1, 1)
High School World History	QA, Summarization, Logic & Math	+4.67 (6, 4, 4)
Human Aging	Language Style, Creativity, Summarization	-5.67 (15, 6, -38)
Human Sexuality	Logic & Math, Commonsense, Dialog Gen.	+22 (25, 22, 19)
International Law	Text Classif., Logic & Math, QA	+29 (31, 28, 28)
Jurisprudence	Language Style, Logic & Math, Misc.	-6 (3, -9, -12)
Logical Fallacies	Summarization, QA, Logic & Math	0 (0, 0, 0)
Machine Learning	Language Style, Misc., Creativity	36, 20, 15
Management	Dialog Gen., Commonsense, Logic & Math	+92.33 (94, 94, 89)
Marketing	Summarization, Language Style, Creativity	0 (0, 0, 0)
Medical Genetics	Misc., Dialog Gen., Language Style	+26 (42, 26, 10)
Miscellaneous	Creativity, Language Style, Misc.	0, -1, -3
Moral Disputes	Summarization, QA, Creativity	+5 (5, 5, 5)
Moral Scenarios	Dialog Gen., Commonsense, Logic & Math	+6.33 (7, 7, 5)
Nutrition	Dialog Gen., Commonsense, Misc.	26, 26, 21
Philosophy	Summarization, QA, Code	-3 (-3, -3, -3)
Prehistory	Summarization, QA, Logic & Math	2 (2, 2, 2)
Professional Accounting	QA, Summarization, Text Classif.	+17.33 (20, 16, 16)
Professional Law	Summarization, Code, QA	+0.67 (1, 1, 0)
Professional Medicine	Dialog Gen., Commonsense, Text Classif.	139, 72, 56
Professional Psychology	QA, Logic & Math, Summarization	+0.67 (1, 1, 0)
Public Relations	Creativity, Language Style, Misc.	0 (2, 0, -2)
Security Studies	Dialog Gen., Commonsense, Text Classif.	+84.33 (104, 97, 52)
Sociology	Dialog Gen., Commonsense, QA	+5 (7, 7, 1)
US Foreign Policy	Dialog Gen., Commonsense, Summarization	+34.33 (36, 36, 31)
Virology	Language Style, Misc., Creativity	-1.67 (7, 2, -14)
World Religions	Summarization, Text Classif., QA	+37.33 (38, 38, 36)

Table 3: A continuation of the analysis in Table 2, determining the top-3 experts based on individual adapter MMLU evaluation. Ties are broken arbitrarily, as before, and the full results are contained in the Appendix.

Figure 3 contains a histogram with the frequencies of each task clusters in the top-3 for the 58 categories. We observe that for the MMLU benchmark, it appears that Dialogue Generation, Summarization, Commonsense Reasoning, Logical and Mathematical Reasoning, and Question Answering are the most impactful, each addressing over a third of the domains; conversely, Code is the least used adapter. Future studies could examine the results of TIES-Merging these adapters, as well as other benchmarks for which the Code Generation adapter may be more useful.

5 Related Work

Works related to Congressional AI largely fall into two categories: (language) model merging and ensembling, and Mixture-of-Experts-style architectures.

5.1 Model Merging

As the development of large models becomes increasingly democratized, and we are interested in cross-task transfer learning at great scale (both in model size and the number of tasks), model merging

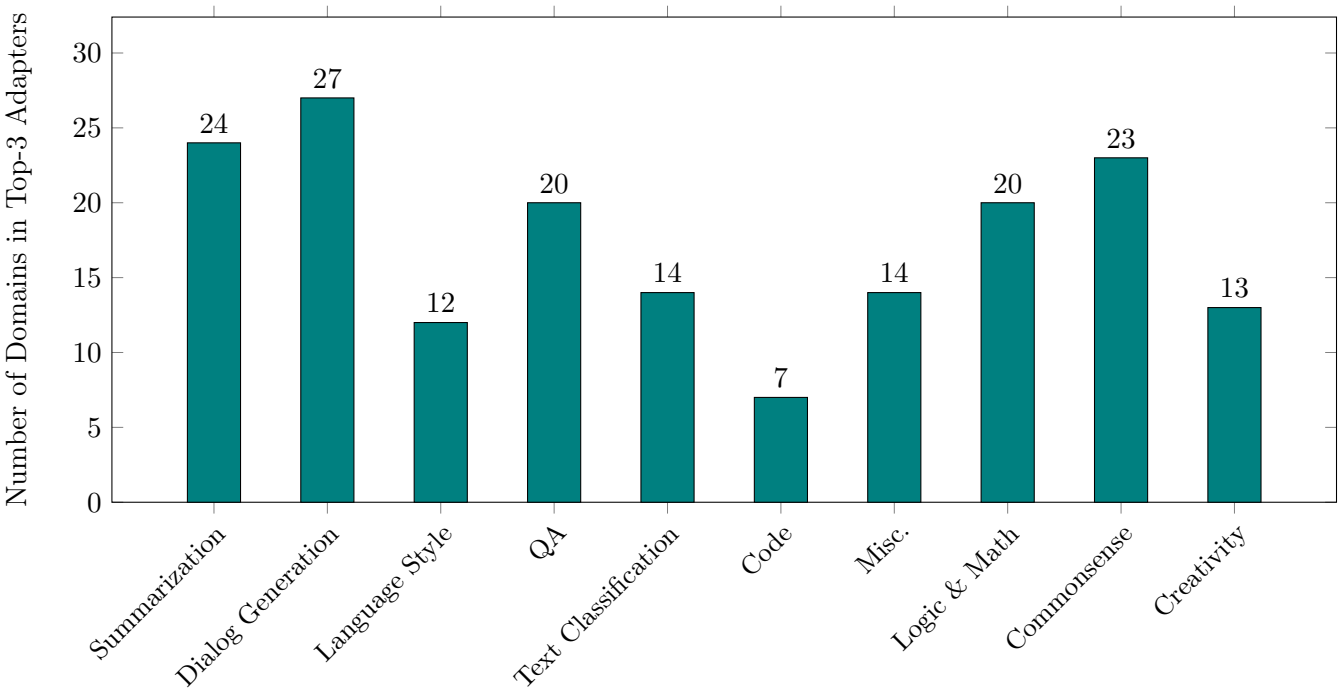

Figure 3: The figure above plots the number of occurrences for which each of the 10 task clusters examined appeared in the Top-3 adapters for one of the 58 MMLU (Hendrycks et al. [24]) domains in the Hendrycks-test.

– by way of averaging the parameters of multiple models – has become a framework of interest. In considering collaborative development, the federated learning setting and the FedAvg approach are an early example of the effectiveness of parameter averaging (McMahan et al. [52]). Model Soups (Wortsman et al. [96]) found that parameter averaging is effective for ensembling language models into a single model which exhibits improvements from across its contributors. Matena and Raffel [50] uses Laplace approximation and Fisher Information to perform merging, enabling new transfer learning strategies such as involving an intermediate task representation or a donor model in a parameter-distillation fashion. ColD Fusion (Don-Yehiya et al. [17]) introduces an iterative loop to improve language models by parallel fine-tuning, merging the parameters with uniform weights, and then recycling the fine-tuned model as a base model, which can then be further fine-tuned; they demonstrate this helps with yielding a stronger base model for downstream adaptation to unseen tasks. Task Arithmetic (Ilharco et al. [29]) suggests that given task vector representation (specifying the direction of improvement within the weight space for the task), there are compositional properties – notably, addition, subtraction, and analogies. That is, one could construct a multi-task model by iterative task addition. MaTS merging (Tam et al. [82]) also explores the notion of tasks, and in conjunction with Fisher Information (FI) matrix insights, uses conjugate gradient methods to solve a system of linear equations, defining task subspaces based on an approximation of the FI matrix and performing merging with those subspaces. TIES-Merging (Yadav et al. [101]) considers interference when seeking to merge models given task vectors, trimming redundant parameters, creating an elected sign vector for the merged model to be created by resolving sign disagreements by magnitude, and performing a disjoint merge using parameters from sign-consistent models with respect to the sign vector. DARE merging (Yu et al. [103]) considers redundancy in the delta parameters from fine-tuning and drop some while re-scaling others, yielding an approach that allows merging task-specific models into a single unified model representation. We refer the readers to Li et al. [41] for a survey on model merging and fusion.

5.2 Language Modeling with Mixture-of-Experts

Examinations into Mixture-of-Experts (MoE) frameworks began in the late 1980s and early 1990s, in Jacobs et al. [31] and Xu et al. [99], wherein the notion of a MoE with pre-trained classifiers which correspond to a subset division of the data, with correspondence to desired tasks was introduced. These works serve as major inspiration for the *Congressional AI* framework proposed in this work, wherein we seek to adapt such notions in a flexible, modular manner to language modeling at scale. Shazeer et al. [73] popularized MoEs for the era of deep learning at scale, by enabling sparsity across many small subnetworks, such that only a fraction of the network's parameters need to be activated for inference, while maintaining its full parameter count. Notably, MoE yielded computational efficiency at inference time, which made it attracting for use at scale. This led to a wide range of interest in various routing concepts beyond the token-level: sentence-level (Kudugunta et al. [38]) and task-level (Kudugunta et al. [37]) as potential candidates. Switch Transformers (Fedus et al. [19]) were a pivotal breakthrough for demonstrating that MoE architectures can scale to the 1T parameter level, through an expert choice routing algorithm. With regards to expert language models, Branch-Train-Merge (Li et al. [40]) introduced a framework for constructing domain specialized "expert" language models (ELMs), trained in an "embarrasingly parallel" manner, such that these models are merged into a modular ELMForest. ELM (Jang et al. [32]) considers the difference between single-task expert LMs as opposed to multi-task instruction tuning, and finds that the former can outperfom the latter even when scaling the number of tasks, perhaps indicating the presence of negative task transfer. Our work blends notions from ELM and ELMForest – the hierarchy induced in Congressional AI using task clusters is akin to ELMForest, while being designed in a modular manner for the instruction tuning setting so as to avoid inter-cluster negative task transfer.

We also note the following concurrent works, with a similar aim to improve downstream adaptation through specialized experts while routing effectively to improve zero-shot generalization, through routers constructed post-hoc. PHATGOOSE (Muqeeth et al. [53]) trains lightweight gates for each (frozen) LoRA module with sigmoid activations, and combines the gates to perform top-k routing. FLAN-MOE (Shen et al. [74]) finds that sparse mixture-of-experts models do indeed benefit from instruction tuning relative to dense models, while improving computational performance. MoLora, introduced in Zadouri et al. [105], proposes a similar framework as Congressional AI, wherein lightweight experts are used with a post-hoc constructed router network through an MoE layer (see Section 3). However, the work primarily explores changing the *number* of experts, rather than the *definition* of the experts, as in our work, where we take experts in our instruction tuning setting to be largely disjoint task clusters. MoLE (Wu et al. [97]) learns a distribution over the experts to determine weight assignment in a hierarchical manner, and at inference time, assigns those weights to the respective adapters in a query-conditioned fashion. We note that zeroed-out weights during this procedure is equivalent to electing not to route to an expert in an MoE framework. Branch-Train-Mix (Sukhbaatar et al. [79]) perhaps most closely follows the formulation laid out in Congressional AI, with regards to flexible fine-tuning strategies (full supervised fine-tuning in their work), construction of an MoE layer, and takes the experts to be specialized in some way, by construction, as in Branch-Train-Merge. The encouraging results yielded by their approach, concurrent with our work, suggests promising potential for our lightweight framework. Aside from the algorithmic approaches highlighted thus far, we must also acknowledge a number of recent Mixture-of-Experts models which have achieved strong performance as "frontier models". These include Mixtral 8x7B (Jiang et al. [33]), DBRX[10], and Grok[11], as well as works at the intersection of MoE with structured state space models such as Mamba (Gu and Dao [23]), including Jamba (Lieber et al. [43]) and MoE-Mamba (Pióro et al. [57]).

[10] https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm
[11] https://x.ai/blog/grok-os

6 Discussion and Future Work

We believe that the insights from the experimental results, with regards to the correlations between Flan Collection task clusters and MMLU categories can shed light into the skills in language models which are necessary to perform various tasks, or excel in certain specialized domains. While we primarily used a linear combination-style of task addition for adapter merging, as in Ilharco et al. [30], it is certainly worth considering the directionality of individual adapters. For example, if using the top-k adapter merging strategy, one may run into interference – in such cases, perhaps a strategy such as TIES-Merging (Yadav et al. [101]) could be effective. Furthermore, we believe that our results provide intuition which may be helpful in skill-driven data selection as in Skill-It (Chen et al. [10]). For example, one could choose consider a hierarchy wherein both instruction tuning-guided skills as well as domain-specific skills are imbued via fine-tuning – e.g. commonsense reasoning and chemistry question answering, respectively – wherein it may behoove one to first instruction tune the model, and then fine-tune on a domain.

The lightweight nature of LoRA adapters enables both efficient fine-tuning as well as memory storage. This easens the process of building intuition about the relationship between task clusters and downstream tasks of interest. At the same time, the use of broad instruction data can help prevent overfitting to target benchmarks, a risk with domain-specific tuning. Furthermore, as only k adapters are used (along with the base model) for a given inference, there is sparsity among the LoRA adapters that may be served with regards to active parameters. Therefore, one could feasibly, in a top-k adapter merging setting for inference, pre-compute and load merged adapters and thus increase throughput and reduce latency. For example, for $k = 3$, while this would introduce $\binom{18}{3} = 816$ adapters (as there are 18 task clusters), each has a parameter count of roughly 0.1% of the base model's parameters, thus adding $\sim$ 81-82% of the base model's count. This therefore creates a tradeoff between the time to perform inference with the memory required to store many (merged) LoRA adapters to be served, which we believe is worth studying. For example, using knowledge about the domains and the nature of the queries being posed (e.g. for a particular downstream task), one could re-define the weights for adapter merging to more strongly address one adapter over the others, as opposed to uniform weights. This view is supported by the observation in Section 4 that many of the domains featured the same top adapters, so only a few such combinations would actually be necessary.

Aside from validating the efficacy of the *Expert-Instruct* algorithm as a means for high-quality synthetic data generation for further instruction-tuning, the nature of diverse candidate generation via *instruction refinement* could lead to synthetically generated instructions which may lie at the intersection of multiple clusters. This is especially likely to occur, by intuition, as we increase the granularity of the clusters; our subdivisions assume disjointness in tasks (instructions + samples), which may not be the case in practice. As such, this method could reveal cross-task relationships which can better inform our hierarchy construction. For example, if we find that two tasks are related as running *Expert-Instruct* on the former yields an instruction which is highly similar to the latter, then it is possible that there may be positive cross-task transfer between this pair – that is, multi-task training on these two (consolidating them into one cluster) would be beneficial.

We also acknowledge that valuable comparisons to our technique would include[12] reproducing FLAN-style multi-task instruction tuning with parameter efficient fine-tuning (PEFT) methods such as LoRA, as opposed to full fine-tuning as in FLAN (Wei et al. [93]; Chung et al. [14]). Such analysis would help with attributing performance differences to parameter efficiency as opposed to task transfer effects in contrasting parallel serving of adapters with multi-task training. Resource constraints also limited the extent of hyperparameter tuning that was possible for our LoRA adapters, as well as

[12]We note these as limitations on demonstrating the efficacy of our approach, as a result of computational resource constraints.

exploration of various fine-tuning strategies. Furthermore, it would be greatly helpful to examine these effects across scales, and across selections of base model (e.g. an already instruction-tuned or chat-tuned model, or a decoder-only architecture instead of T5's encoder-decoder architecture).

An important feature of the Congressional AI framework is its generality: we make no assumptions about the base model other than that it may be fine-tuned. That is, the definition of a "*representative expert*" is left entirely up to the user, who we suggest should take an interpretability-guided approach to determine distinguishing characteristics which could comprise an expertise. We have explored this paradigm in the context of data mixtures, where each data mixture has evident meaning – the task (broad cluster, or more granular) being performed, or a demographic group $\mathcal{G}$ for whom we have an opinion distribution $\mathcal{D}_{\mathcal{G}}$ yielded over. However, we believe that there may be creative uses for this framework, such as serving across different fine-tuning strategies, or even enabling multimodality with different experts that may flexibly combine, as long as the same base model is preserved. With regards to the router design, one could, for example, augment routing decisions with respect to a desired cost function, given priors about the nature of the experts.

Finally, we connect our discussion to the term *Congressional AI*. We consider the nested *Congressional AI* units perspective, yielding a hierarchy, to be akin to state-level congressional bodies, who are in turn responsible to report to their national-level representatives (the cluster-specific router) who are the "torch-bearer" for the state in the broader assembly. We view routing as a problem to be somewhat akin to caucusing – aligning powerful allies for the goal in hand (in this case, to respond to the query). Cost-aware routing is also a valuable extension on the view – it is perhaps analogous to selecting a path of least resistance towards the desired goal. The flexibility of this paradigm yields nearly unbounded potential, and we are excited to see how it may be leveraged in the years to come.

References

[1] Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.468. URL https://aclanthology.org/2021.emnlp-main.468.

[2] Akari Asai, Mohammadreza Salehi, Matthew Peters, and Hannaneh Hajishirzi. ATTEMPT: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6655–6672, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.446. URL https://aclanthology.org/2022.emnlp-main.446.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[4] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws, 2021.

[5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

[6] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022.

[7] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi

Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2021.

[8] 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, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[9] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

[10] Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models, 2023.

[11] Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding mixture of experts in deep learning, 2022.

[12] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[13] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

[14] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[15] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale, 2019.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[17] Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. Cold fusion: Collaborative descent for distributed multitask finetuning, 2022.

[18] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey on in-context learning, 2022.

[19] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021.

[20] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020.

[21] Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization, 2023.

[22] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. Olmo: Accelerating the science of language models, 2024.

[23] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

[24] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2020.

[25] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer, 2021.

[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[27] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization, 2018.

[28] 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, 2021.

[29] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023.

[30] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023.

[31] Robert Jacobs, Michael Jordan, Steven Nowlan, and Geoffrey Hinton. Adaptive mixture of local expert. *Neural Computation*, 3:78–88, 02 1991. doi: 10.1162/neco.1991.3.1.79.

[32] 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, 2023.

[33] 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, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[34] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[35] Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation, 2020.

[36] Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting, 2022.

[37] Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference, 2021.

[38] Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Thang Luong, and Orhan Firat. Exploring routing strategies for multilingual mixture-of-experts models, 2021. URL https://openreview.net/forum?id=ey1XXNzcIZS.

[39] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[40] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models, 2022.

[41] Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. Deep model fusion: A survey, 2023.

[42] Zheng Li, Zijian Wang, Ming Tan, Ramesh Nallapati, Parminder Bhatia, Andrew Arnold, Bing Xiang, and Dan Roth. Dq-bart: Efficient sequence-to-sequence model via joint distillation and quantization, 2022.

[43] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model, 2024.

[44] Hao Liu, Wilson Yan, and Pieter Abbeel. Language quantized autoencoders: Towards unsupervised text-image alignment, 2023.

[45] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[46] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation, 2020.

[47] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: Designing data and methods for effective instruction tuning, 2023.

[48] Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, Kartik Perisetla, Xinyi Wu, Enrico Shippole, Kurt Bollacker, Tongshuang Wu, Luis Villa, Sandy Pentland, and Sara Hooker. The data provenance initiative: A large scale audit of dataset licensing attribution in ai, 2023.

[49] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023.

[50] Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging, 2022.

[51] Ian McKenzie, Alexander Lyzhov, Alicia Parrish, Ameya Prabhu, Aaron Mueller, Najoung Kim, Sam Bowman, and Ethan Perez. The inverse scaling prize, 2022. URL https://github.com/inverse-scaling/prize.

[52] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

[53] Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. Learning to route among specialized experts for zero-shot generalization, 2024.

[54] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux,

Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[55] 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. Training language models to follow instructions with human feedback, 2022.

[56] Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. Distributional reinforcement learning for energy-based sequential models, 2019.

[57] Maciej Pióro, Kamil Ciebiera, Krystian Król, Jan Ludziejewski, Michał Krutul, Jakub Krajewski, Szymon Antoniak, Piotr Miłoś, Marek Cygan, and Sebastian Jaszczur. Moe-mamba: Efficient selective state space models with mixture of experts, 2024.

[58] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63: 1872 – 1897, 2020.

[59] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[60] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[62] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

[63] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

[64] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale, 2022.

[65] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[66] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[67] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[68] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019.

[69] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect?, 2023.

[70] Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with language feedback at scale, 2023.

[71] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093.

[72] Noam Shazeer. Fast transformer decoding: One write-head is all you need, 2019.

[73] 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, 2017.

[74] Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. Mixture-of-experts meets instruction tuning:a winning combination for large language models, 2023.

[75] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2019.

[76] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation, 2019.

[77] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2020.

[78] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021.

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

[80] Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. Nsp-bert: A prompt-based few-shot learner through an original pre-training task–next sentence prediction, 2021.

[81] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[82] Derek Tam, Mohit Bansal, and Colin Raffel. Merging by matching models in task subspaces, 2023.

[83] 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, 2023.

[84] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem 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, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng 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, 2023.

[85] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.

[86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[87] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax, 2018.

[88] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390294. URL https://doi.org/10.1145/1390156.1390294.

[89] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[90] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions, 2022.

[91] Zijie J. Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. Putting humans in the natural language processing loop: A survey, 2021.

[92] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2021.

[93] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2021.

[94] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.

[95] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022.

[96] 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. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022.

[97] Xun Wu, Shaohan Huang, and Furu Wei. MoLE: Mixture of loRA experts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uWvKBCYh4S.

[98] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions, 2023.

[99] Lei Xu, Michael Jordan, and Geoffrey E Hinton. An alternative model for mixtures of experts. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994. URL https://proceedings.neurips.cc/paper_files/paper/1994/file/c8fbbc86abe8bd6a5eb6a3b4d0411301-Paper.pdf.

[100] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer, 2020.

[101] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models, 2023.

[102] Seonghyeon Ye, Joel Jang, Doyoung Kim, Yongrae Jo, and Minjoon Seo. Efficiently enhancing zero-shot performance of instruction following model via retrieval of soft prompt, 2023.

[103] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch, 2024.

[104] Youngjae Yu, Jiwan Chung, Heeseung Yun, Jack Hessel, JaeSung Park, Ximing Lu, Prithviraj Ammanabrolu, Rowan Zellers, Ronan Le Bras, Gunhee Kim, and Yejin Choi. Multimodal knowledge alignment with reinforcement learning, 2022.

[105] Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning, 2023.

[106] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.

[107] Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. Ternary-BERT: Distillation-aware ultra-low bit BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 509–521, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.37. URL https://aclanthology.org/2020.emnlp-main.37.

[108] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities, 2019.

A Appendix

A.1 Summarization

Benchmark	T5-3B	Summarization Adapter	% Change
Abstract Algebra	0.330	0.310	-6
Anatomy	0.333	0.333	0
Astronomy	0.224	0.303	35
Business Ethics	0.320	0.230	-28
Clinical Knowledge	0.230	0.219	-4
College Biology	0.278	0.257	-7
College Chemistry	0.170	0.180	5
College Computer Science	0.250	0.260	4
College Mathematics	0.220	0.250	13
College Medicine	0.260	0.249	-4
College Physics	0.235	0.216	-8
Computer Security	0.280	0.300	7
Conceptual Physics	0.204	0.204	0
Econometrics	0.246	0.246	0
Electrical Engineering	0.255	0.297	16
Elementary Mathematics	0.259	0.267	3
Formal Logic	0.191	0.151	-20
Global Facts	0.310	0.330	6
High School Biology	0.242	0.252	4
High School Chemistry	0.286	0.296	3
High School Computer Science	0.200	0.330	65
High School European History	0.248	0.285	14
High School Geography	0.232	0.253	8
High School Government And Politics	0.030	0.030	1
High School Macroeconomics	0.200	0.213	6
High School Mathematics	0.244	0.263	7
High School Microeconomics	0.244	0.210	-13
High School Physics	0.265	0.272	2
High School Psychology	0.216	0.222	2
High School Statistics	0.171	0.213	24
High School Us History	0.255	0.260	1
High School World History	0.253	0.266	4
Human Aging	0.327	0.202	-38
Human Sexuality	0.237	0.229	-3
International Law	0.289	0.372	28

Table 4: Performance comparison between T5-3b (baseline) and the T5-3B model with the Summarization adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Summarization Adapter	% Change
Jurisprudence	0.287	0.232	-19
Logical Fallacies	0.301	0.301	0
Machine Learning	0.223	0.241	8
Management	0.194	0.194	0
Marketing	0.256	0.256	0
Medical Genetics	0.190	0.200	5
Miscellaneous	0.292	0.272	-7
Moral Disputes	0.278	0.292	5
Moral Scenarios	0.254	0.247	-2
Nutrition	0.232	0.252	8
Philosophy	0.309	0.299	-3
Prehistory	0.287	0.293	2
Professional Accounting	0.231	0.270	16
Professional Law	0.267	0.271	1
Professional Medicine	0.188	0.165	-11
Professional Psychology	0.276	0.276	0
Public Relations	0.345	0.209	-39
Security Studies	0.196	0.241	22
Sociology	0.249	0.249	0
Us Foreign Policy	0.190	0.250	31
Virology	0.289	0.205	-29
World Religions	0.210	0.292	38
Average	0.246982	0.250474	

Table 5: Performance comparison between T5-3b (baseline) and the T5-3B model with the Summarization adapter loaded in, on the MMLU benchmark domains (Continued).

A.2 Dialogue Generation

Benchmark	T5-3B	Dialogue Gen. Adapter	% Change
Abstract Algebra	0.330	0.210	-36
Anatomy	0.333	0.230	-31
Astronomy	0.224	0.336	49
Business Ethics	0.320	0.210	-34
Clinical Knowledge	0.230	0.298	29
College Biology	0.278	0.264	-5
College Chemistry	0.170	0.410	141
College Computer Science	0.250	0.320	28
College Mathematics	0.220	0.310	40
College Medicine	0.260	0.335	28
College Physics	0.235	0.372	58
Computer Security	0.280	0.180	-35
Conceptual Physics	0.204	0.208	2
Econometrics	0.246	0.237	-3
Electrical Engineering	0.255	0.262	2
Elementary Mathematics	0.259	0.270	4
Formal Logic	0.191	0.365	91
Global Facts	0.310	0.180	-41
High School Biology	0.242	0.316	30
High School Chemistry	0.286	0.281	-1
High School Computer Science	0.200	0.190	-5
High School European History	0.248	0.255	2
High School Geography	0.232	0.353	52
High School Government And Politics	0.030	0.035	16
High School Macroeconomics	0.200	0.369	84
High School Mathematics	0.244	0.263	7
High School Microeconomics	0.244	0.349	43
High School Physics	0.265	0.331	24
High School Psychology	0.216	0.349	61
High School Statistics	0.171	0.472	175
High School US History	0.255	0.255	0
High School World History	0.253	0.203	-20
Human Aging	0.327	0.108	-67
Human Sexuality	0.237	0.282	19
International Law	0.289	0.141	-51

Table 6: Performance comparison between T5-3B (baseline) and the T5-3B model with the dialogue generation adapter, on the MMLU benchmark domains.

Benchmark	T5-3B	Dialogue Gen. Adapter	% Change
Jurisprudence	0.287	0.213	-25
Logical Fallacies	0.301	0.233	-22
Machine Learning	0.223	0.161	-28
Management	0.194	0.379	94
Marketing	0.256	0.197	-23
Medical Genetics	0.190	0.240	26
Miscellaneous	0.292	0.204	-30
Moral Disputes	0.278	0.214	-22
Moral Scenarios	0.254	0.273	7
Nutrition	0.232	0.294	26
Philosophy	0.309	0.241	-21
Prehistory	0.287	0.225	-21
Professional Accounting	0.231	0.241	4
Professional Law	0.267	0.244	-8
Professional Medicine	0.188	0.449	139
Professional Psychology	0.276	0.217	-21
Public Relations	0.345	0.227	-34
Security Studies	0.196	0.400	104
Sociology	0.249	0.269	7
US Foreign Policy	0.190	0.260	36
Virology	0.289	0.193	-33
World Religions	0.210	0.175	-16
Average	0.246982	0.264877	

Table 7: Performance comparison between T5-3B (baseline) and the T5-3B model with the dialogue generation adapter, on the MMLU benchmark domains (Continued).

A.3 Language Style Analysis

Benchmark	T5-3B	Language Style Analysis Adapter	% Change
Abstract Algebra	0.330	0.260	-21
Anatomy	0.333	0.252	-24
Astronomy	0.224	0.184	-17
Business Ethics	0.320	0.260	-18
Clinical Knowledge	0.230	0.268	16
College Biology	0.278	0.222	-20
College Chemistry	0.170	0.220	29
College Computer Science	0.250	0.150	-40
College Mathematics	0.220	0.230	4
College Medicine	0.260	0.208	-19
College Physics	0.235	0.196	-16
Computer Security	0.280	0.240	-14
Conceptual Physics	0.204	0.323	58
Econometrics	0.246	0.281	14
Electrical Engineering	0.255	0.221	-13
Elementary Mathematics	0.259	0.257	-1
Formal Logic	0.191	0.198	4
Global Facts	0.310	0.310	0
High School Biology	0.242	0.255	5
High School Chemistry	0.286	0.281	-1
High School Computer Science	0.200	0.240	19
High School European History	0.248	0.236	-4
High School Geography	0.232	0.217	-6
High School Gov. And Pol.	0.030	0.029	-1
High School Macroeconomics	0.200	0.221	10
High School Mathematics	0.244	0.263	7
High School Microeconomics	0.244	0.231	-5
High School Physics	0.265	0.199	-24
High School Psychology	0.216	0.237	9
High School Statistics	0.171	0.171	0
High School US History	0.255	0.230	-9
High School World History	0.253	0.262	3
Human Aging	0.327	0.377	15
Human Sexuality	0.237	0.229	-3
International Law	0.289	0.248	-14

Table 8: Performance comparison between T5-3B (baseline) and the T5-3B model with the Language Style Analysis adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Language Style Analysis Adapter	% Change
Jurisprudence	0.287	0.296	3
Logical Fallacies	0.301	0.239	-20
Machine Learning	0.223	0.304	36
Management	0.194	0.252	29
Marketing	0.256	0.256	0
Medical Genetics	0.190	0.210	10
Miscellaneous	0.292	0.287	-1
Moral Disputes	0.278	0.246	-11
Moral Scenarios	0.254	0.242	-4
Nutrition	0.232	0.229	-1
Philosophy	0.309	0.273	-11
Prehistory	0.287	0.265	-7
Professional Accounting	0.231	0.255	10
Professional Law	0.267	0.235	-11
Professional Medicine	0.188	0.173	-7
Professional Psychology	0.276	0.258	-6
Public Relations	0.345	0.345	0
Security Studies	0.196	0.167	-14
Sociology	0.249	0.239	-4
US Foreign Policy	0.190	0.210	10
Virology	0.289	0.319	10
World Religions	0.210	0.210	0
Average	0.246982	0.240632	

Table 9: Performance comparison between T5-3B (baseline) and the T5-3B model with the Language Style Analysis adapter loaded in, on the MMLU benchmark domains (Continued).

A.4 Question Answering

Benchmark	T5-3B	Question Answering Adapter	% Change
Abstract Algebra	0.330	0.310	-6
Anatomy	0.333	0.333	0
Astronomy	0.224	0.303	35
Business Ethics	0.320	0.270	-15
Clinical Knowledge	0.230	0.283	22
College Biology	0.278	0.264	-5
College Chemistry	0.170	0.180	5
College Computer Science	0.250	0.290	15
College Mathematics	0.220	0.250	13
College Medicine	0.260	0.289	11
College Physics	0.235	0.216	-8
Computer Security	0.280	0.200	-28
Conceptual Physics	0.204	0.204	0
Econometrics	0.246	0.246	0
Electrical Engineering	0.255	0.297	16
Elementary Mathematics	0.259	0.267	3
Formal Logic	0.191	0.198	4
Global Facts	0.310	0.320	3
High School Biology	0.242	0.265	9
High School Chemistry	0.286	0.305	6
High School Computer Science	0.200	0.320	60
High School European History	0.248	0.303	21
High School Geography	0.232	0.227	-2
High School Gov. And Pol.	0.030	0.030	1
High School Macroeconomics	0.200	0.215	7
High School Mathematics	0.244	0.270	10
High School Microeconomics	0.244	0.214	-12
High School Physics	0.265	0.272	2
High School Psychology	0.216	0.222	2
High School Statistics	0.171	0.232	35
High School US History	0.255	0.250	-1
High School World History	0.253	0.270	6
Human Aging	0.327	0.135	-58
Human Sexuality	0.237	0.229	-3
International Law	0.289	0.372	28

Table 10: Performance comparison between T5-3B (baseline) and the T5-3B model with the Question Answering adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Question Answering Adapter	% Change
Jurisprudence	0.287	0.232	-19
Logical Fallacies	0.301	0.301	0
Machine Learning	0.223	0.232	3
Management	0.194	0.252	29
Marketing	0.256	0.252	-1
Medical Genetics	0.190	0.200	5
Miscellaneous	0.292	0.272	-7
Moral Disputes	0.278	0.292	5
Moral Scenarios	0.254	0.241	-4
Nutrition	0.232	0.258	11
Philosophy	0.309	0.299	-3
Prehistory	0.287	0.293	2
Professional Accounting	0.231	0.277	20
Professional Law	0.267	0.265	0
Professional Medicine	0.188	0.176	-5
Professional Psychology	0.276	0.281	1
Public Relations	0.345	0.209	-39
Security Studies	0.196	0.261	33
Sociology	0.249	0.254	1
US Foreign Policy	0.190	0.250	31
Virology	0.289	0.205	-29
World Religions	0.210	0.269	27
Average	0.246982	0.253018	

Table 11: Performance comparison between T5-3B (baseline) and the T5-3B model with the Question Answering adapter loaded in, on the MMLU benchmark domains (Continued).

A.5 Text Classification

Benchmark	T5-3B	Text Classification Adapter	% Change
Abstract Algebra	0.330	0.300	-9
Anatomy	0.333	0.222	-33
Astronomy	0.224	0.309	38
Business Ethics	0.320	0.190	-40
Clinical Knowledge	0.230	0.298	29
College Biology	0.278	0.257	-7
College Chemistry	0.170	0.370	117
College Computer Science	0.250	0.240	-4
College Mathematics	0.220	0.260	18
College Medicine	0.260	0.324	24
College Physics	0.235	0.226	-4
Computer Security	0.280	0.230	-17
Conceptual Physics	0.204	0.200	-2
Econometrics	0.246	0.246	0
Electrical Engineering	0.255	0.241	-5
Elementary Mathematics	0.259	0.257	-1
Formal Logic	0.191	0.278	45
Global Facts	0.310	0.280	-9
High School Biology	0.242	0.248	2
High School Chemistry	0.286	0.300	5
High School Computer Science	0.200	0.260	30
High School European History	0.248	0.291	17
High School Geography	0.232	0.298	28
High School Gov. And Pol.	0.030	0.030	2
High School Macroeconomics	0.200	0.210	5
High School Mathematics	0.244	0.270	10
High School Microeconomics	0.244	0.340	39
High School Physics	0.265	0.232	-12
High School Psychology	0.216	0.237	9
High School Statistics	0.171	0.264	54
High School US History	0.255	0.221	-13
High School World History	0.253	0.257	1
Human Aging	0.327	0.202	-38
Human Sexuality	0.237	0.214	-9
International Law	0.289	0.380	31

Table 12: Performance comparison between T5-3B (baseline) and the T5-3B model with the Text Classification adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Text Classification Adapter	% Change
Jurisprudence	0.287	0.232	-19
Logical Fallacies	0.301	0.288	-4
Machine Learning	0.223	0.214	-3
Management	0.194	0.204	4
Marketing	0.256	0.235	-8
Medical Genetics	0.190	0.190	0
Miscellaneous	0.292	0.231	-20
Moral Disputes	0.278	0.289	4
Moral Scenarios	0.254	0.255	0
Nutrition	0.232	0.203	-12
Philosophy	0.309	0.293	-5
Prehistory	0.287	0.207	-27
Professional Accounting	0.231	0.270	16
Professional Law	0.267	0.260	-2
Professional Medicine	0.188	0.294	56
Professional Psychology	0.276	0.221	-20
Public Relations	0.345	0.200	-42
Security Studies	0.196	0.298	52
Sociology	0.249	0.254	1
US Foreign Policy	0.190	0.220	15
Virology	0.289	0.229	-20
World Religions	0.210	0.286	36
Average	0.246982	0.251842	

Table 13: Performance comparison between T5-3B (baseline) and the T5-3B model with the Text Classification adapter loaded in, on the MMLU benchmark domains (Continued).

A.6 Code Generation

Benchmark	T5-3B	Code Generation Adapter	% Change
Abstract Algebra	0.330	0.310	-6
Anatomy	0.333	0.333	0
Astronomy	0.224	0.303	35
Business Ethics	0.320	0.230	-28
Clinical Knowledge	0.230	0.219	-4
College Biology	0.278	0.257	-7
College Chemistry	0.170	0.180	5
College Computer Science	0.250	0.260	4
College Mathematics	0.220	0.250	13
College Medicine	0.260	0.249	-4
College Physics	0.235	0.216	-8
Computer Security	0.280	0.300	7
Conceptual Physics	0.204	0.204	0
Econometrics	0.246	0.246	0
Electrical Engineering	0.255	0.297	16
Elementary Mathematics	0.259	0.267	3
Formal Logic	0.191	0.151	-20
Global Facts	0.310	0.330	6
High School Biology	0.242	0.252	4
High School Chemistry	0.286	0.296	3
High School Computer Science	0.200	0.330	65
High School European History	0.248	0.285	14
High School Geography	0.232	0.253	8
High School Gov. And Pol.	0.030	0.030	1
High School Macroeconomics	0.200	0.213	6
High School Mathematics	0.244	0.263	7
High School Microeconomics	0.244	0.210	-13
High School Physics	0.265	0.272	2
High School Psychology	0.216	0.222	2
High School Statistics	0.171	0.213	24
High School US History	0.255	0.260	1
High School World History	0.253	0.266	4
Human Aging	0.327	0.202	-38
Human Sexuality	0.237	0.229	-3
International Law	0.289	0.372	28

Table 14: Performance comparison between T5-3B (baseline) and the T5-3B model with the Code adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Code Adapter	% Change
Jurisprudence	0.287	0.232	-19
Logical Fallacies	0.301	0.301	0
Machine Learning	0.223	0.241	8
Management	0.194	0.194	0
Marketing	0.256	0.256	0
Medical Genetics	0.190	0.200	5
Miscellaneous	0.292	0.272	-7
Moral Disputes	0.278	0.292	5
Moral Scenarios	0.254	0.247	-2
Nutrition	0.232	0.252	8
Philosophy	0.309	0.299	-3
Prehistory	0.287	0.293	2
Professional Accounting	0.231	0.270	16
Professional Law	0.267	0.271	1
Professional Medicine	0.188	0.165	-11
Professional Psychology	0.276	0.276	0
Public Relations	0.345	0.209	-39
Security Studies	0.196	0.241	22
Sociology	0.249	0.249	0
US Foreign Policy	0.190	0.250	31
Virology	0.289	0.205	-29
World Religions	0.210	0.292	26
Average	0.246982	0.253831	

Table 15: Performance comparison between T5-3B (baseline) and the T5-3B model with the Code adapter loaded in, on the MMLU benchmark domains (Continued).

A.7 Miscellaneous

Benchmark	T5-3B	Misc. Flan Adapter	% Change
Abstract Algebra	0.330	0.260	-21
Anatomy	0.333	0.267	-19
Astronomy	0.224	0.322	44
Business Ethics	0.320	0.250	-21
Clinical Knowledge	0.230	0.215	-6
College Biology	0.278	0.215	-22
College Chemistry	0.170	0.180	5
College Computer Science	0.250	0.150	-40
College Mathematics	0.220	0.230	4
College Medicine	0.260	0.208	-19
College Physics	0.235	0.157	-33
Computer Security	0.280	0.240	-14
Conceptual Physics	0.204	0.319	56
Econometrics	0.246	0.237	-3
Electrical Engineering	0.255	0.290	13
Elementary Mathematics	0.259	0.212	-18
Formal Logic	0.191	0.151	-20
Global Facts	0.310	0.340	9
High School Biology	0.242	0.287	18
High School Chemistry	0.286	0.281	-1
High School Computer Science	0.200	0.330	65
High School European History	0.248	0.279	12
High School Geography	0.232	0.237	2
High School Gov. And Pol.	0.030	0.035	16
High School Macroeconomics	0.200	0.254	26
High School Mathematics	0.244	0.270	10
High School Microeconomics	0.244	0.357	46
High School Physics	0.265	0.225	-14
High School Psychology	0.216	0.237	9
High School Statistics	0.171	0.454	164
High School US History	0.255	0.250	-1
High School World History	0.253	0.236	-6
Human Aging	0.327	0.202	-38
Human Sexuality	0.237	0.260	9
International Law	0.289	0.265	-8

Table 16: Performance comparison between T5-3b (baseline) and the T5-3B model with the Miscellaneous Flan adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Misc. Flan Adapter	% Change
Jurisprudence	0.287	0.250	-12
Logical Fallacies	0.301	0.258	-14
Machine Learning	0.223	0.268	20
Management	0.194	0.184	-4
Marketing	0.256	0.205	-20
Medical Genetics	0.190	0.270	42
Miscellaneous	0.292	0.281	-3
Moral Disputes	0.278	0.240	-13
Moral Scenarios	0.254	0.242	-4
Nutrition	0.232	0.281	21
Philosophy	0.309	0.289	-6
Prehistory	0.287	0.219	-23
Professional Accounting	0.231	0.255	10
Professional Law	0.267	0.233	-12
Professional Medicine	0.188	0.213	13
Professional Psychology	0.276	0.204	-26
Public Relations	0.345	0.336	-2
Security Studies	0.196	0.171	-12
Sociology	0.249	0.239	-4
US Foreign Policy	0.190	0.240	26
Virology	0.289	0.295	2
World Religions	0.210	0.210	0
Average	0.246982	0.247105	

Table 17: Performance comparison between T5-3b (baseline) and the T5-3B model with the Miscellaneous Flan adapter loaded in, on the MMLU benchmark domains.

A.8 Logical and Mathematical Reasoning

Benchmark	T5-3B	Logical and Mathematical Reasoning Adapter	% Change
Abstract Algebra	0.330	0.310	-6
Anatomy	0.333	0.348	4
Astronomy	0.224	0.303	35
Business Ethics	0.320	0.220	-31
Clinical Knowledge	0.230	0.215	-6
College Biology	0.278	0.278	0
College Chemistry	0.170	0.200	17
College Computer Science	0.250	0.310	24
College Mathematics	0.220	0.310	40
College Medicine	0.260	0.243	-6
College Physics	0.235	0.216	-8
Computer Security	0.280	0.170	-39
Conceptual Physics	0.204	0.204	0
Econometrics	0.246	0.219	-10
Electrical Engineering	0.255	0.214	-16
Elementary Mathematics	0.259	0.262	1
Formal Logic	0.191	0.357	87
Global Facts	0.310	0.330	6
High School Biology	0.242	0.293	21
High School Chemistry	0.286	0.296	3
High School Computer Science	0.200	0.200	0
High School European History	0.248	0.242	-2
High School Geography	0.232	0.253	8
High School Gov. And Pol.	0.030	0.030	1
High School Macroeconomics	0.200	0.280	39
High School Mathematics	0.244	0.259	6
High School Microeconomics	0.244	0.210	-13
High School Physics	0.265	0.298	12
High School Psychology	0.216	0.270	24
High School Statistics	0.171	0.463	170
High School US History	0.255	0.196	-23
High School World History	0.253	0.266	4
Human Aging	0.327	0.193	-41
Human Sexuality	0.237	0.298	25
International Law	0.289	0.372	28

Table 18: Performance comparison between T5-3B (baseline) and the T5-3B model with the Logical & Mathematical Reasoning adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Logical & Mathematical Reasoning Adapter	% Change
Jurisprudence	0.287	0.259	-9
Logical Fallacies	0.301	0.301	0
Machine Learning	0.223	0.205	-7
Management	0.194	0.369	89
Marketing	0.256	0.227	-11
Medical Genetics	0.190	0.210	10
Miscellaneous	0.292	0.203	-30
Moral Disputes	0.278	0.289	4
Moral Scenarios	0.254	0.267	5
Nutrition	0.232	0.252	8
Philosophy	0.309	0.283	-8
Prehistory	0.287	0.293	2
Professional Accounting	0.231	0.248	7
Professional Law	0.267	0.270	0
Professional Medicine	0.188	0.279	49
Professional Psychology	0.276	0.279	1
Public Relations	0.345	0.209	-39
Security Studies	0.196	0.241	22
Sociology	0.249	0.249	0
US Foreign Policy	0.190	0.250	31
Virology	0.289	0.211	-27
World Religions	0.210	0.246	16
Average	0.246982	0.259088	

Table 19: Performance comparison between T5-3B (baseline) and the T5-3B model with the Logical & Mathematical Reasoning adapter loaded in, on the MMLU benchmark domains (Continued).

A.9 Commonsense Reasoning

Benchmark	T5-3B	Commonsense Reasoning Adapter	% Change
Abstract Algebra	0.330	0.210	-36
Anatomy	0.333	0.230	-31
Astronomy	0.224	0.303	35
Business Ethics	0.320	0.210	-34
Clinical Knowledge	0.230	0.302	31
College Biology	0.278	0.264	-5
College Chemistry	0.170	0.410	141
College Computer Science	0.250	0.340	36
College Mathematics	0.220	0.310	40
College Medicine	0.260	0.330	26
College Physics	0.235	0.372	58
Computer Security	0.280	0.180	-35
Conceptual Physics	0.204	0.208	2
Econometrics	0.246	0.237	-3
Electrical Engineering	0.255	0.241	-5
Elementary Mathematics	0.259	0.267	3
Formal Logic	0.191	0.365	91
Global Facts	0.310	0.180	-41
High School Biology	0.242	0.316	30
High School Chemistry	0.286	0.281	-1
High School Computer Science	0.200	0.190	-5
High School European History	0.248	0.255	2
High School Geography	0.232	0.353	52
High School Gov. And Pol.	0.030	0.035	16
High School Macroeconomics	0.200	0.364	82
High School Mathematics	0.244	0.263	7
High School Microeconomics	0.244	0.349	43
High School Physics	0.265	0.331	24
High School Psychology	0.216	0.350	61
High School Statistics	0.171	0.472	175
High School US History	0.255	0.255	0
High School World History	0.253	0.207	-18
Human Aging	0.327	0.108	-67
Human Sexuality	0.237	0.290	22
International Law	0.289	0.141	-51

Table 20: Performance comparison between T5-3B (baseline) and the T5-3B model with the Commonsense Reasoning Adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Commonsense Reasoning Adapter	% Change
Jurisprudence	0.287	0.213	-25
Logical Fallacies	0.301	0.233	-22
Machine Learning	0.223	0.161	-28
Management	0.194	0.379	94
Marketing	0.256	0.197	-23
Medical Genetics	0.190	0.180	-5
Miscellaneous	0.292	0.206	-29
Moral Disputes	0.278	0.214	-22
Moral Scenarios	0.254	0.274	7
Nutrition	0.232	0.294	26
Philosophy	0.309	0.244	-20
Prehistory	0.287	0.222	-22
Professional Accounting	0.231	0.248	7
Professional Law	0.267	0.247	-7
Professional Medicine	0.188	0.324	72
Professional Psychology	0.276	0.217	-21
Public Relations	0.345	0.227	-34
Security Studies	0.196	0.388	97
Sociology	0.249	0.269	7
US Foreign Policy	0.190	0.260	36
Virology	0.289	0.193	-33
World Religions	0.210	0.175	-16
Average	0.246982	0.261123	

Table 21: Performance comparison between T5-3B (baseline) and the T5-3B model with the Commonsense Reasoning Adapter loaded in, on the MMLU benchmark domains (Continued).

A.10 Creativity

Benchmark	T5-3B	Creativity Adapter	% Change
Abstract Algebra	0.330	0.310	-6
Anatomy	0.333	0.319	-4
Astronomy	0.224	0.210	-5
Business Ethics	0.320	0.310	-3
Clinical Knowledge	0.230	0.223	-3
College Biology	0.278	0.285	2
College Chemistry	0.170	0.180	5
College Computer Science	0.250	0.260	4
College Mathematics	0.220	0.220	0
College Medicine	0.260	0.254	-2
College Physics	0.235	0.206	-12
Computer Security	0.280	0.280	0
Conceptual Physics	0.204	0.204	0
Econometrics	0.246	0.237	-3
Electrical Engineering	0.255	0.276	8
Elementary Mathematics	0.259	0.259	0
Formal Logic	0.191	0.182	-4
Global Facts	0.310	0.310	0
High School Biology	0.242	0.252	4
High School Chemistry	0.286	0.296	3
High School Computer Science	0.200	0.240	19
High School European History	0.248	0.236	-4
High School Geography	0.232	0.247	6
High School Gov. And Pol.	0.030	0.031	4
High School Macroeconomics	0.200	0.215	7
High School Mathematics	0.244	0.237	-3
High School Microeconomics	0.244	0.210	-13
High School Physics	0.265	0.278	4
High School Psychology	0.216	0.218	0
High School Statistics	0.171	0.176	2
High School US History	0.255	0.265	3
High School World History	0.253	0.240	-5
Human Aging	0.327	0.350	6
Human Sexuality	0.237	0.191	-19
International Law	0.289	0.297	2

Table 22: Performance comparison between T5-3B (baseline) and the T5-3B model with the Creativity Adapter loaded in, on the MMLU benchmark domains.

Benchmark	T5-3B	Creativity Adapter	% Change
Jurisprudence	0.287	0.232	-19
Logical Fallacies	0.301	0.294	-2
Machine Learning	0.223	0.259	15
Management	0.194	0.204	4
Marketing	0.256	0.256	0
Medical Genetics	0.190	0.200	5
Miscellaneous	0.292	0.291	0
Moral Disputes	0.278	0.292	5
Moral Scenarios	0.254	0.250	-1
Nutrition	0.232	0.239	2
Philosophy	0.309	0.293	-5
Prehistory	0.287	0.272	-5
Professional Accounting	0.231	0.238	3
Professional Law	0.267	0.267	0
Professional Medicine	0.188	0.169	-9
Professional Psychology	0.276	0.263	-4
Public Relations	0.345	0.354	2
Security Studies	0.196	0.188	-4
Sociology	0.249	0.249	0
US Foreign Policy	0.190	0.220	15
Virology	0.289	0.247	-14
World Religions	0.210	0.210	0
Average	0.246982	0.245456	

Table 23: Performance comparison between T5-3B (baseline) and the T5-3B model with the Creativity Adapter loaded in, on the MMLU benchmark domains (Continued).