Rethinking the Role of Scale for In-Context Learning: An Interpretability-based Case Study at 66 Billion Scale

Hritik Bansal[1*] **Karthik Gopalakrishnan**[2†] **Saket Dingliwal**[2] **Sravan Bodapati**[2]
Katrin Kirchhoff[2] **Dan Roth**[2]
[1]University of California, Los Angeles [2]AWS AI Labs
hbansal@cs.ucla.edu
{karthgop, skdin, sravanb, katrinki, drot}@amazon.com

Abstract

Language models have been shown to perform better with an increase in scale on a wide variety of tasks via the in-context learning paradigm. In this paper, we investigate the hypothesis that the ability of a large language model to in-context learn-perform a task is not uniformly spread across all of its underlying components. Using a 66 billion parameter language model (OPT-66B) across a diverse set of 14 downstream tasks, we find this is indeed the case: $\sim$70% of attention heads and $\sim$20% of feed forward networks can be removed with minimal decline in task performance. We find substantial overlap in the set of attention heads (un)important for in-context learning across tasks and number of in-context examples. We also address our hypothesis through a task-agnostic lens, finding that a small set of attention heads in OPT-66B score highly on their ability to perform primitive *induction* operations associated with in-context learning, namely, prefix matching and copying. These *induction* heads overlap with task-specific important heads, suggesting that induction heads are among the heads capable of more sophisticated behaviors associated with in-context learning. Overall, our study provides several insights that indicate large language models may be under-trained to perform in-context learning and opens up questions on how to pre-train language models to more effectively perform in-context learning.

1 Introduction

In recent years, large language models (LLMs) (Brown et al., 2020; Rae et al., 2021; Lieber et al., 2021; Black et al., 2022; Zhang et al., 2022; Chowdhery et al., 2022; Hoffmann et al., 2022; Smith et al., 2022) based on the Transformer architecture (Vaswani et al., 2017) pre-trained using self-supervision on web-scale textual corpora have revolutionized the field of natural language processing (NLP). At larger scales, these models demonstrate remarkable *emergent* (Wei et al., 2022) prowess in performing a wide variety of tasks without any form of fine-tuning, via the zero/few-shot in-context learning paradigm (Brown et al., 2020). In this paradigm (depicted in Figure 1), LLMs are prompted to generate output text conditioned on a few (or zero) in-context examples that form solved "input-output" pairs along with a query input.

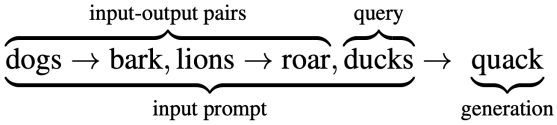


Figure 1: A sample input prompt for in-context learning and the model output.

How in-context learning works has been an open question since its advent and recent studies (Xie et al., 2021; Garg et al., 2022; Olsson et al., 2022; Min et al., 2022b) have begun scratching the surface toward better understanding the paradigm. In this paper, we empirically address the following key question:

> *Are all LLM components really needed to perform in-context learning?*

The first way we address the aforementioned question is through the lens of task-specific importance scores and structured pruning (Li et al., 2016; Molchanov et al., 2016; Anwar et al., 2017) of components underlying modern LLMs, which are primarily stacks composed of multiple high-dimensional self-attention blocks that form *multi-headed attention* and densely activated *feed forward networks* (FFNs). We pick the Open Pre-trained Transformer (OPT) (Zhang et al., 2022) model with 66B parameters for our analyses, which yield several surprising observations. We find that

[*]Work done as an intern at AWS AI Labs.
[†]Corresponding author.
Code: github.com/amazon-science/llm-interpret

important attention heads are primarily clustered in the intermediate layers and important FFNs are primarily in later (31+) layers (§4). We find that the ability to perform zero/few-shot in-context learning on almost all of a variety of 14 NLP datasets/tasks stays nearly intact when up to 70% (∼15.7B parameters in OPT-66B) of the attention heads are removed (§5.1). The attention heads that are (un)important for in-context learning also seem to overlap across tasks (§6.1) and shots (§6.2), and pruning of attention heads based on a "universal" importance order computed using all 14 datasets generalizes to varying degrees on out-of-distribution datasets (§6.1.2). These observations indicate that a common task-agnostic subset of the attention heads are responsible for in-context learning. We also find that only up to 20% of the FFNs (∼8.5B parameters) can be removed with minimal decline in zero/few-shot in-context learning performance (§5.2), indicating the importance of FFNs toward in-context learning. In addition, when accounting for the interplay between attention heads and FFNs by jointly pruning both based on their individual importance orders, the newly obtained inflection points to in-context learning performance do not deviate much from the inflection points obtained via standalone pruning (§5.3).

The second way we address the aforementioned question is by quantifying the capacity of all attention heads in OPT-66B to perform a subset of task-agnostic primitive operations associated with in-context learning, namely, *prefix matching* and *copying*: explicitly searching for a prior occurrence of the current token in-context and copying over its suffix. Elhage et al. (2021) and Olsson et al. (2022) developed a mathematical framework to reverse-engineer a Transformer and also find such heads, termed *induction heads*, and explored the hypothesis that such heads drive in-context learning with model sizes up to 13B parameters in a mostly task-agnostic fashion. Using this framework, we compute task-agnostic scores for prefix matching and copying for each attention head and find that a small set of heads in OPT-66B have non-trivial scores for both primitives (§6.3). Qualitative inspection and quantitative analyses show that these heads overlap (to varying degrees) with the ones identified earlier to be important for in-context learning via our set of 14 NLP datasets/tasks, suggesting that induction heads are capable of more sophisticated behaviors associated with in-context learning such as latent concept matching but are not the only heads with such capabilities (§6.3.1).

Overall, our study provides several insights about in-context learning at massive scale using both task-specific and task-agnostic settings. In a world of ever increasing language model sizes, we believe these insights serve as a strong foundation for researchers and practitioners in language modeling to build and leverage compact language models that can also demonstrate emergent abilities.

2 Background & Methods

In this section, we briefly describe the Open Pre-trained Transformer (OPT) (Zhang et al., 2022) model used for our experiments, provide background on in-context learning and the mathematical formulation of induction heads by Olsson et al. (2022) that we leverage, and describe our adaptation of oracle and gradient-based importance score formulations for in-context learning.

2.1 Open Pre-trained Transformer (OPT)

OPT is a suite of language models of varying sizes that are aimed at serving as open replicas of GPT-3. The largest openly accessible model from this suite is OPT-66B with 66 billion parameters, which was also the largest publicly available dense decoder-only language model at the time of our experiments. Hence, we chose OPT-66B for our study.

Architecture: Consider a tokenized input sentence to OPT, $\mathbf{X} \in \mathbb{R}^{N \times d_e}$, where N is the number of tokens in the sentence and d_e is the embedding dimension. The input is processed by multiple decoder layers consisting of *multi-headed attention* (MHA) blocks, *layer norm* (LN) and *feed forward networks* (FFN), followed by a final FFN to produce logits over the output vocabulary. The decoder layers can be formally expressed as follows:

$$\mathbf{t}^{(\ell+1)} = \mathbf{z}^{\ell} + \text{MHA}^{\ell}(\text{LN}^{\ell}(\mathbf{z}^{\ell})) \quad (1)$$

$$\mathbf{z}^{(\ell+1)} = \mathbf{t}^{(\ell+1)} + \text{FFN}^{\ell}(\mathbf{t}^{(\ell+1)}) \quad (2)$$

where $\mathbf{z}^1 = \mathbf{X}$, and (1) & (2) are the residual connections corresponding to the MHA and FFN in layer $\ell >= 1$ respectively. OPT-66B was pre-trained with a maximum sequence length of 2048 and embedding dimension $d_e = 9216$.

MHA: In an MHA block, H attention heads are applied in parallel to the input and their outputs are concatenated. In OPT-66B, there are $H = 72$ attention heads of dimension $d_h = 128$ in every layer ℓ.

An individual attention head h in layer ℓ consists of three learnable matrices, $\mathbf{W}_k^h, \mathbf{W}_q^h, \mathbf{W}_v^h \in \mathbb{R}^{d_e \times d_h}$, all unique to the head, such that it applies self-attention $A^h(.)$ on the input, where $d_h = d_e/H$. Formally, for input $\mathbf{M}$ in layer ℓ:

$$\text{MHA}^\ell(\mathbf{M}) = [A^1(\mathbf{M}); \cdots; A^H(\mathbf{M})]\mathbf{W}_o^\ell \quad (3)$$

$$A^h(\mathbf{M}) = s^h(\mathbf{M})\mathbf{M}\mathbf{W}_v^h \quad (4)$$

$$s^h(\mathbf{M}) = \sigma\Big(\frac{\mathbf{M}\mathbf{W}_q^h(\mathbf{W}_k^h)^T\mathbf{M}^T}{\sqrt{d_h}}\Big) \quad (5)$$

where σ is the softmax function and $\mathbf{W}_o^\ell \in \mathbb{R}^{d_e \times d_e}$ is a learnable output matrix unique to the MHA block in layer ℓ. To ensure OPT is auto-regressive, the output of $s^h(.)$ is masked to prevent the dependence of the hidden state of the token i, $z_i^\ell \in \mathbb{R}^{d_e}$, on the tokens that lie to the right of it in indices $\{i+1, \ldots, N\}$.

To remove a head h in layer ℓ in practice, we set $A^h(\mathbf{M})$ to be the zero matrix in Equation (3). This implies that $\mathbf{W}_k^h, \mathbf{W}_q^h, \mathbf{W}_v^h$ can be entirely removed, and the corresponding d_h rows in $\mathbf{W}_o^\ell$ can also be removed. In total, there are 4608 attention heads across 64 layers in OPT-66B that constitute 21.7B of the total 66B parameters.

FFN: Each layer ℓ consists of a feed forward network (FFN) parameterized by a high-dimensional projection matrix, $\mathbf{W}_1^\ell \in \mathbb{R}^{d_e \times d}$ followed by a low-dimensional projection matrix, $\mathbf{W}_2^\ell \in \mathbb{R}^{d \times d_e}$ where $d = 36864$ for OPT-66B. Formally, for input $\mathbf{M}$ in layer ℓ:

$$\text{FFN}^\ell(\mathbf{M}) = \text{ReLU}(\text{LN}^\ell(\mathbf{M})\mathbf{W}_1^\ell)\mathbf{W}_2^\ell \quad (6)$$

where ReLU is the rectified linear unit activation function and LN is the layer norm.

To remove an FFN in layer ℓ in practice, we set $\text{FFN}^\ell(\mathbf{M})$ to be the zero matrix in Equation (6). This implies $\mathbf{W}_1^\ell$, $\mathbf{W}_2^\ell$ and the layer norm $\text{LN}^\ell(.)$ for the FFN can be entirely removed. In total, FFNs constitute 43.4B parameters in OPT-66B.

2.2 In-Context Learning & Induction Heads

With increasingly larger language models being trained in recent years, a new paradigm of learning termed *in-context learning* (Brown et al., 2020) has become popular. In this paradigm, language models perform tasks by being prompted to generate output text conditioned on a few (or zero) in-context training examples that form solved "input-output" pairs for the task along with a query input. There are no gradient-based model updates involved as this is a purely inference-time paradigm. Figure 1 illustrates the paradigm for the task of identifying the sound that an animal makes. In some cases, tasks can also be accompanied by task descriptions/templates to help prime the language model better, e.g., zero-shot translating from English to German using the prompt:

> *English phrase: I like dogs.*
> *German phrase:*

While these examples involve learning and relying on latent concepts during inference, few-shot in-context learning can additionally involve explicit primitive interactions between the in-context examples. For example, with the prompt:

> *English phrase: I like dogs.*
> *German phrase: ich mag Hunde.*
> *English phrase: I like ducks.*
> *German phrase:*

the model may rely on prior in-context translations of the tokens *I* and *like* when performing the task for the query input. Olsson et al. (2022) develop a mathematical framework toward better understanding such mechanics, starting off with a task-agnostic formulation of in-context learning as the ability of a model to better predict tokens later in the context than the tokens earlier (Kaplan et al., 2020) and defining a heuristic for it. They define a set of task-agnostic primitive operations that reflect the kind of interactions we refer to in the above example, namely, *prefix matching* and *copying*. These operations are defined in a simplistic fashion on a repeated sequence of randomly generated tokens: explicitly searching for a prior occurrence of the current token in-context and copying over its suffix. The heads that are capable of performing these operations are termed *induction heads*. Figure 2 depicts these operations for a repeated sequence of tokens. While these operations are intertwined in practice, the capacity of attention heads to *independently* perform them is computed with the scoring algorithms described in detail in Appendix A.4.

2.3 Importance Scores

Consider a model $\mathcal{M}$ and a dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_L\}$ and $\mathcal{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_L\}$ such that $\mathbf{x}_i$ represents a prompt with few (or zero) in-context training examples along with a query input and $\mathbf{y}_i$ represents the corresponding target output sequence. We define and compute importance

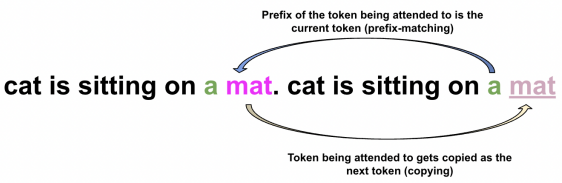


Figure 2: Prefix matching and copying depicted at a given time-step for a repeated sequence of tokens.

scores for model components using such datasets to quantify their relative contributions to the model's ability to perform in-context learning.

2.3.1 Oracle

Let $\mathcal{P}_{\mathcal{M}}(\mathcal{D})$ denote a dataset/task-specific performance metric, e.g., accuracy. Given dataset $\mathcal{D}$, the oracle importance score of a component $\mathcal{C}$ in $\mathcal{M}$ is computed as follows:

$$IS_{\mathcal{C}}(\mathcal{D}) = \mathcal{P}_{\mathcal{M}}(\mathcal{D}) - \mathcal{P}_{\mathcal{M}_{\setminus\mathcal{C}}}(\mathcal{D}) \tag{7}$$

where $\mathcal{M}_{\setminus\mathcal{C}}$ denotes the resultant model when $\mathcal{C}$ is pruned from $\mathcal{M}$. Clearly, if pruning a component leads to poor model performance on the task, it must be important for the task. Similarly, if there is no difference or an improvement in performance upon pruning a component, it must be unimportant.

Computing oracle importance scores for K model components requires us to perform $\mathcal{O}(K)$ evaluations for each dataset $\mathcal{D}$. Given that OPT-66B only has 64 FFNs, this is computationally feasible. However, it becomes infeasible to score the 4608 attention heads in OPT-66B in this manner, so we adopt the second method described below to compute importance scores for attention heads.

2.3.2 Gradient-based

Given dataset $\mathcal{D}$, the gradient-based importance score (Molchanov et al., 2016; Michel et al., 2019) of an attention head h captures the expected sensitivity of the model to h and is computed as follows:

$$IS_h(\mathcal{D}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})} \left| A^h([\mathbf{x};\mathbf{y}])^T \frac{\partial \mathcal{L}(\mathbf{y}|\mathbf{x})}{\partial A^h([\mathbf{x};\mathbf{y}])} \right| \tag{8}$$

where ; is the concatenation operator, $(\mathbf{x},\mathbf{y}) \sim \mathcal{D}$ such that $\mathbf{x}$ is a sequence of T_x tokens $x_{1:T_x}$, $\mathbf{y}$ is a sequence of T_y tokens $y_{1:T_y}$, A^h is defined in (4) and the loss term in (8) is computed using the auto-regressive decomposition of the log-likelihood:

$$\mathcal{L}(\mathbf{y}|\mathbf{x}) = -\frac{1}{T_y} \sum_{j=1}^{j=T_y} \log(p(y_j|\mathbf{x}, y_{1:j-1})) \tag{9}$$

These importance scores can be efficiently computed for all heads by simply performing a single forward and backward pass over the model with $\mathcal{D}$.

We also define the aggregated importance score of an attention head on a set of datasets $\mathbb{S} = \{\mathcal{D}_1, \cdots, \mathcal{D}_K\}$ as follows:

$$IS_h(\mathbb{S}) = \mathbb{E}_{\mathcal{D}\sim\mathbb{S}}\left[IS_h(\mathcal{D})\right] \tag{10}$$

3 Experimental Setup

We perform our experiments with OPT-66B on a variety of 14 NLP datasets/tasks. For consistency in the evaluation metric, we report accuracy on all tasks. Our choice of datasets and metric is in line with that of Meta AI in the OPT paper (Zhang et al., 2022). The datasets include ARC Easy and Challenge (Clark et al., 2018) and OpenBookQA (Mihaylov et al., 2018) for advanced question-answering, HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020) and Winogrande (Sakaguchi et al., 2021) for various forms of commonsense reasoning, and the following datasets from the standard SuperGLUE benchmark (Wang et al., 2019): BoolQ, CB, COPA, MultiRC, ReCoRD, RTE, WiC, and WSC. For a subset of experiments involving evaluation for out-of-distribution generalization, we also use 2 additional datasets: MathQA (Amini et al., 2019) and LAMBADA (Paperno et al., 2016).

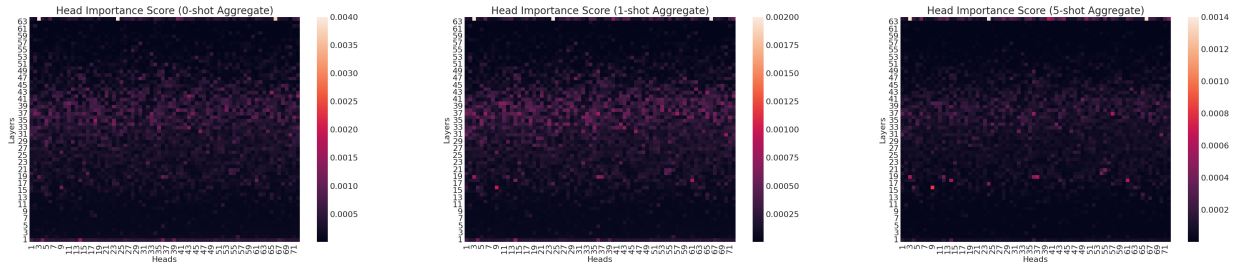
We use a modified version of Eleuther AI's *lm-evaluation-harness* framework (Gao et al., 2021) for our experiments. The default framework samples in-context examples at random, which we use as-is without modification.

4 Importance Scores for OPT-66B

In this section, we present importance scores for all attention heads and feed forward networks in OPT-66B toward performing zero-shot and few-shot (1-shot and 5-shot) in-context learning.

4.1 Attention Heads

Figure 3 depicts heatmaps of the head importance scores averaged across all tasks (as described in §2.3.2) in the zero-shot, one-shot and five-shot settings. Task-specific heatmaps are provided in Appendix A.1. We observe that the important attention heads are primarily clustered in the intermediate layers of OPT-66B in both the task-averaged and task-specific cases. We also observe some overlap in the most important attention heads across the



(a) Zero-shot (b) One-shot (c) Five-shot

Figure 3: Attention head aggregate importance score heatmaps for in-context learning with OPT-66B.

different zero/few-shot settings. This is further confirmed in follow-up analysis in §6.2.

4.2 Feed Forward Networks

We compute oracle importance scores (both task-specific and averaged across tasks) for each of the 64 FFNs in OPT-66B as described in §2.3.1 in the zero-shot and few-shot settings and present them in Figure 4. Note that if the importance score, i.e., accuracy difference, for an FFN f is near zero, f is redundant. If it is negative, the model is better off without f, and if it is positive, f is important.

We observe that in the zero-shot and one-shot settings, the removal of any FFN in the early (1-30) layers of OPT-66B either gives comparable or better performance for a vast majority of tasks. In the five-shot setting however, both the early and later layers seem to have important FFNs for most tasks. We also generally observe high variance in FFN importance scores in later layers. We particularly note high variance for WSC and MultiRC, observing that removal of some individual FFNs can lead to absolute accuracy improvements/degradation of up to 20%! One hypothesis for the cause for such variance could be the more challenging nature of some of these tasks such as MultiRC, which has fairly large sequence lengths, and such FFNs might be relied upon to be able to process challenging task instances. We leave further investigation into the cause for this variance for future work.

5 Iterative Pruning

The observations in the previous section (§4) indicated that the ability to in-context learn-perform a task is not uniformly spread across all of OPT-66B's underlying components. In this section, we build on these observations and assess to what extent we can remove *multiple* attention heads and FFNs with minimal decline in task performance.

5.1 Removing Attention Heads

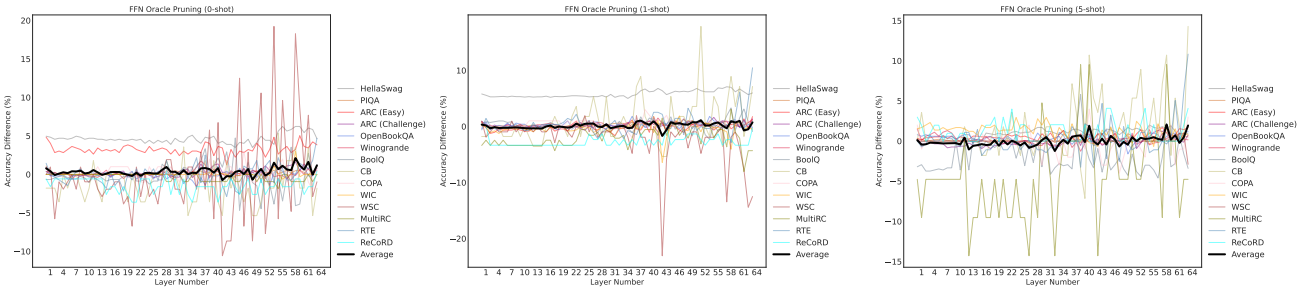
For each task in each (zero-shot, one-shot and five-shot) in-context learning setting, we sort all attention heads in OPT-66B in ascending order by importance score (§4.1). We then remove attention heads in an iterative fashion from the start of this order, 10% at a time, and re-evaluate task performance after each removal.[1] Figure 5 depicts the resulting accuracy trends for each task and the trends when averaged across all tasks.

We observe that the average accuracy across tasks does not change much up until ∼70% of the attention heads are removed. A fine-grained look at the individual tasks also mostly shows similar trends, with accuracy staying fairly intact until a large proportion of the attention heads are removed. Some oddities include tasks such as WSC and CB, wherein we observe that the zero-shot accuracy actually increases after the removal of 70% of the attention heads. As noted by Zhang et al. (2022), we also observe low absolute accuracy values on MultiRC for the full model and upon pruning.

5.2 Removing FFNs

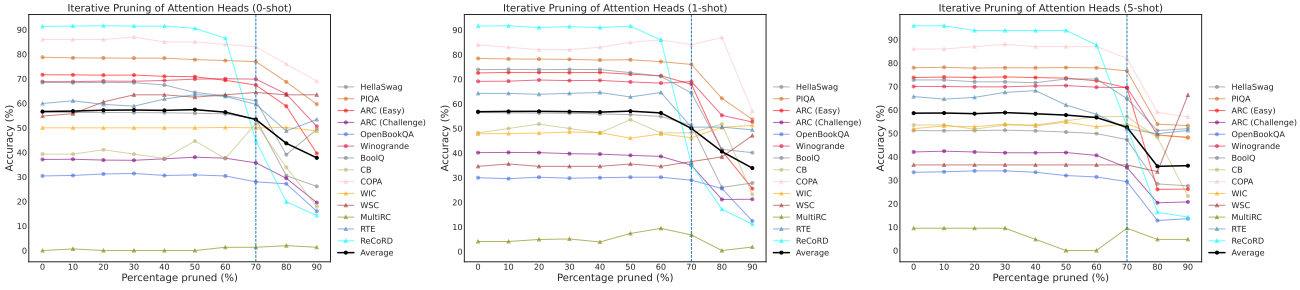
For each task in each (zero-shot, one-shot and five-shot) in-context learning setting, we sort all FFNs in OPT-66B in ascending order by importance score (§4.2). We then remove FFNs in an iterative fashion from the start of this order, 10% at a time, and re-evaluate task performance after each removal. Figure 6 depicts the resulting accuracy trends for each task and the trends when averaged across all tasks. We observe that in the zero-shot setting, the average accuracy across tasks does not change up until ∼20% of the FFNs are removed.

[1] We do not remove attention heads one at a time and re-evaluate given the number of heads and evaluation cost.



(a) Zero-shot (b) One-shot (c) Five-shot

Figure 4: Feed forward network (FFN) oracle importance scores for in-context learning with OPT-66B. Each FFN is knocked off independently to compute these scores, i.e., the curves are discrete and **not** cumulative.



(a) Zero-shot (b) One-shot (c) Five-shot

Figure 5: Effect on in-context learning accuracy when removing attention heads in OPT-66B in an iterative manner based on task-specific and shot-specific importance scores.

For some tasks such as PIQA, Winogrande and RTE, the accuracy does not change even if 30% of the FFNs (∼13B of the 66B parameters) are removed. We also observe that the inflection point after which we observe a sharp decline in accuracy changes to 10% for the few-shot settings. Overall, these observations indicate that FFNs play a critical role toward in-context learning.

5.3 Combined Removal of Heads & FFNs

We now investigate whether the inflection points to in-context learning performance when removing either attention heads or FFNs in an iterative fashion still hold when removing them in *tandem*. Figure 7 depicts the average accuracy of all tasks on joint iterative removal of attention heads and FFNs, 10% at a time, based on their respective importance scores in the zero and few-shot settings.

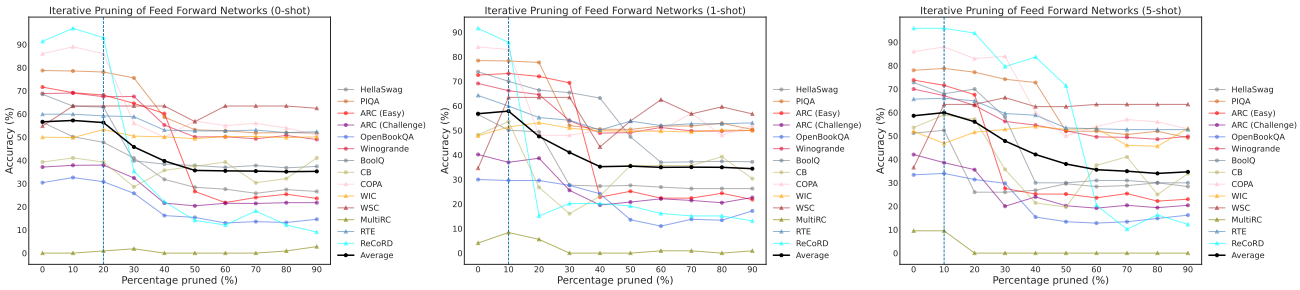
We observe that the removal of 70% of the attention heads (∼15.7B parameters) and 20% of the FFNs (∼8.5B parameters) leads to a mere 5% absolute drop in the average zero-shot accuracy. In the one-shot setting, the absolute drop in accuracy is 6% on removing 70% of the attention heads and 10% of the FFNs. In the five-shot setting, the absolute drop in accuracy is 4% on removing 60% of the attention heads and 20% of the FFNs. Overall, these new inflection points have deviated by at most a factor of 10% absolute, which may be attributed to the interplay between heads and FFNs.

6 Detailed Analysis of Attention Heads

In this section, we perform a detailed analysis of the attention heads in OPT-66B, given that in-context learning is auto-regressive in nature and attention heads explicitly encode cross-token interactions. We perform cross-task analyses to understand whether various tasks share (un)important attention heads. We also perform cross-shot analyses to study whether the (un)important attention heads for a task are shared across the zero-shot and few-shot settings. We finally quantify the capacity of the attention heads to perform task-agnostic induction operations (defined in §2.2) and study correlations with task-specific importance scores.

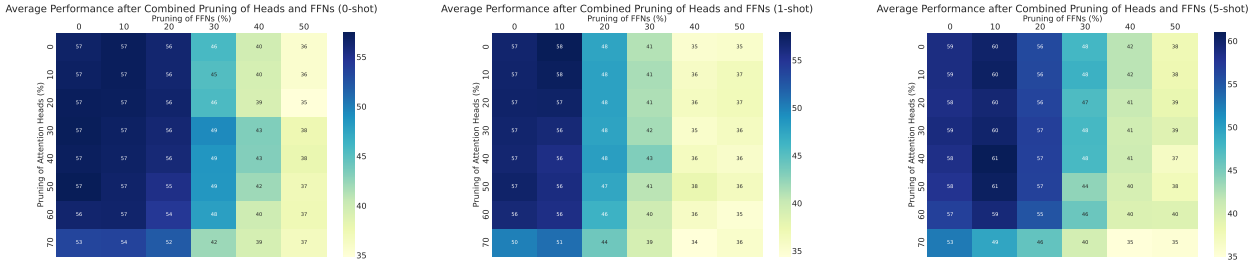
6.1 Cross-Task Analysis

Michel et al. (2019) found preliminary empirical evidence of the existence of "universally" important



(a) Zero-shot (b) One-shot (c) Five-shot

Figure 6: Effect on in-context learning accuracy when removing feed forward networks (FFNs) in OPT-66B in an iterative manner based on task-specific and shot-specific importance scores.



(a) Zero-shot (b) One-shot (c) Five-shot

Figure 7: Effect on average in-context learning accuracy when removing **both** attention heads and feed forward networks (FFNs) in OPT-66B in an iterative manner based on shot-specific task-aggregate importance scores.

attention heads in trained task-specific Transformer and BERT models via evaluating on out-of-domain test sets for machine translation and natural language inference respectively. With similar motivation, we study if the (un)important attention heads identified in various in-context learning settings for OPT-66B are shared across tasks.

6.1.1 Spearman's Rank Correlation

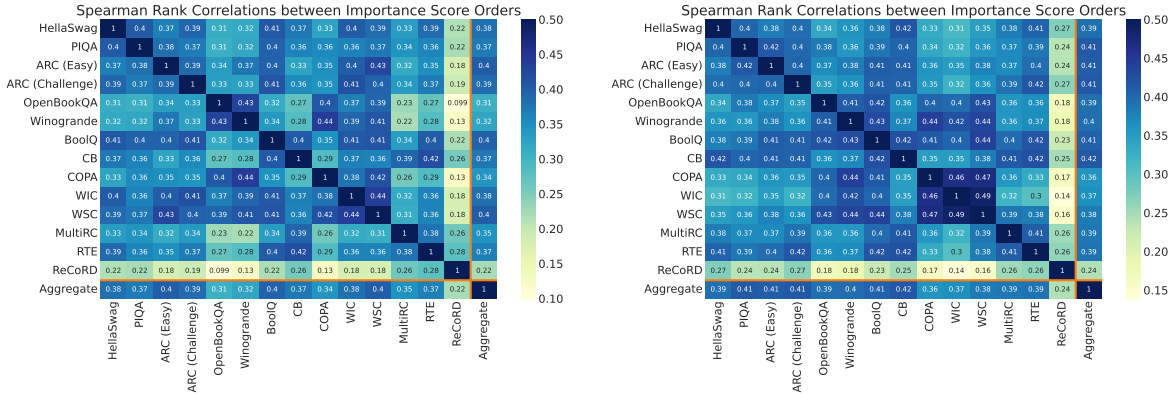
We assess overlap in (un)important attention heads across tasks by sorting task-specific head importance scores to get head importance rankings and computing the Spearman's rank correlation coefficient (SRCC) between the rankings for every pair of tasks in the zero-shot and few-shot settings. We also sort the task-aggregate head importance scores to get the aggregate ranking and compute the SRCC against the ranking for every constituent task. All pairwise correlations for the zero and one-shot settings are depicted in Figure 8, and depicted for the five-shot setting in Appendix A.2.

In both zero and few-shot settings, we observe statistically significant ($p < 0.01$) positive correlations in the head importance rankings for every pair of tasks, as well as between every task's ranking and the aggregate ranking. This indicates that the set of (un)important attention heads are clustered together across tasks. We also observe seemingly lower magnitude SRCC values between every task and ReCoRD, a long reading comprehension task which requires commonsense reasoning, indicating the amount of head overlap is proportionally lower.

6.1.2 Generalization Trends

We now investigate how well head importance rankings generalize across tasks by studying accuracy trends for a task when pruning using its own head importance ranking as well as using different head importance rankings. We perform this investigation for two sets of tasks.

The first set of tasks we investigate were used to compute the aggregate ranking: COPA, Winogrande and ReCoRD. For each of these 3 tasks, we use the self-ranking, aggregate ranking and the rankings from the tasks which share the *highest* and *lowest* SRCC with them based on Figure 8. For instance, we see that ReCoRD has the highest and lowest SRCCs with RTE and OpenBookQA respectively in the zero-shot setting, so we consider the impact of pruning based on their head



(a) Zero-shot (b) One-shot

Figure 8: Spearman's rank correlation coefficients between the attention head importance rankings for various tasks. All p-values < 0.01. Coefficients in the five-shot setting are provided in Appendix A.2.

importance rankings on ReCoRD accuracy. Figures 9a, 9b and 9c depict the accuracy trends for these 3 tasks in the zero-shot setting. We observe that the accuracy on all 3 tasks when pruning using the rankings described is almost unaffected up to the 50% mark. We then observe a sharp decline in accuracy on COPA and Winogrande when the model is pruned to the 70% mark using the ranking identified via ReCoRD, the task with the lowest SRCC (0.13) with both COPA and Winogrande in Figure 8a. This indicates that even if the rankings vary between ReCoRD and COPA/Winogrande (as reflected in the low magnitude of the SRCC score), the set of attention heads important for zero-shot learning with ReCoRD are important for COPA/Winogrande too. To further verify this, we calculated and found 71% and 76% overlap between the top 30% important attention heads for ReCoRD-COPA and ReCoRD-Winogrande respectively. We also surprisingly observe that the accuracy on ReCoRD at the 70% pruning mark is better using the aggregate ranking than using the ranking for ReCoRD itself. We observe similar trends in the few-shot settings as well, depicted in Appendix A.3. A key distinction we observe relative to the zero-shot setting is that the decline/divergence in accuracy beyond the 50% pruning mark using the ReCoRD ranking is less sharp for COPA and Winogrande in the one-shot setting and fades away in the five-shot setting, indicating a convergence of important attention heads across tasks.

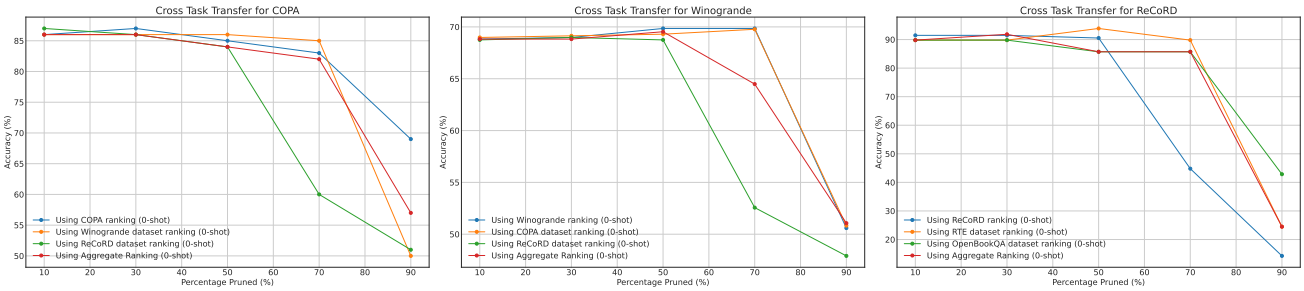
The second set of tasks we investigate were unseen, i.e., not used to compute the aggregate ranking: MathQA and LAMBADA. For these tasks, we analyze accuracy trends when pruning using the self-ranking and aggregate ranking. Figures 9d and 9e depict their accuracy trends in the zero-shot setting. As expected, we observe that the self-ranking accuracy curves are somewhat higher than the aggregate ranking accuracy curves in general across both tasks. For MathQA, we also observe that the absolute difference in accuracy for both cases is within 1-2%. These indicate that the aggregate rankings generalize well to MathQA but not as much to LAMBADA. We observe similar trends in the few-shot settings, depicted in Appendix A.3.

	0-shot	1-shot	5-shot
0-shot	1	$0.39_{\mathbf{0.001}}$	$0.37_{\mathbf{0.001}}$
1-shot	$0.39_{\mathbf{0.001}}$	1	$0.41_{\mathbf{0.001}}$
5-shot	$0.37_{\mathbf{0.001}}$	$0.41_{\mathbf{0.001}}$	1

Table 1: Spearman's rank correlation coefficient (SRCC) between attention head importance rankings for different in-context learning settings. Each cell depicts the mean and variance (subscript) in SRCC across 14 tasks. p-value < 0.01 for every correlation pair.

6.2 Cross-Shot Analysis

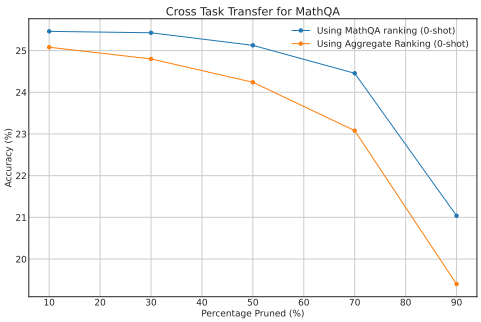
To see if the attention heads that are identified to be (un)important for a task are shared across the different zero and few-shot settings, we compute Spearman's rank correlation coefficient (SRCC) between the cross-shot head importance rankings for each task and compute the mean and variance across all 14 tasks. Table 1 depicts these metrics. We observe that the SRCC is higher for rankings *within* the few-shot setting (0.41 between 1-shot and 5-shot) than for rankings *across* the zero and



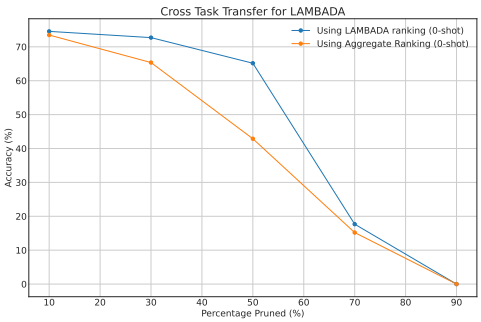
(a) COPA

(b) Winogrande

(c) ReCoRD



(d) MathQA



(e) LAMBADA

Figure 9: Cross-task transfer of attention head importance rankings as measured by impact of pruning on accuracy in the zero-shot setting.

few-shot settings. This matches the intuition that a similar set of heads must be important within the different few-shot settings than across the zero-shot and any of the few-shot settings. However, we also see that the SRCC magnitudes for the latter (0.39 and 0.37) are not very far off. In totality, these indicate non-trivial overlap in the (un)important attention heads for tasks across shots.

6.3 Induction Heads in OPT-66B

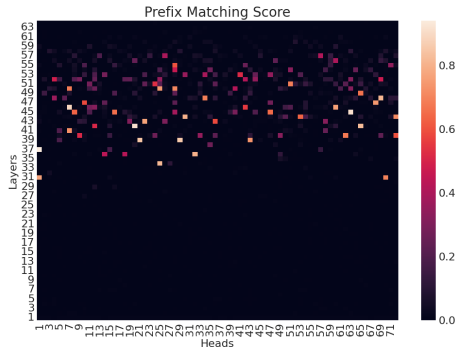
We look for induction heads in OPT-66B by quantifying the capacity of all attention heads to perform prefix matching and copying using random input sequences in a task-agnostic fashion, following the definition and algorithms by Olsson et al. (2022) discussed in §2.2 and Appendix A.4.

Figures 10a and 10b depict the prefix matching and copying score heatmaps respectively for OPT-66B. We observe that a small subset of attention heads in OPT-66B have high prefix matching scores, located in the upper layers (31+) of the model. On the other hand, there are a relatively larger number of attention heads with high copying scores, although the vast majority of these are also located in the upper layers (41+). When seen in conjunction, these observations indicate that there is a sparse set of attention heads that are capable of performing both primitive operations and thus can be deemed plausible induction heads.

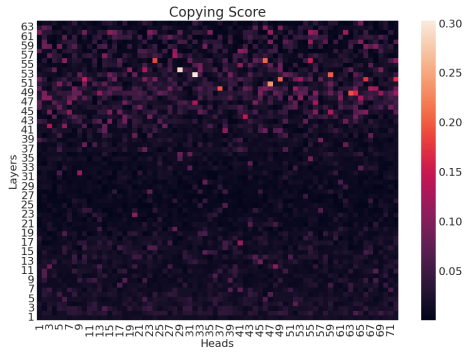
6.3.1 Are Induction Heads Important?

We previously (§4.1) computed task-specific and task-aggregated importance scores for attention heads in the zero-shot and few-shot in-context learning settings. By virtue of being important for the challenging downstream tasks we consider, these attention heads are capable of sophisticated and latent behaviors associated with in-context learning that go well beyond the basic primitives of explicit prefix matching and copying. We now study whether induction heads overlap with attention heads deemed important for our chosen tasks.

A qualitative comparison of the heatmaps in Figure 10 against the heatmaps in Figure 3 indicate that induction heads do overlap with task-aggregated important attention heads. To better facilitate this comparison, we first formalize the total capacity of a model to perform prefix matching (or copying) to be the sum of the respective scores for individual attention heads in the model. We then investigate how much of this capacity is retained when attention heads are pruned in the order of least important heads first. Figure 11 depicts this comparison. We observe that much of the total prefix matching

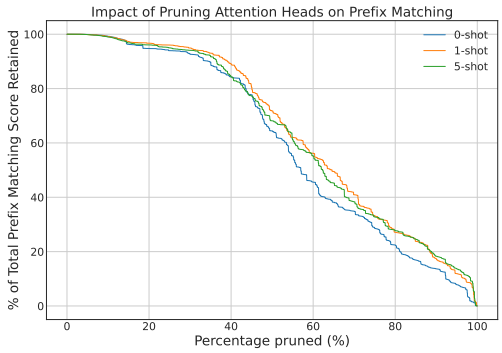


(a) Prefix Matching

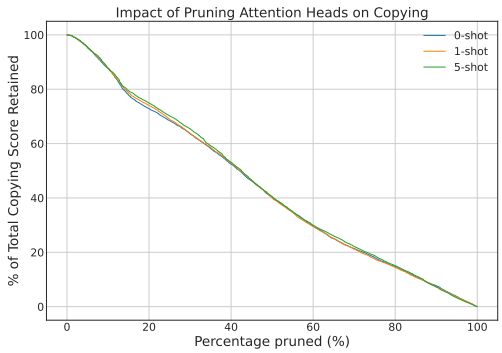


(b) Copying

Figure 10: Attention head prefix matching and copying score heatmaps for OPT-66B.



(a) Prefix Matching



(b) Copying

Figure 11: Total inductive (prefix matching / copying) capacity retained as a function of percentage of attention heads pruned, where heads are pruned based on task-aggregate importance score rankings in the order of least important first.

score is retained when 20% of the least important heads are removed, with the slope of decline becoming sharp only after the 40% pruning mark. This indicates that unimportant heads also have low prefix matching scores. We also observe that the prefix matching scores are generally higher for heads important for few-shot in-context learning than for heads important for zero-shot learning. On the other hand, we observe across the zero-shot and few-shot settings that the total copying score retained on pruning attention heads rapidly and consistently declines, indicating that even unimportant heads have a non-trivial capacity to perform copying. When seen in conjunction, these observations indicate that induction heads in OPT-66B are capable of sophisticated behaviors associated with in-context learning popular downstream NLP tasks and reinforce the induction head generality arguments Olsson et al. (2022) make in the context of smaller models with stylized and synthetic tasks. We provide further evidence of this in Appendix A.5 with per-task plots which showcase that some tasks rely on induction heads more than other tasks.

7 Related Work

There has been an interest in effectively *leveraging* the in-context learning paradigm (Zhao et al., 2021; Holtzman et al., 2021; Min et al., 2022a; Liu et al., 2022; Lu et al., 2022; Rubin et al., 2022; Mishra et al., 2022) ever since its introduction by Brown et al. (2020), but there have been relatively fewer studies toward better *understanding* the paradigm itself. Xie et al. (2021) cast in-context learning as implicit Bayesian inference where the language model implicitly infers a shared concept among in-context examples when making a prediction, showing that this occurs when the pre-training distribution is a mixture of Hidden Markov Models (HMMs) over latent concepts despite a mismatch with the prompt distribution. Min et al. (2022b) study the role of the in-context examples themselves, finding that the ground-truth labels are not needed in the examples and that the more important drivers are provision of the label space, the distribu-

tion of the input text and the overall format of the sequence. Garg et al. (2022) showcase that Transformer models trained from scratch can in-context learn the class of linear functions with performance comparable to the optimal least squares estimator even under distribution shifts. Razeghi et al. (2022) showcase that in-context learning performance is correlated strongly with term frequencies in the pre-training corpora used. Olsson et al. (2022) consider an alternate framing of in-context learning as the ability of a language model to better predict tokens later in the context than tokens earlier and hypothesize the existence of induction heads that are responsible for in-context learning. Chan et al. (2022) explore training Transformers separately on controlled stimuli and natural language to showcase that Transformers exhibit striking differences in their generalization from in-context vs. in-weights information.

Several works have also focused on analyzing and interpreting how attention works. Vig and Belinkov (2019) performed a study on GPT-2 *small* with Wikipedia sentences, finding that attention targets different parts of speech at different layer depths and aligns with dependency relations most strongly in the middle layers. Tenney et al. (2019) showcase that BERT encodes the classical NLP pipeline in an interpretable way across layers. There are works relying on different formulations for head importance, such as layer-wise relevance propagation (Voita et al., 2019), gradient-based importance and oracle knock-off importance (Michel et al., 2019), with small task-specific trained models and report the existence of specialized heads. Given the recent trend of increasing model scale (Lieber et al., 2021; Chowdhery et al., 2022; Smith et al., 2022; Rae et al., 2021; Hoffmann et al., 2022) toward tuning-free general-purpose language models that exhibit emergent in-context learning abilities, we draw and build on prior work to understand just how much scale is really needed and/or used for in-context learning *downstream*, an aspect somewhat eclipsed by the focus on the pre-training loss curve in scaling laws (Hoffmann et al., 2022). It is also worth noting that some of our empirical observations rely on a simple greedy approach to training-free pruning since our focus was not to optimally prune a language model with respect to performing in-context learning. Li et al. (2021) show the greedy approach is sub-optimal and produces under-estimates and Halabi et al. (2022) account for the need to re-compute importance scores after removal of each attention head or FFN by formulating pruning as weakly sub-modular maximization.

8 Conclusion & Future Work

In this paper, we studied the efficacy of attention heads and feed forward networks (FFNs) in a large language model (OPT-66B) in performing in-context learning in both task-specific and task-agnostic settings. We observed that while in-context learning may have emerged via self-supervised pre-training at scale, only a core nucleus of attention heads and FFNs seem to be important for in-context learning across a wide variety of downstream tasks. We observed that a small set of attention heads have the capacity to perform task-agnostic primitive induction operations associated with in-context learning, namely prefix matching and copying. We also saw that these induction heads overlap with task-specific important attention heads to varying degrees, indicating that induction heads are capable of more sophisticated forms of in-context learning and reinforcing claims (Olsson et al., 2022) about the generality of induction heads. Overall, our in-context learning-centric observations complement recent work (Hoffmann et al., 2022) in indicating that large language models may be under-trained and motivate several interesting directions for future work. While induction heads are formed naturally during self-supervised pre-training in its current form, we believe it may be possible to increase the number and strength of induction heads formed by defining auxiliary pre-training objectives for primitives like prefix matching and copying. This could, in turn, translate to improved in-context learning of downstream tasks, especially tasks we observed to currently rely highly on induction heads. More generally, it may also be prudent to investigate and improve pre-training regimes to increase the number of important model components (including both attention heads and FFNs) to in-context learn-perform a wide variety of downstream tasks. We hope our work spurs research toward better understanding the behavior of large language models and consequently building compact ones afresh that can also demonstrate emergent abilities.

References

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. 2017. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):1–18.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Stephanie CY Chan, Ishita Dasgupta, Junkyung Kim, Dharshan Kumaran, Andrew K Lampinen, and Felix Hill. 2022. Transformers generalize differently from information stored in context vs in weights. *arXiv preprint arXiv:2210.05675*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *arXiv preprint arXiv:2208.01066*.

Marwa El Halabi, Suraj Srinivas, and Simon Lacoste-Julien. 2022. Data-efficient structured pruning via submodular optimization. *arXiv preprint arXiv:2203.04940*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2021. Differentiable subset pruning of transformer heads. *Transactions of the Association for Computational Linguistics*, 9:1442–1459.

Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

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

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

A Appendix

A.1 Task-Specific Head Importance Scores

Figures 12, 13 and 14 depict the attention head importance scores for each task in the zero-shot, one-shot and five-shot settings respectively.

A.2 Cross-Task Analysis: Spearman's Rank Correlation

Figure 15 depicts the Spearman's rank correlation coefficients (SRCC) between the attention head importance rankings for every pair of tasks in the five-shot setting. It also depicts the SRCC between the aggregate ranking and the ranking for each constituent task.

A.3 Cross-Task Analysis: Generalization Trends

Figures 16 and 17 depict the cross-task head importance ranking generalization plots in the one-shot and five-shot settings.

A.4 Details of Prefix Matching and Copying Scores

Algorithms 1 and 2 contain pseudo-code to compute prefix matching and copying scores respectively for each attention head in OPT-66B. We follow the approach described by Olsson et al. (2022), but instead of computing scores using 10 sequences with fixed length of 25, we compute these scores using 100 sequences with varying lengths to account for OPT-66B's large maximum sequence length. As in Olsson et al. (2022), we exclude a small fraction of the most and least common tokens from the model's vocabulary and randomly sample tokens for these sequences to strip out the effects of pretraining corpora memorization from our scores and inductive behavior analyses.

For prefix matching, the high-level approach is the following: take a random sequence, repeat it 4 times, perform a forward pass and then for each head, compute the attention pattern and take the average of all attention pattern entries attending from a given token back to tokens that succeeded the same token in earlier repeats.

For copying, the high-level approach is the following: take a random sequence, directly feed the sequence through each head and compute the contribution of the head to the output logits, and then measure how much the head increased the logit of the maximally attended to token over increasing the logits of other attendable tokens at each timestep. Unlike Olsson et al. (2022), we do not scale the raw scores to be in the range of -1 to 1.

A.5 Importance of Induction Heads to Each Task

Figures 18 and 19 showcase the importance of induction heads to each task via measuring the percentage of the total prefix matching and copying capacities retained as a function of percentage of attention heads pruned, where heads are pruned based on each task's head importance ranking for each in-context learning setting (zero-shot, one-shot and five-shot) in the order of least important first. A small initial slope of decline implies that unimportant heads also have low prefix matching or copying scores while a steep initial slope of decline implies unimportant heads also have high prefix matching or copying scores. We observe differences in the slopes of decline across different tasks, with tasks like HellaSwag and ReCoRD (which have high accuracies in Figure 5) having smaller initial slopes than a task like OpenBookQA (which has relatively lower accuracy in Figure 5). When seen in conjunction, these plots not only point to the generality of induction heads to more sophisticated behaviors associated with in-context learning but also indicate that some tasks rely on induction heads more than others.

Algorithm 1 Prefix Matching Scores for Attention Heads

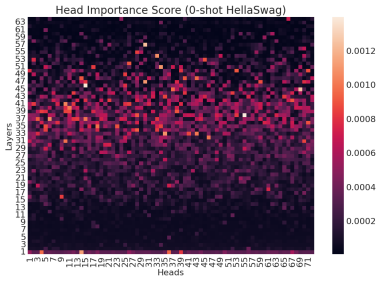
Arguments: Model $\mathcal{M}$

$model \leftarrow$ Pretrained($\mathcal{M}$)
$layers, heads \leftarrow model$.num_layers, $model$.num_heads_per_layer
$ranked\_vocab\_list \leftarrow model$.tokenizer.vocab ▷ term-frequency based vocabulary list of model
$exclude\_vocab\_size \leftarrow 0.04 \times$ len($ranked\_vocab\_list$) ▷ remove 4% most & least common tokens
$ranked\_vocab\_list \leftarrow ranked\_vocab\_list[exclude\_vocab\_size : -exclude\_vocab\_size]$
$prefix\_matching \leftarrow [\,]$
for $seed$ in $\{1 \cdots 100\}$ **do**
  $L \leftarrow 2 \times seed + 23$ ▷ ensure $4L \in [100, 892]$
  $X \leftarrow random.choice(ranked\_vocab\_list, \text{size} = L, \text{seed} = seed, \text{replace} = False)$ ▷ L length random sequence with all unique tokens
  $X \leftarrow$ repeat($X, 4$) ▷ Repeat it four times
  $Y \leftarrow model$.forward(X) ▷ Forward pass the repeated sequence
  $score \leftarrow$ zeros($layers$, $heads$) ▷ Zero matrix of shape $layers \times heads$
  for $layer$ in $layers$ **do**
    for $head$ in $heads$ **do**
      $att \leftarrow model$.get_attention($layer, head$) ▷ Shape: $4L \times 4L$
      for $token$ in $\{L + 1 \cdots 4L\}$ **do** ▷ Repetition starts from token $L + 1$
        $att\_token \leftarrow att[token]$ ▷ Shape: $4L$
        for every $prev\_token$ == $token$ **do** ▷ Look at the previous repetitions of the token
          $prefix\_score = att\_token[prev\_token + 1]$ ▷ Attention given to token whose prefix is current token
          $score[layer][head] \leftarrow score[layer][head] + prefix\_score$
        end for
      end for
      $score[layer][head] \leftarrow score[layer][head]/3L$ ▷ Normalizing by length of for loop
    end for
  end for
  $prefix\_matching$.append($score$) ▷ Prefix matching scores via one randomly generated example
end for
$prefix\_matching \leftarrow$ average($prefix\_matching$) ▷ Attention head-wise average over all examples
return $prefix\_matching$

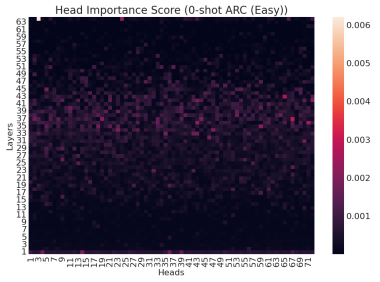
Algorithm 2 Copying Scores for Attention Heads

Arguments: Model $\mathcal{M}$
Definitions: Dimension per Head D, Vocabulary Size V

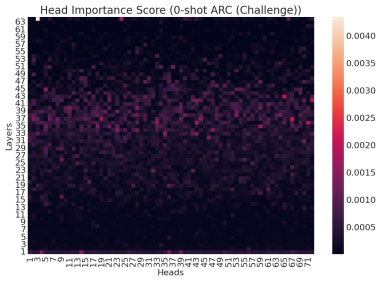
$model \leftarrow$ Pretrained($\mathcal{M}$)
$layers, heads \leftarrow model$.num_layers, $model$.num_heads_per_layer
$ranked\_vocab\_list \leftarrow model$.tokenizer.vocab ▷ term-frequency based vocabulary list of model
$exclude\_vocab\_size \leftarrow 0.04 \times$ len($ranked\_vocab\_list$) ▷ remove 4% most & least common tokens
$ranked\_vocab\_list \leftarrow ranked\_vocab\_list[exclude\_vocab\_size : -exclude\_vocab\_size]$
$copying\_score \leftarrow [\,]$
for $seed$ in $\{1 \cdots 100\}$ **do**
  $L \leftarrow 4 \times (2 \times seed + 23)$ ▷ $L \in [100, 892]$
  $X \leftarrow random.choice(ranked\_vocab\_list, \text{size} = L, \text{seed} = seed, \text{replace} = False)$ ▷ L length random sequence with all unique tokens
  $score \leftarrow$ zeros($layers$, $heads$) ▷ Zero matrix of shape $layers \times heads$
  for $layer$ in $layers$ **do**
    for $head$ in $heads$ **do**
      $attn\_layer\_head \leftarrow model$.get_attention_head($layer, head$)
      $out \leftarrow attn\_layer\_head(X)$ ▷ Shape: $L \times D$
      $attention \leftarrow model$.get_attention($layer, head$) ▷ Shape: $L \times L$
      $logits \leftarrow model$.hidden_to_vocab(out) ▷ Shape: $L \times V$
      $logits \leftarrow$ softmax($logits, dim = 1$)
      for $token$ in $\{1 \cdots L\}$ **do**
        $max\_ind \leftarrow$ argmax($attention[token]$) ▷ Index of the token being max attended to
        $attendable\_input \leftarrow X[1 : token]$ ▷ Attendable input tokens
        $attendable\_logits \leftarrow logits[token][attendable\_input]$ ▷ Logits of attendable tokens
        $mean\_of\_logits \leftarrow$ average($attendable\_logits$)
        $raised\_logits \leftarrow attendable\_logits - mean\_of\_logits$
        $relu\_raised\_logits \leftarrow$ ReLU($raised\_logits$) ▷ Computing raise in logit values
        $relu\_raised\_logit\_max\_ind \leftarrow relu\_raised\_logits[X[max\_ind]]$
        $temp\_score \leftarrow relu\_raised\_logit\_max\_ind / \text{sum}(relu\_raised\_logits)$
        $score[layer][head] \leftarrow score[layer][head] + temp\_score$
      end for
      $score[layer][head] \leftarrow score[layer][head]/L$ ▷ Normalizing by length of for loop
    end for
  end for
  $copying\_score$.append($score$) ▷ Copying scores via one randomly generated example
end for
$copying\_score \leftarrow$ average($copying\_score$) ▷ Attention head-wise average over all examples
return $copying\_score$



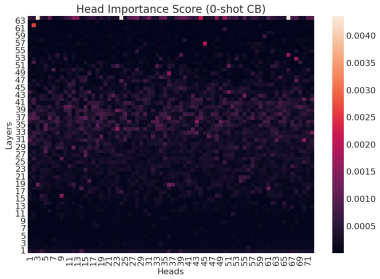
(a) HellaSwag



(b) ARC (Easy)

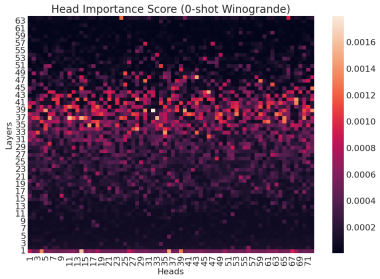


(c) ARC (Challenge)

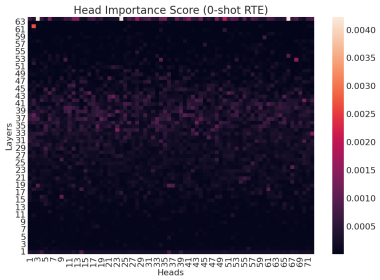


(d) CB

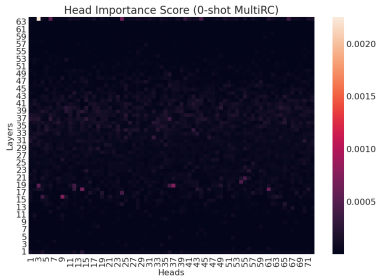
(e) BoolQ



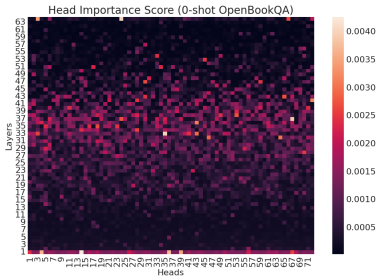
(f) Winogrande



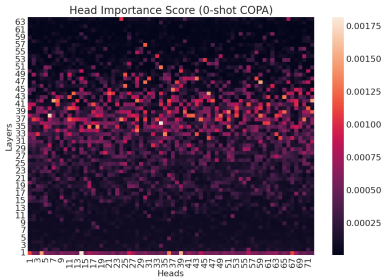
(g) RTE



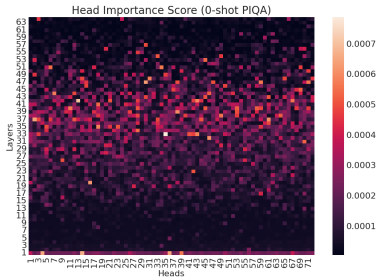
(h) MultiRC



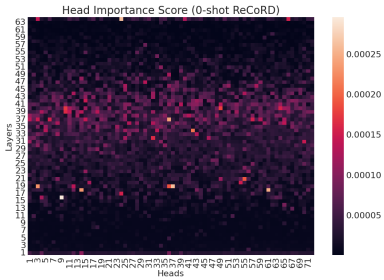
(i) OpenBookQA



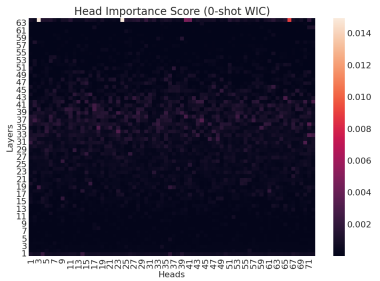
(j) COPA



(k) PIQA



(l) ReCoRD

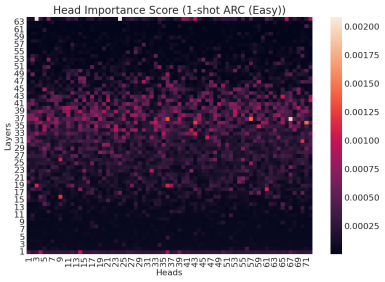


(m) WIC

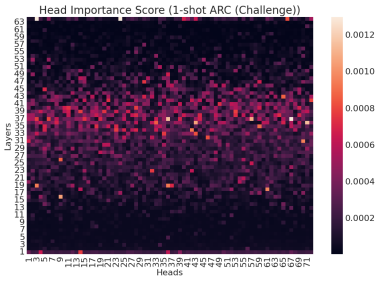
(n) WSC

Figure 12: Attention head importance score heatmaps for zero-shot in-context learning with OPT-66B for each task.

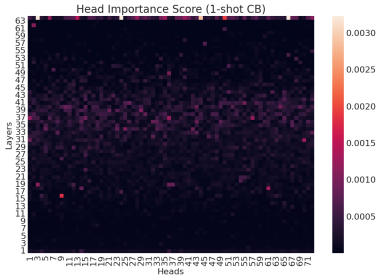
(a) HellaSwag



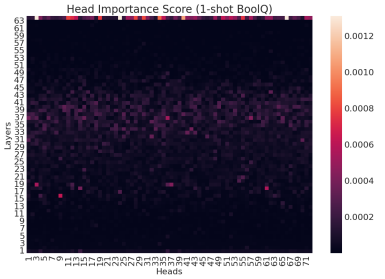
(b) ARC (Easy)



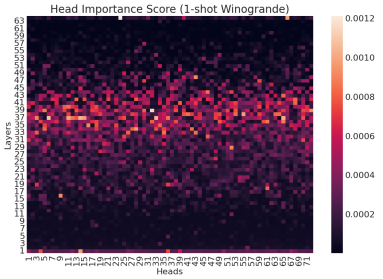
(c) ARC (Challenge)



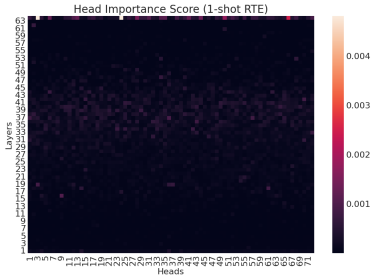
(d) CB



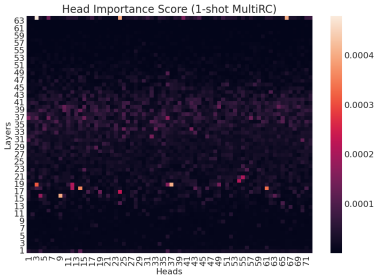
(e) BoolQ



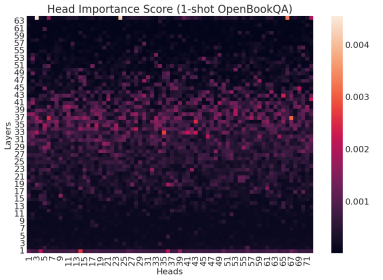
(f) Winogrande



(g) RTE



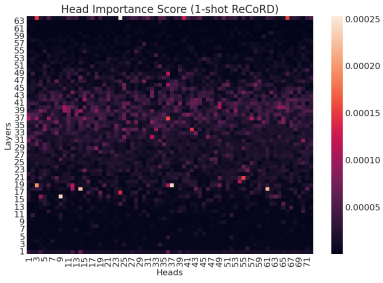
(h) MultiRC



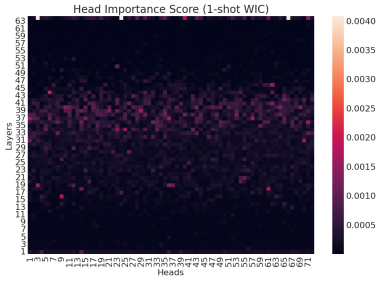
(i) OpenBookQA

(j) COPA

(k) PIQA



(l) ReCoRD

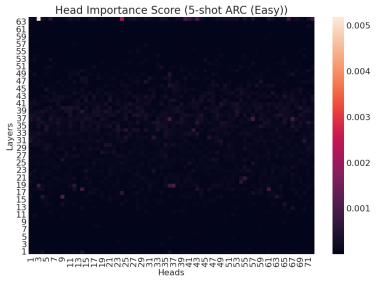


(m) WIC

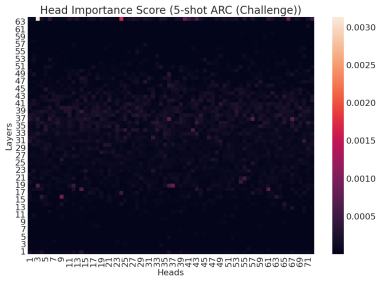
(n) WSC

Figure 13: Attention head importance score heatmaps for one-shot in-context learning with OPT-66B for each task.

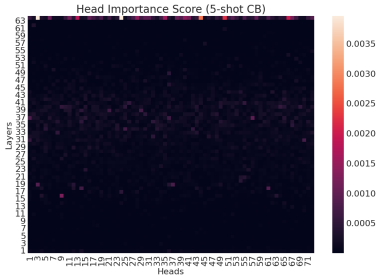
(a) HellaSwag



(b) ARC (Easy)

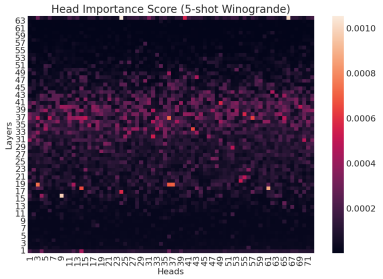


(c) ARC (Challenge)



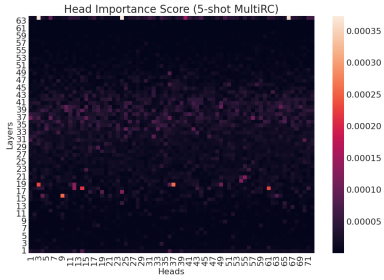
(d) CB

(e) BoolQ

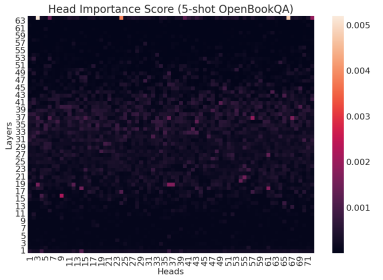


(f) Winogrande

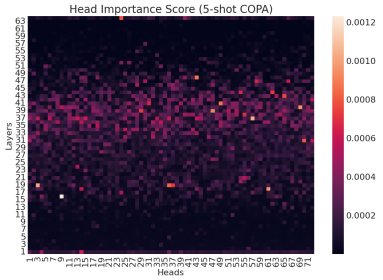
(g) RTE



(h) MultiRC

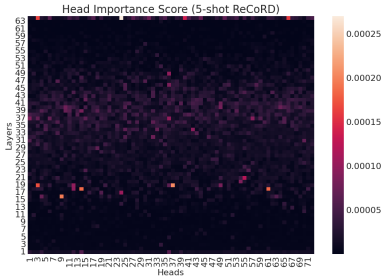


(i) OpenBookQA



(j) COPA

(k) PIQA



(l) ReCoRD

(m) WIC

(n) WSC

Figure 14: Attention head importance score heatmaps for five-shot in-context learning with OPT-66B for each task.

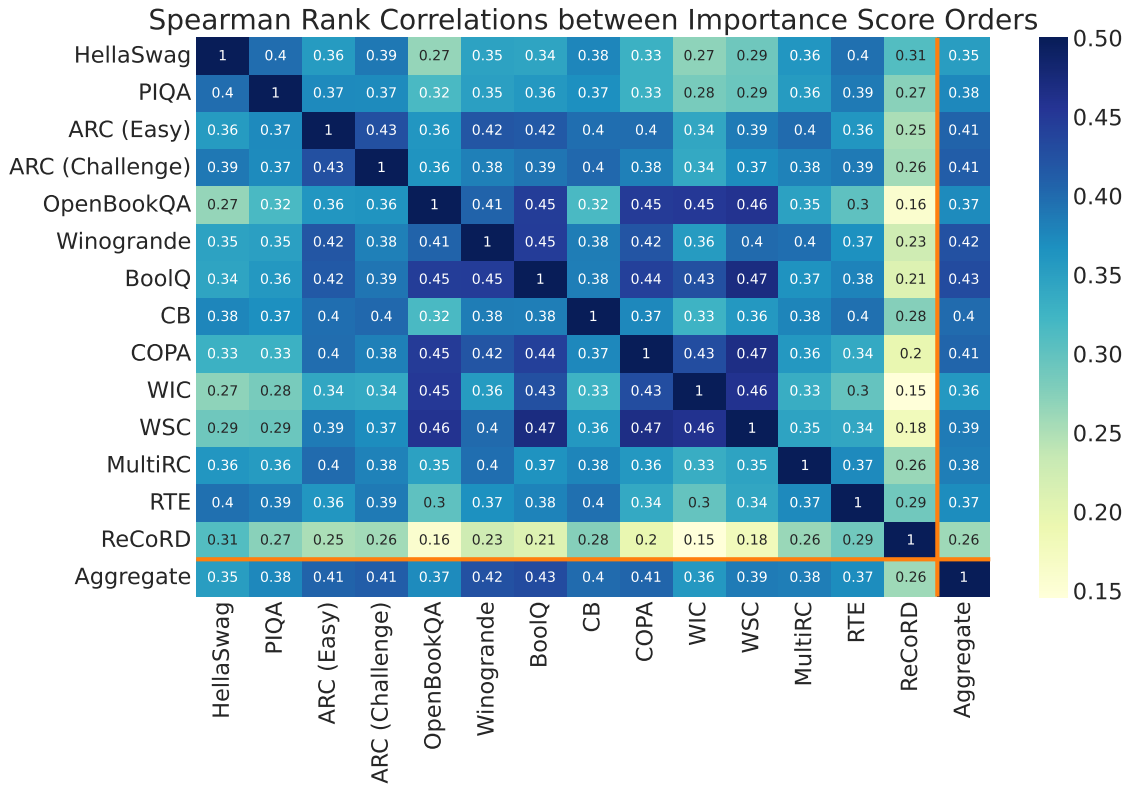
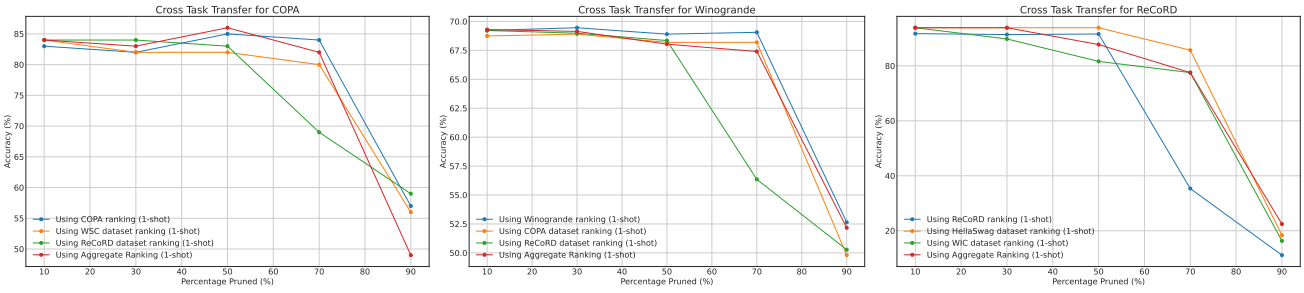
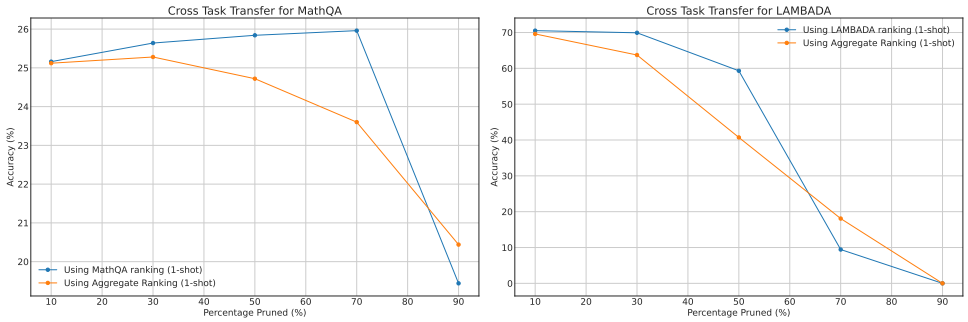


Figure 15: Spearman's rank correlation coefficients between the attention head importance rankings for various tasks in the five-shot setting. All p-values < 0.01.

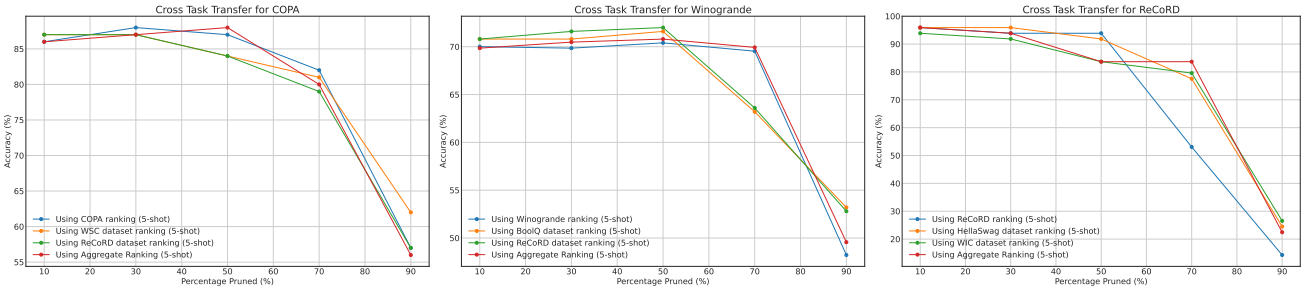


(a) COPA (b) Winogrande (c) ReCoRD

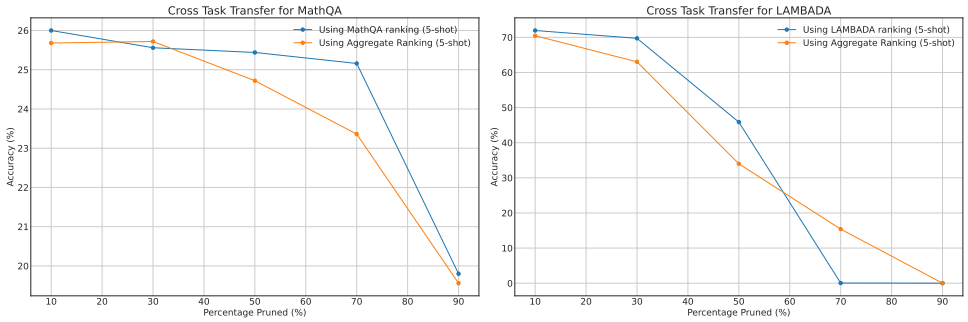


(d) MathQA (e) LAMBADA

Figure 16: Cross-task transfer of attention head importance rankings as measured by impact of pruning on accuracy in the one-shot setting.

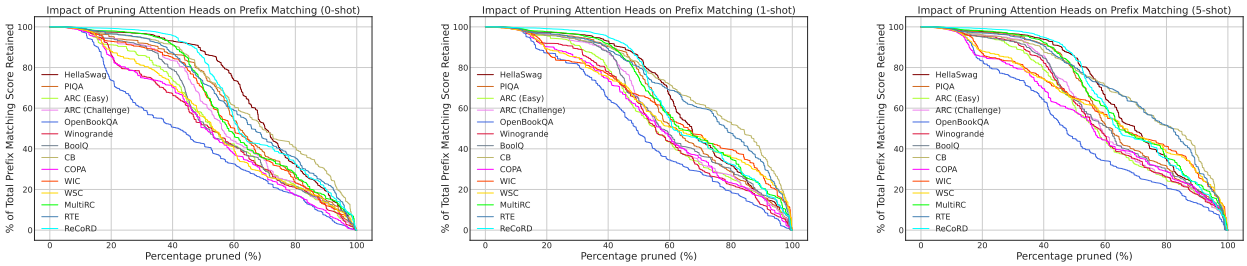


(a) COPA (b) Winogrande (c) ReCoRD



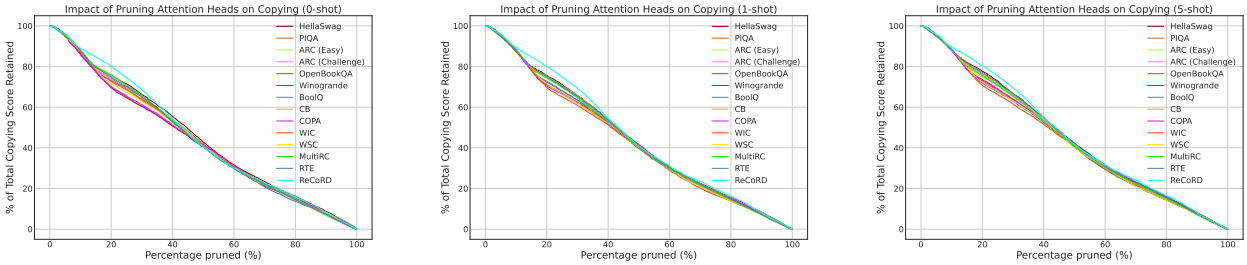
(d) MathQA (e) LAMBADA

Figure 17: Cross-task transfer of attention head importance rankings as measured by impact of pruning on accuracy in the five-shot setting.



(a) Zero-shot (b) One-shot (c) Five-shot

Figure 18: Total prefix matching capacity retained as a function of percentage of attention heads pruned, where heads are pruned based on task-specific importance score rankings in the order of least important first.



(a) Zero-shot (b) One-shot (c) Five-shot

Figure 19: Total copying capacity retained as a function of percentage of attention heads pruned, where heads are pruned based on task-specific importance score rankings in the order of least important first.