Is C4 Dataset Optimal for Pruning? An Investigation of Calibration Data for LLM Pruning

Abhinav Bandari[1,3*], **Lu Yin**[2], **Cheng-Yu Hsieh**[1], **Ajay Kumar Jaiswal**[3]
Tianlong Chen[4], **Li Shen**[5], **Ranjay Krishna**[1], **Shiwei Liu**[6]
[1] University of Washington [2] University of Surrey [3] The University of Texas at Austin
[4] The University of North Carolina at Chapel Hill [5] Sun Yat-sen University [6] University of Oxford

Abstract

Network pruning has emerged as a potential solution to make LLMs cheaper to deploy. However, existing LLM pruning approaches universally rely on the C4 dataset as the calibration data for calculating pruning scores, leaving its optimality unexplored. In this study, we evaluate the choice of calibration data on LLM pruning, across a wide range of datasets that are most commonly used in LLM training and evaluation, including four pre-training datasets as well as three categories of downstream tasks encompassing nine datasets. Each downstream dataset is prompted with In-Context Learning (ICL) and Chain-of-Thought (CoT), respectively. Besides the already intriguing observation that the choice of calibration data significantly impacts the performance of pruned LLMs, our results also uncover several subtle and often unexpected findings, summarized as follows: (1) C4 is not the optimal choice for LLM pruning, even among commonly used pre-training datasets; (2) arithmetic datasets—when used as calibration data—performs on par or even better than pre-training datasets; (3) pruning with downstream datasets does not necessarily help the corresponding downstream task, compared to pre-training data; (4) ICL is widely beneficial to all data categories, whereas CoT is only useful on certain tasks. Our findings shed light on the importance of carefully selecting calibration data for LLM pruning and pave the way for more efficient deployment of these powerful models in real-world applications. We release our code at: `https://github.com/abx393/llm-pruning-calibration-data`.

1 Introduction

In the 2020s, the landscape of AI has transitioned into a new era, propelled forward by the advancements made in large language models (LLMs) (Brown et al., 2020; Gemini Team et al., 2023; Touvron et al., 2023). The astonishing language capacities of LLMs have significantly shaped the solutions to various real-life tasks such as natural language understanding (Brown et al., 2020; Touvron et al., 2023), text generation (Kocoń et al., 2023; Anil et al., 2023), vision tasks (Radford et al., 2021; Zhou et al., 2022a), coding (Chen et al., 2022), and math (Romera-Paredes et al., 2024). However, the enormous size of these powerful LLMs poses a significant challenge for deployment in many real-world applications. For instance, deploying a 7B LLM requires around 10GB of main memory (DRAM) even after adopting INT8 quantization, which unfortunately exceeds the memory capacity of most commodity edge devices.

Network pruning, as one of the most well-established approaches in model compression, demonstrated the possibility of removing around 50% of the parameters (Frantar and Alistarh, 2023a; Sun et al., 2023; Zhang et al., 2023), or even more (Yin et al., 2023b; Agarwalla et al., 2024) with minimal performance degradation. Interestingly, while consistently producing robust performance in small-scale deep neural networks (Han et al., 2015; Frankle and Carbin, 2019; Mocanu et al., 2018; Gale et al., 2019), magnitude pruning (Han et al., 2015) seems to lose importance in the context of LLM pruning. All state-of-the-art LLM pruning approaches unanimously choose to use a small set of data (known as *calibration data*) from the C4 training dataset (Raffel et al., 2020) to calculate their pruning scores (Frantar and Alistarh, 2023a; Sun et al., 2023; Yin et al., 2023b).

Using C4 as the calibration data for pruning makes sense if the models are pre-trained on it to preserve better the desired distribution learned during pre-training. However, not all large language models are pre-trained with the C4 dataset, raising the question of whether the C4 is the optimal

*Work done while the author was at University of Washington. Correspondence to: Abhinav Bandari <abhinavbandari@utexas.edu>

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 18089–18099
November 12-16, 2024

choice for the calibration data for LLM pruning. In addition, it is well-known that LLMs are very sensitive to how the input is structured and provided to them (Zhou et al., 2022b; Shi et al., 2023). As a result, it is unclear how the input format of calibration data would affect LLM pruning.

To answer these questions, in this work, we conduct a comprehensive study to investigate the effect of calibration data on LLM pruning across a broad range of evaluation tasks, along two dimensions of interest: varying types of datasets and different data input formats. Specifically, we investigate the following possible alternatives of calibration data for LLM pruning, as illustrated in Figure 1:

- **Pre-training Data**: Apart from the C4 dataset, several other datasets are widely used for pre-training LLMs. We examine three of the most representative datasets: Pile (Gao et al., 2020), OSCAR (Suárez et al., 2020), and RedPajama (Together Computer, 2023).

- **Downstream Data**: While pruning with pre-training datasets is intuitively preferred to preserve pre-training knowledge, it is essential to empirically verify this assumption and identify whether pruning with any downstream datasets may yield superior outcomes for LLM pruning. To investigate this, we consider three categories of downstream tasks, encompassing a total of nine datasets (see Section 3.1 for details). An intriguing research question arises: *will pruning with downstream data produce a better sparse model for the corresponding downstream task than pruning with pre-training data?*

- **Prompted Downstream Data**: Acknowledging the significant impact of prompts on LLM performance, we explore two variants of prompting strategies to construct different formats of calibration data: In-Context Learning (ICL) (Brown et al., 2020) and In-Context Learning w/ Chain-of-Thought (ICL w/ CoT) (Wei et al., 2022).

- **Nonsense Data**: In addition, we explore two variants of nonsensical calibration data—ellipses and random alphanumeric strings—to investigate the necessity of semantically meaningful calibration data for effective LLM pruning.

To investigate the impact of these datasets, we prune LLMs using various calibration datasets and evaluate the resulting sparse models across nine downstream tasks. Our key and encouraging finding is that, while C4 consistently produces robust sparse models, it is not the best calibration dataset for pruning. In addition, our study unveils several more subtle and unexpected findings, which can be summarized as follows:

- C4, although consistent in producing robust sparse models, is not the optimal choice for LLM pruning, and it is also not the best among various pre-training datasets. Pile consistently outperforms C4 with higher average accuracy.

- Certain types of downstream data lead to better sparse LLMs than others. Arithmetic downstream datasets in general perform on par or even better than pre-training datasets in this context of LLM pruning.

- Pruning with downstream data does not necessarily lead to the best performance on that downstream task than pruning with a pre-training dataset like Pile.

- ICL calibration data broadly benefits all data categories, while ICL w/ CoT calibration data is only advantageous for arithmetic reasoning datasets.

2 Related Work

2.1 Large Language Model Pruning

Network pruning is a widely utilized technique to reduce model size with negligible performance loss (Mozer and Smolensky, 1989; Han et al., 2015; Molchanov et al., 2017). While numerous pruning approaches have been proposed, the success of pruning is inextricably linked to sufficient retraining (Liu et al., 2022; Wang et al., 2023). However, training large language models is prohibitively expensive and not feasible for most practitioners. Fortunately, recent research efforts have proposed effective methods that enable accurate pruning of LLMs without the need for extensive fine-tuning. SparseGPT (Frantar and Alistarh, 2023a) employs second-order pruning followed by column-wise weight updates, allowing the removal of 50% of weights while maintaining the original perplexity. Wanda (Sun et al., 2023), motivated by the goal of preserving crucial outliers in LLMs, proposes pruning weights based on the multiplication of weight

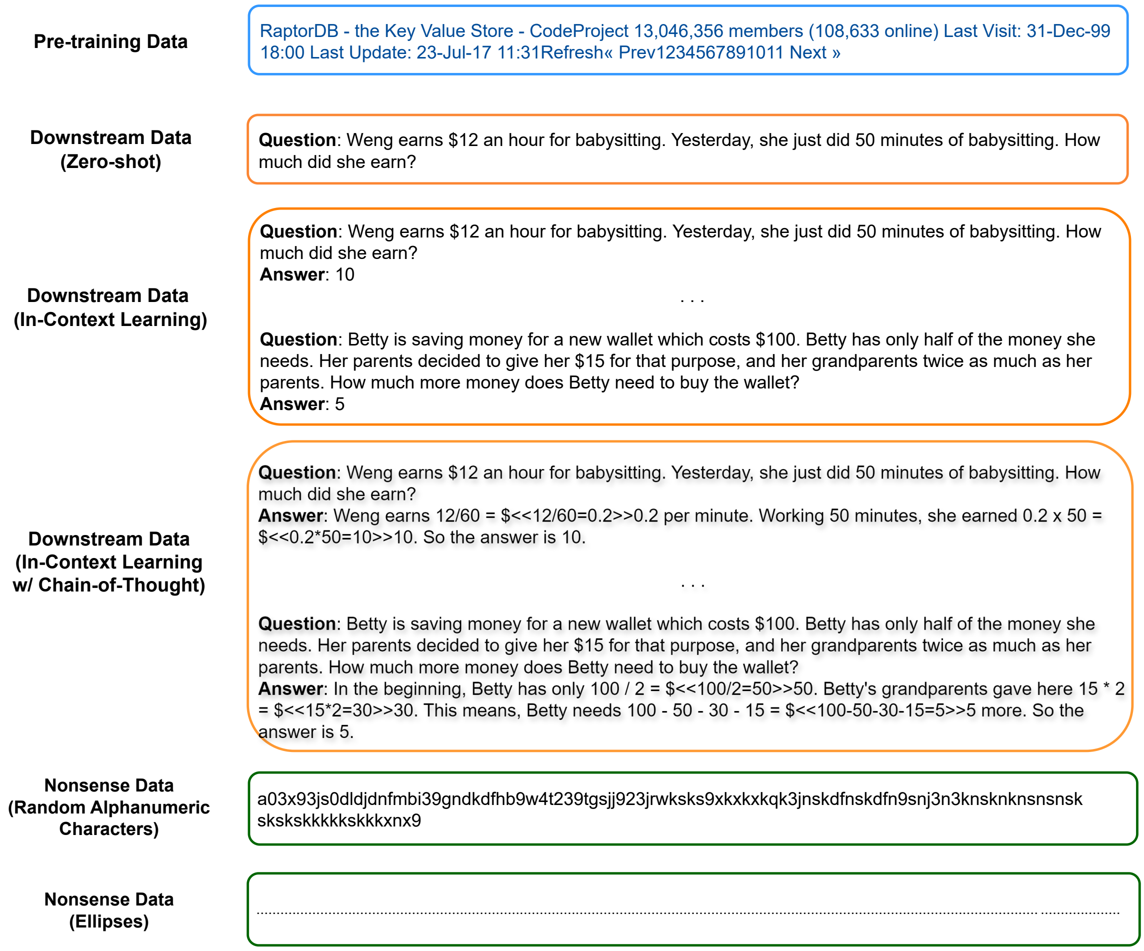

Figure 1: Examples of various calibration data formats examined in this paper.

magnitude with their input activation, demonstrating strong performance. OWL (Yin et al., 2023b) introduces a novel non-uniform layerwise sparsity approach for LLM pruning, showing promising results at high levels of sparsity. In addition to exploring accurate pruning methods, other studies focus on efficiently fine-tuning sparse LLMs to further enhance their performance (Zhang et al., 2023; Zimmer et al., 2023). In contrast to these previous works, our paper investigates the efficacy of input data for LLM pruning. This novel perspective is crucial for understanding and improving LLM pruning methodologies, as LLMs are sensitive to their input (Zhao et al., 2021).

2.2 Prompting for Sparse LLMs

Prompting involves providing instructions to a pre-trained language model, either as a single instruction (zero-shot) or through one or more examples (one/few-shot) that demonstrate the task. Brown et al. (2020) demonstrated that prompt design is highly effective for guiding a non-modifiable GPT-3 model in zero, one, and few-shot settings. Initially, efforts in prompt-tuning focused on the discrete selection of prompt template tokens, as explored by Jiang et al. (2020). Later studies, such as those by Lester et al. (2021), shifted towards using continuous prompts that were refined through backpropagation.

Xu et al. (2023) first discovered that the generation quality of a compressed LLM could be significantly improved by adding carefully designed hard prompts and proposed a soft prompt learning method to improve the compressed LLM. Hoang et al. (2023) argued that the performance drop caused by pruning is because the pre-trained knowledge is displaced rather than being forgotten. Williams and Aletras (2023) examined the impact of multiple pre-training data sources on pruning. However, their study was confined to pre-training data sources. Our research extends this investigation by not only analyzing four commonly used pre-training datasets but also exploring various downstream datasets with In-Context Learning and Chain of Thought prompts, leading to more intriguing findings and a deeper understanding of the effects of different data sources on pruning.

Table 1: Pruning metrics of Wanda and SparseGPT.

Method	Weight Update	Pruning Metric $\mathbf{S}_{ij}$
SparseGPT	✓	$\left[\|\mathbf{W}\|^2/\mathrm{diag}[(\mathbf{X}\mathbf{X}^T+\lambda\mathbf{I})^{-1}]\right]_{ij}$
Wanda	✗	$\|\mathbf{W}_{ij}\| \cdot \\|\mathbf{X}_j\\|_2$

3 Methodology

In this section, we describe in detail how we assess the effects of various calibration datasets and data formats on LLM pruning.

3.1 Pruning Methods

We choose the two most widely-used pruning methods, i.e., Wanda (Sun et al., 2023) and SparseGPT (Frantar and Alistarh, 2023b) as our pruning methods. Both pruning methods necessitate a small subset of calibration data to calculate pruning scores, which are shown in Table 1. In this context, $\mathbf{X}$ symbolizes layer activations and $\mathbf{W}$ represents weights. The expression $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ in the denominator forms the Hessian $\mathbf{H}$, essential for the layer-wise reconstruction issue, with λ serving as a dampening factor to prevent computational collapse during inversion. Wanda augments the standard weight magnitude pruning metric by integrating input activations, whereas SparseGPT incorporates an additional weight update step within its column-wise pruning process. The weights with the lowest scores will be pruned, resulting in a sparse LLM.

3.2 Model, Dataset, and Evaluation

Model. We use the common models used in previous work (Sun et al., 2023; Yin et al., 2023a), i.e., Llama 2-Chat 7B (Touvron et al., 2023) and Llama 7B (Touvron et al., 2023) as the base models for pruning.

Dataset. The source of our calibration data is divided into two categories: pre-training datasets and downstream datasets. For pre-training data, we selected four widely-used datasets: C4 (Raffel et al., 2020), Pile (Gao et al., 2020), OSCAR (Suárez et al., 2020), and RedPajama (Together Computer, 2023). To ensure the diversity of the downstream calibration data, we focused on three major tasks: arithmetic reasoning, natural language inference, and commonsense reasoning, selecting three datasets for each category.

For arithmetic reasoning, we chose the following three datasets:

- **GSM8K** (Cobbe et al., 2021) is a dataset of grade school math word problems, where each problem takes between 2 and 8 steps to solve.
- **SVAMP** (Patel et al., 2021) is another dataset of grade school math word problems, where each problem requires no more than 2 arithmetic operations to solve.
- **MAWPS** (Koncel-Kedziorski et al., 2016) is another dataset of grade school math word problems of varying complexity.

For natural language inference, we use the following datasets:

- **e-SNLI** (Camburu et al., 2018) is a dataset of entailment relations along with human-annotated natural language explanations of the labels.
- **ANLI** (Nie et al., 2020) is a dataset of entailment relations that was iteratively and adversarially generated with a human-and-model-in-the-loop procedure. **ANLI R1** represents the data produced in the first round of this.
- **ANLI R3** (Nie et al., 2020) represents the data produced in the third round of the aforementioned iterative procedure. The adversarial model is trained on data produced in previous rounds, so crowdworkers are incentivized to create distinct entailment relations to challenge the model, so ANLI R3 is distinct from ANLI R1.

For commonsense reasoning, we use the following:

- **CommonsenseQA (CSQA)** (Talmor et al., 2019) is a commonsense question answering dataset with multiple choice questions that require some prior knowledge not provided in the question.
- **RACE** (Lai et al., 2017) is a commonsense question answering dataset where each question is related to a provided text passage. It evaluates understanding and reasoning abilities.

Evaluation	Dense	Wanda *w.* Calibration Data				SparseGPT *w.* Calibration Data			
		C4	RedPajama	Oscar	Pile	C4	RedPajama	Oscar	Pile
GSM8K	0.0576	**0.0457** ± 0.0008	0.0412 ± 0.0062	0.0450 ± 0.0088	0.0404 ± 0.0048	**0.0440 ± 0.0052**	0.0430 ± 0.0048	0.0412 ± 0.0046	0.0384 ± 0.0038
SVAMP	0.3867	0.2756 ± 0.0102	0.2733 ± 0.0133	**0.2922** ± 0.0102	0.2878 ± 0.0038	0.3011 ± 0.0193	0.3033 ± 0.0370	0.3089 ± 0.0278	**0.3445 ± 0.0139**
MAWPS	0.4462	0.3160 ± 0.0293	0.3154 ± 0.0308	0.3436 ± 0.0097	**0.3635** ± 0.0115	0.3295 ± 0.0235	0.3487 ± 0.0289	0.3500 ± 0.0416	**0.3820 ± 0.0262**
e-SNLI	0.6050	0.4934 ± 0.0096	0.4940 ± 0.0377	0.4812 ± 0.0205	**0.5376** ± 0.0023	0.5447 ± 0.0326	**0.5641 ± 0.0295**	0.5485 ± 0.0487	0.5498 ± 0.0289
ANLI R1	0.3900	0.3250 ± 0.0156	0.3240 ± 0.0255	0.3203 ± 0.0042	**0.3420** ± 0.0087	0.3580 ± 0.0356	**0.3640 ± 0.0183**	0.3463 ± 0.0261	0.3380 ± 0.0020
ANLI R3	0.4192	0.3361 ± 0.0106	0.3405 ± 0.0058	0.3220 ± 0.0042	**0.3597** ± 0.0145	**0.3575 ± 0.0153**	0.3478 ± 0.0226	0.3480 ± 0.0136	0.3408 ± 0.0043
CSQA	0.6208	0.5171 ± 0.0024	0.5184 ± 0.0184	0.5225 ± 0.0043	**0.5239** ± 0.0078	0.5266 ± 0.0314	**0.5304 ± 0.0204**	0.5233 ± 0.0141	0.5258 ± 0.0331
RACE	0.6501	0.4686 ± 0.0052	0.4386 ± 0.0109	0.4632 ± 0.0224	**0.4692** ± 0.0079	0.5305 ± 0.0272	**0.5407 ± 0.0101**	0.5374 ± 0.0279	0.5376 ± 0.0215
WinoGrande	0.5122	**0.5141** ± 0.0094	**0.5141** ± 0.0067	**0.5141** ± 0.0087	0.5125 ± 0.0051	0.5183 ± 0.0143	0.5193 ± 0.0016	**0.5240 ± 0.0311**	0.5164 ± 0.0194
Average	0.4542	0.3657 ± 0.0041	0.3622 ± 0.0005	0.3671 ± 0.0055	**0.3819** ± 0.0029	0.3900 ± 0.0063	0.3957 ± 0.0090	0.3920 ± 0.0153	**0.3970 ± 0.0045**

Table 2: Accuracy of Llama 2-Chat 7B model pruned with Wanda and SparseGPT to 50% unstructured sparsity using different pre-training datasets, averaged over three random seeds. The value after ± indicates 2 standard deviations. Results for both pruning methods are shown alongside the original dense model for comparison. The best performance on each evaluation task for each pruning algorithm is bold.

Evaluation task	Dense Model	Wanda *w.* Calibration Data			
		C4	RedPajama	Oscar	Pile
GSM8K	0.0576	**0.0269**	0.0186	0.0208	0.0239
SVAMP	0.3867	**0.0200**	0.0133	**0.0200**	0.0133
MAWPS	0.4462	**0.0019**	0.0000	0.0000	0.0000
e-SNLI	0.6050	0.1313	0.2432	0.0687	**0.3249**
ANLI R1	0.3900	0.0000	0.0000	0.0000	**0.1190**
ANLI R3	0.4192	0.0000	0.0000	0.0000	**0.0925**
CSQA	0.6208	0.2138	0.2072	0.2113	**0.2170**
RACE	0.6501	0.2528	0.2197	0.2514	**0.2540**
WinoGrande	0.5102	**0.5012**	0.4925	0.4743	0.4972
Average	0.4542	0.1275	0.1327	0.1163	**0.1713**

Table 3: Accuracy of Llama 2-Chat 7B model pruned with Wanda to 70% unstructured sparsity using different pre-training datasets. Results are shown alongside the original dense model for comparison. The best performance on each evaluation task is bold.

- **WinoGrande** (Sakaguchi et al., 2019) is a commonsense question answering dataset with fill-in-the-blank statements and binary answer options.

Evaluation. To evaluate the performance of different calibration datasets, we first prune the dense LLM with certain calibration data and then evaluate the resulting sparse LLM on all the downstream tasks considered using few-shot prompting (Brown et al., 2020).

3.3 Calibration Data Formulation

The pruning calibration data have 128 sequences of length 2048 tokens each, following prior work (Frantar and Alistarh, 2023a; Sun et al., 2023; Yin et al., 2023b).

Pre-training Data. For each pre-training dataset, we create each calibration data sample of length 2048 tokens by concatenating text segments from the dataset until it exceeds 2048 tokens and then selecting a segment of length 2048 from this.

Downstream Data. To provide a comprehensive evaluation of downstream data, we use the following three variants.

- **Zero-Shot**. We create each calibration data sample by selecting a random question from the dataset without the answer. We fill up the remaining context length with padding tokens.
- **In-Context Learning.** We create each calibration data sample by concatenating multiple randomly selected question-answer pairs to fill up the context length of 2048 tokens.
- **In-Context Learning w/ Chain-of-Thought.** We create each calibration data sample by concatenating randomly selected question-answer pairs, where the answer contains CoT rationale, to fill up the context length of 2048 tokens.

4 Results

In this section, we report the results of our experiments. Our primary goal is to explore how performance fluctuates when using various calibration data across different formats. We analyze overall performance trends across these differing setups.

4.1 Pre-training Dataset as Calibration Data

We evaluate pruning performance using calibration data derived from a range of pre-training datasets including C4, RedPajama, Oscar, and Pile. The results are detailed in Table 2. Our analysis reveals that the average accuracy of Pile consistently outperforms the C4 dataset. Using Wanda with target sparsity 0.5, calibration with the Pile dataset exhibits superior performance in terms of average accuracy across nine downstream tasks, surpassing other pre-training datasets in six out of nine tasks. Similarly, for SparseGPT pruning, the Pile dataset achieves the highest average accuracy, although the differences among the four pre-training datasets are small.

Notably, when compared with the commonly used C4 dataset, our analysis reveals that RedPajama achieves comparable performance, and Pile demonstrates an improvement, outperforming C4 in Wanda pruning across a majority of downstream tasks. Specifically, using the Llama 2-Chat 7b model, Pile leads C4 in seven out of nine tasks when using Wanda. Although when using SparseGPT, Pile outperforms C4 in only four out of nine tasks, Pile still has higher average accuracy across nine tasks. In Table 3, when we target 70% sparsity, we can clearly see that RedPajama and Pile achieve significantly higher average accuracy than C4. These findings underscore that C4 is not the optimal choice of calibration data for LLM pruning. Pile consistently serves as better calibration data in LLM pruning.

4.2 Downstream Dataset as Calibration Data

While using pre-training datasets for pruning may preserve acquired knowledge, it is crucial to empirically validate this strategy and determine if alternative downstream datasets might yield superior results for pruning LLMs. To this end, we utilized downstream datasets both as calibration data for pruning and as benchmarks for evaluation.

We compare three formats of downstream data: Zero-Shot, ICL and ICL w/ CoT. We systematically assessed the pruning performance across various downstream tasks using different calibration data formats: single GSM8K question (Zero-Shot), concatenated GSM8K question-answer pairs (ICL), and concatenated GSM8K question-answer pairs with Chain of Thought (ICL w/ CoT). Our findings, detailed in Table 4, reveal that ICL consistently enhances performance across all data categories compared to the baseline zero-shot approach, achieving an average accuracy improvement of 0.1754. We also observed that GSM8K (ICL w/ CoT) calibration data outperforms GSM8K (ICL) data in Arithmetic Reasoning tasks. An explanation for this could be that the step-by-step reasoning in CoT calibration data helps guide the pruning to better preserve the model weights for arithmetic reasoning. However, GSM8K (ICL) surpasses GSM8K (ICL w/ CoT) in average performance across a broader set of downstream tasks as GSM8K (ICL) outperforms GSM8K (ICL w/ CoT) for tasks outside of arithmetic reasoning. This may be because the step-by-step reasoning in CoT introduces biases that are detrimental when the sparse model is used outside of the domain of the calibration data.

We also compare the pruning performance of e-SNLI (Zero-Shot), e-SNLI (ICL) and e-SNLI (ICL w/ CoT) in Table 4. We find that ICL again enhances performance compared to the baseline zero-shot format, with an average accuracy improvement of 0.0826. We also find that, compared to the ICL format, including CoT in the calibration data only improves performance on ANLI R3 among the three NLI evaluation tasks. For the other categories of evaluation tasks, we find that e-SNLI (ICL) and e-SNLI (ICL w/ CoT) have similar pruning performance, and the former is better for some tasks and the latter is better for others.

4.3 Winning Dataset?

We evaluated the performance of ICL tasks against the top-performing pre-training dataset, Pile, with both the Llama 2-Chat 7B and LLaMA 7B models and have presented our findings in Table 5. Specifically, using the Llama 2-Chat 7B model, in the Arithmetic Reasoning category, Pile led in two out of three tasks. For NLI and Commonsense Reasoning tasks, the best calibration datasets come from the downstream dataset and from different task categories. Upon reviewing average performance across all tasks, we observed that Arithmetic Reasoning generally matched the performance of the best pre-training dataset, Pile. Notably, SVAMP emerged as the most effective dataset overall, outperforming Pile with an average accuracy margin of 0.52% with the Llama 2-Chat 7B model and with an average accuracy margin of 2.21% with the Llama 7B model. Consequently, SVAMP has been designated as the winning dataset.

Additionally, an intriguing observation from our study was that the optimal calibration data for each downstream task did not necessarily coincide with the data from the corresponding task itself. This suggests that calibration data efficacy may not be task-specific and invites further exploration into the dynamics of calibration data across varied contexts.

5 Further Analysis

Can we do better by including more steps in CoT? In our previous construction of the calibration data, we selected question-answer pairs with no restriction on the number of steps in CoT in the answer. This inspires a follow-up question: does

Evaluation task	Dense Model	Wanda *w.* Calibration Data					
		GSM8K (Zero-shot)	GSM8K (ICL)	GSM8K (ICL w/ CoT)	e-SNLI (Zero-shot)	e-SNLI (ICL)	e-SNLI (ICL w/ CoT)
GSM8K	0.0576	0.0205	0.0425	**0.0432**	0.0303	**0.0432**	0.0379
SVAMP	0.3867	0.0233	0.2867	**0.3067**	0.1233	0.2100	**0.2133**
MAWPS	0.4462	0.0058	0.3442	**0.3519**	0.0635	**0.2635**	0.2404
e-SNLI	0.6050	0.3292	**0.5438**	0.5080	0.3428	**0.5541**	0.5517
ANLI R1	0.3900	0.2920	**0.3180**	0.3050	0.3340	**0.3350**	0.3330
ANLI R3	0.4192	0.2417	**0.3567**	0.3108	0.3350	0.3450	**0.3717**
CSQA	0.6208	0.2138	**0.5381**	0.5184	0.4087	0.5127	**0.5201**
RACE	0.6501	0.2067	**0.4793**	0.4698	0.3522	0.4653	**0.4710**
WinoGrande	0.5122	0.5114	0.5130	**0.5154**	0.5051	**0.5091**	0.5075
Average	0.4542	0.2049	**0.3803**	0.3699	0.2772	0.3598	**0.3607**

Table 4: Accuracy of Llama 2-Chat 7B model pruned with Wanda to 50% unstructured sparsity using different formats of GSM8K and e-SNLI as calibration data. For each evaluation task, the best performance among the GSM8K calibration data variants and the best performance among the e-SNLI calibration data variants is bold.

Model	Evaluation	Dense	Wanda *w.* Calibration Data									
			PD	Arithmetic Reasoning			NLI			Commonsense Reasoning		
			Pile	GSM8K	SVAMP	MAWPS	e-SNLI	ANLI R1	ANLI R3	CSQA	RACE	WinoGrande
Llama 2-Chat 7B	GSM8K	0.0576	0.0404	0.0425	0.0425	**0.0462**	0.0432	0.0417	0.0455	0.0417	0.0409	0.0432
	SVAMP	0.3867	**0.2878**	0.2867	0.2833	0.2733	0.2100	0.2633	0.2667	0.2233	0.2667	0.2600
	MAWPS	0.4462	**0.3635**	0.3442	0.3365	0.3346	0.2635	0.3038	0.3038	0.2654	0.3231	0.2731
	e-SNLI	0.6050	0.5376	0.5438	0.5711	0.5436	0.5541	0.5345	0.5441	0.5768	0.5317	**0.5955**
	ANLI R1	0.3900	0.3420	0.3180	0.3440	0.313	0.3350	0.3500	0.3490	0.3360	0.3370	**0.3520**
	ANLI R3	0.4192	0.3597	0.3567	**0.3875**	0.3700	0.3450	0.3700	0.3575	0.3633	0.3642	0.3792
	CSQA	0.6208	0.5239	0.5381	0.5233	0.5045	0.5127	0.5045	0.5364	**0.5479**	0.5373	0.5070
	RACE	0.6501	0.4692	**0.4793**	**0.4793**	0.4726	0.4653	0.4341	0.4645	0.4706	0.4625	0.4422
	WinoGrande	0.5122	0.5125	0.5130	0.5162	0.5114	0.5091	**0.5257**	0.5241	0.5107	0.5162	0.5209
	Average	0.4542	0.3819	0.3803	**0.3871**	0.3744	0.3598	0.3697	0.3768	0.3706	0.3755	0.3748
LLaMA 7B	GSM8K	0.0447	0.0409	**0.0462**	0.0440	0.0417	0.0394	0.0394	0.0417	**0.0462**	0.0387	0.0447
	SVAMP	0.3267	**0.2733**	0.1533	0.2533	0.1900	0.1833	0.0867	0.1067	0.0733	0.0967	0.0800
	MAWPS	0.3596	0.3173	0.3327	**0.3577**	0.3096	0.1615	0.2942	0.2846	0.1615	0.2808	0.2385
	e-SNLI	0.5556	0.3284	0.3433	**0.3767**	0.3678	0.3653	0.3430	0.3411	0.3304	0.3306	0.3291
	ANLI R1	0.3800	0.3210	**0.4000**	**0.4000**	0.3700	0.3340	0.2600	0.3100	0.3800	0.2600	0.3900
	ANLI R3	0.3167	0.3625	0.3833	0.3833	0.3750	0.3317	0.3583	**0.4167**	0.3417	0.3667	0.3917
	CSQA	0.3948	0.2613	**0.2907**	0.2793	0.2523	0.1974	0.2604	0.2735	0.2629	0.2752	0.2883
	RACE	0.3134	0.2758	0.2972	0.2748	0.2525	0.2839	0.2657	**0.3103**	0.2698	0.2880	0.2748
	WinoGrande	0.5130	0.4964	0.5154	0.5067	**0.5264**	0.5138	0.5162	0.5036	0.5043	0.5178	0.5225
	Average	0.3561	0.2974	0.3069	**0.3195**	0.2984	0.2678	0.2693	0.2876	0.2633	0.2727	0.2844

Table 5: Accuracy of Llama 2-Chat 7B model and LLaMA 7B model pruned with Wanda to 50% sparsity using various downstream datasets with ICL format. PD denotes pre-training data. The best performance on each evaluation task among sparse models is bold.

the number of steps of CoT rationale in the calibration data affect the sparse LLM's performance? We investigated this by constructing calibration data by concatenating multiple question-answer pairs, where each answer rationale contains exactly x steps. Since 1-step or 2-step CoT data was scarce, we performed this for $x = \{3, 4, 5\}$ as seen in Table 6. We find no clear relationship between the number of steps of CoT in calibration data and the performance of the sparse LLM. However, we note that it is possible to produce a better sparse LLM for a given task by restricting the calibration data to a specific number of steps, which may vary based on the evaluation task.

Does more Q-A pairs in ICL calibration data lead to a better sparse model? To investigate this, we evaluated the pruning performance when calibration data contains 5, 10, 15, 20, and 25 Q-A pairs, filling the rest of the context window with padding tokens. Our default ICL calibration data fills the context window with Q-A pairs until it reaches length 2048 tokens, which in practice can be anywhere from 25 to 30 Q-A pairs. We compare the pruning performance of all of these calibration data formats in Table 7. The results confirm our conjecture that an increase in in-context examples in the pruning calibration data generally correlates with enhanced performance of the sparse model.

How does input length affect the pruning performance? In our main experiments, the calibration

Evaluation task	Dense Model	Sparse Model					
		Pile	GSM8K (ICL w/ CoT)				
			Default (any # of steps of CoT)	3-Step CoT	4-Step CoT	5-Step CoT	
GSM8K	0.0576	0.0404	**0.0432**	0.0402	0.0409	0.0387	
SVAMP	0.3867	0.2878	0.3067	0.3100	**0.3133**	0.3033	
MAWPS	0.4462	0.3635	0.3519	0.3558	0.3673	**0.3808**	

Table 6: Accuracy of Llama 2-Chat 7B model pruned with Wanda to 50% sparsity using different numbers of steps of CoT in the calibration data. For instance, GSM8K (ICL w/ x-step CoT) indicates the calibration data consists of concatenations of several question-answer pairs where each answer has exactly x steps of reasoning. The default configuration of GSM8K (ICL w/ CoT) has no restriction on the number of steps of CoT.

Evaluation task	Dense Model	Calibration Data	# In-Context Q-A Pairs	Sparse Model
GSM8K	0.0576	C4	-	0.0455
GSM8K	0.0576	Pile	-	0.0404
GSM8K	0.0576	GSM8K	5	0.0288
GSM8K	0.0576	GSM8K	10	0.0440
GSM8K	0.0576	GSM8K	15	0.0455
GSM8K	0.0576	GSM8K	20	0.0417
GSM8K	0.0576	GSM8K	25	**0.0470**
GSM8K	0.0576	GSM8K	Fill Q-A pairs to sequence length (2048 tokens)	0.0425

Table 7: Accuracy of Llama 2-Chat 7B model pruned with Wanda to 50% unstructured sparsity using GSM8K with different calibration data lengths and pre-training data.

Evaluation task	Dense Model	Sparse Model		
		Pile	ellipses	random alphanumeric
GSM8K	0.0576	**0.0404**	0.0273	0.0402
SVAMP	0.3867	**0.2878**	0.0576	0.1433
MAWPS	0.4462	**0.3635**	0.0096	0.1462
e-SNLI	0.6050	**0.5376**	0.3295	0.3679
ANLI R1	0.3900	**0.3420**	0.3100	0.3250
ANLI R3	0.4192	**0.3597**	0.3300	0.3275
CSQA	0.6208	**0.5239**	0.1925	0.3170
RACE	0.6501	**0.4692**	0.2631	0.3293
WinoGrande	0.5122	**0.5125**	0.4972	0.5043
Average	0.4542	**0.3819**	0.2241	0.2779

Table 8: Accuracy of Llama 2-Chat 7B model pruned with Wanda to 50% unstructured sparsity using Pile, ellipses, and random alphanumeric characters.

Evaluation task	Dense Model	Pruning Input Length	Sparse Model
WikiText		128	29.22
WikiText		256	15.72
WikiText	6.94	512	11.82
WikiText		1024	9.27
WikiText		2048	**8.48**

Table 9: Perplexity of Llama 2-Chat 7B model on WikiText pruned with Wanda to 50% unstructured sparsity using different input lengths of C4.

data for pruning consisted of 128 sequences, each 2048 tokens in length. It is crucial to investigate whether this specific token length is necessary for effective pruning. To address this question, we used the C4 dataset for calibration and systematically varied the calibration data lengths between 256, 512, 1024, and 2048 tokens. We then evaluated the perplexity of Llama 2-Chat 7B pruned to 50% unstructured sparsity using Wanda. As detailed in Table 9, our findings confirm that increased input lengths correlate positively with improved model performance, aligning with our initial expectations.

Does input data for pruning have to be sensible? In our previous setup, calibration data for pruning is sourced from either pre-training datasets or task-specific downstream datasets. It is intriguing to compare this with the pruning performance of nonsense data calibration data, such as ellipses and random alphanumeric strings, in this context. Consequently, we substituted conventional calibration data with these unconventional types for pruning the Llama 2-Chat 7B model to 50% unstructured sparsity using the Wanda pruning method. The performance outcomes are shown in Table 8. The results clearly show that the Pile dataset, which contains human-readable data, consistently outperforms both ellipses and random alphanumeric strings in nearly all cases except one scenario within the GSM8K task. Moreover, random alphanumeric data generally exhibited better performance compared to ellipses. Therefore we affirm the importance of utilizing sensible calibration data for the effective pruning of LLMs.

6 Conclusion

This study critically examines the widely held belief that the C4 dataset is the optimal calibration choice for pruning LLMs. Through an extensive evaluation encompassing a variety of calibration data types—both pre-training and downstream datasets, our findings reveal that C4 does not hold universal superiority. Specifically, our analysis demonstrates that the pretraining dataset Pile consistently outperforms C4, while alternative downstream datasets, particularly those involving arithmetic reasoning tasks, yield comparable pruning outcomes.

Furthermore, our investigation into various downstream task formats has uncovered that In-Context Learning (ICL) offers significant benefits across all data categories. In-Context Learning w/ Chain-of-Thought (ICL w/ CoT) calibration is particularly effective in enhancing performance in arithmetic reasoning tasks. Our study advocates for a more nuanced selection and curation of calibration data, which could lead to more efficient and effective LLM pruning strategies, ultimately facilitating the deployment of more robust models in practical settings.

7 Limitations

Our study has several limitations. First, all experiments were conducted using the Llama 2-Chat 7B and LLaMA 7B models; we aim to expand our investigations to other LLM architectures and larger models. Second, our analysis was limited to the Wanda and SparseGPT pruning algorithms. Future work will explore a broader range of pruning methods. Third, we plan to evaluate the effects of combining multiple datasets on pruning performance. We believe that our insights regarding calibration data will inspire further research within the community.

Another limitation of this work we aim to address in the future is that we have not rigorously investigated why Pile is better calibration data than C4 for LLM pruning. We conjecture the benefits come from that Pile is a more diverse dataset with higher quality of examples, which is designed such that models trained on it have improved downstream generalization capabilities, compared to the more noisy Common Crawl datasets like C4, as also pointed out in recent work in the context of LLM pretraining (Li et al., 2024). As such, we believe Pile could provide more robust calibration data to guide the pruning of LLMs to optimize the performance of the sparse model on a variety of downstream tasks. We leave the investigation on the correlation between a dataset's effectiveness for LLM pretraining and model pruning as a future direction to explore.

Acknowledgement

S. Liu is funded by the Royal Society with the Newton International Fellowship.

References

Abhinav Agarwalla, Abhay Gupta, Alexandre Marques, Shubhra Pandit, Michael Goin, Eldar Kurtic, Kevin Leong, Tuan Nguyen, Mahmoud Salem, Dan Alistarh, et al. 2024. Enabling high-sparsity foundational llama models with efficient pretraining and deployment. *arXiv preprint arXiv:2405.03594*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

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.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Fuxiang Chen, Fatemeh H Fard, David Lo, and Timofey Bryksin. 2022. On the transferability of pre-trained language models for low-resource programming languages. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*, pages 401–412.

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

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*.

Elias Frantar and Dan Alistarh. 2023a. Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning (ICML)*.

Elias Frantar and Dan Alistarh. 2023b. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR.

Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1135–1143.

Duc NM Hoang, Minsik Cho, Thomas Merth, Mohammad Rastegari, and Zhangyang Wang. 2023. (dynamic) prompting might be all you need to repair compressed llms. *arXiv preprint arXiv:2310.00867*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. Chatgpt: Jack of all trades, master of none. *Information Fusion*, 99:101861.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 1152–1157.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Li Shen, Decebal Constantin Mocanu, Zhangyang Wang, and Mykola Pechenizkiy. 2022. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. *arXiv preprint arXiv:2202.02643*.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. 2018. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9:1–12.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations (ICLR)*.

Michael C Mozer and Paul Smolensky. 1989. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 107–115.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? *CoRR*, abs/2103.07191.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. *arXiv preprint arXiv:2006.06202*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

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. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv e-prints*, arXiv:2307.09288.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Huan Wang, Can Qin, Yue Bai, and Yun Fu. 2023. Why is the state of neural network pruning so confusing? on the fairness, comparison setup, and trainability in network pruning. *arXiv preprint arXiv:2301.05219*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPs)*, 35:24824–24837.

Miles Williams and Nikolaos Aletras. 2023. How does calibration data affect the post-training pruning and quantization of large language models? *arXiv preprint arXiv:2311.09755*.

Zhaozhuo Xu, Zirui Liu, Beidi Chen, Yuxin Tang, Jue Wang, Kaixiong Zhou, Xia Hu, and Anshumali Shrivastava. 2023. Compress, then prompt: Improving accuracy-efficiency trade-off of llm inference with transferable prompt. *arXiv preprint arXiv:2305.11186*.

Lu Yin, Gen Li, Meng Fang, Li Shen, Tianjin Huang, Zhangyang Wang, Vlado Menkovski, Xiaolong Ma, Mykola Pechenizkiy, and Shiwei Liu. 2023a. Dynamic sparsity is channel-level sparsity learner. *arXiv preprint arXiv:2305.19454*.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Mykola Pechenizkiy, Yi Liang, Zhangyang Wang, and Shiwei Liu. 2023b. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. *arXiv preprint arXiv:2310.05175*.

Yuxin Zhang, Lirui Zhao, Mingbao Lin, Yunyun Sun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei Liu, and Rongrong Ji. 2023. Dynamic sparse no training: Training-free fine-tuning for sparse llms. *arXiv preprint arXiv:2310.08915*.

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.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022b. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

Max Zimmer, Megi Andoni, Christoph Spiegel, and Sebastian Pokutta. 2023. Perp: Rethinking the prune-retrain paradigm in the era of llms. *arXiv preprint arXiv:2312.15230*.