Junk DNA Hypothesis: Pruning Small Pre-Trained Weights *Irreversibly* and *Monotonically* Impairs "Difficult" Downstream Tasks in LLMs

Lu Yin [* 1 2] Ajay Jaiswal [* 3] Shiwei Liu [1 4] Souvik Kundu [5] Zhangyang Wang [3]

Abstract

We present *Junk DNA Hypothesis* by adopting a novel *task-centric* angle for the pre-trained weights of large language models (LLMs). It has been believed that weights in LLMs contain significant redundancy, leading to the conception that a considerable chunk of the parameters can be removed by *pruning* without compromising performance. Contrary to this belief, this paper presents a *counter-argument*: small-magnitude weights of pre-trained model weights encode vital knowledge essential for tackling difficult downstream tasks - manifested as the monotonic relationship between the performance drop of downstream tasks across the difficulty spectrum, as we prune more pre-trained weights by magnitude. Moreover, we reveal that these seemingly inconsequential weights can result in irreparable loss of knowledge and performance degradation in difficult tasks, even when downstream continual training is allowed. Interestingly, our evaluations show that the other popular compression, namely *quantization* **fail** to exhibit similar "monotonic" effect and does not as convincingly disentangle this task-difficulty information. To study formally, we introduce several quantifiable metrics to *gauge the downstream task difficulty*: ❶ within the same task category, and ❷ across different task categories. Our extensive experiments substantiate the Junk DNA Hypothesis across a diverse range of model sizes, tasks, datasets, and even pruning methods. Codes are available at https://github.com/VITA-Group/Junk_DNA_Hypothesis.git.

*Equal contribution [1]Eindhoven University of Technology [2]University of Surrey [3]University of Texas at Austin [4]University of Oxford [5]Intel Labs. Correspondence to: Zhangyang Wang <atlaswang@utexas.edu>.

1. Introduction

A prevailing belief suggests that neural networks contain **significant and non-significant components** across their parameters that record necessary or unnecessary expertise respectively to handle a task, which forms the inherent basis of numerous model compression techniques (Han et al., 2016). Moreover, as model sizes continue to expand, the volume of redundant parameters is poised to escalate (Liu et al., 2022; Kim et al., 2021). This principle extends its effectiveness even to billion-level Large Language Models (LLMs) (Jaiswal et al., 2023b; Frantar & Alistarh, 2023; Hu et al., 2021; Lin et al., 2023b). With the negligible loss in performance stemming from the absence of non-significant components, a widely held belief has taken root: these non-significant components are essentially superfluous components that make a scant contribution to the model's functionality. Yet we pause and pose a question: *Could there be crucial facets overlooked in the context of whether these non-significant components are truly inconsequential artifacts for large-scale models?*

This paper addresses the aforementioned query by employing **two** popular weight compression methods namely, *pruning*, and *quantization*, to *concretely discern "what non-significant components are in pre-trained LLM weights, what they do, and what is the best way to disentangle them"*. At this point, it is important to clarify that this paper *neither aims to be an LLM compression exposition* nor to re-justify the commonsense that large pre-trained models are compressible - but rather, to use existing popular compression techniques as *quantitative and easily controllable tool* to probe and comprehend the existence and the functional role of non-significant weight components. The crux of our research lies in adopting a novel task-centric viewpoint towards pre-trained weights. In other words, we investigate how the non-significant weights fraction and their embodied information *correlate with model's ability to successfully perform downstream tasks of different complexity levels.*

Our extensive experiment-based study disrupts conventional assumptions by providing three key observations: ❶ the assumed non-significant weight components play a pivotal role in handling challenging tasks, ❷ element-wise pruning simply by pre-trained weight magnitudes, despite being a

naive criterion, is a reliable indicator to *capture the demarcation of task difficulty* encoded within significant and non-significant components - experimentally shown as "monotonically impairing" downstream task performance from difficult to simple, as more are pruned by magnitude. ❸ Meanwhile, quantization fail to display such monotonic effect and do not as convincingly disentangle this task-difficulty information. Drawing a parallel with biological insights[1], we nickname our discoveries as the **Junk DNA Hypothesis**:

- Contrary to common beliefs, small-magnitude weights of a pre-trained model encode vital knowledge essential for tackling difficult downstream tasks - manifested as **monotonic** relationships between the amount of pruned weights and the difficulty level of tasks that start to be impaired (Section 3.1 & 3.2). Interestingly though, the same monotonicity observation *fail* to hold for quantization, another popular compression method.

- Removing these ostensibly inconsequential weights can cause **irreparable** loss of knowledge and thus performance degradation on difficult tasks. This can be observed even if downstream continual training is allowed (Section 5).

The primary challenge in formalizing and experimentally validating this conjecture lies in providing a precise and "continuously controllable" definition of "task difficulty", for which we explore an extensive range of options:

- **Within the Same Task Category**: To define task difficulty within the same task, we formulate the following **four** settings: ❶ vary the adequacy of target domain data (Liu et al., 2019) (e.g., from few-shot to full-shot); ❷ availability of option counts in multiple-choice question answering (QA) setting; ❸ influence of external information availability with varying context length for retrieval-augmented QA (Ram et al., 2023); ❹ varying k-shot in-context demonstration examples to assist multiple-choice QA (Mosbach et al., 2023).

- **Across Diverse Task Categories**: To define task difficulty across different tasks, we formulate the following **two** settings: ❶ we utilize the gap between the best human performance, and the target LLM model's per-

formance on a specific task (normalized by human performance), as an indicator of complexity "sensed" by LLM for that specific task; ❷ we dive in comparing between two QA tasks: factoid-based QA which involve generating precise facts about entities, versus multiple-choice QA setting where models have to choose from a set of provided answer options.

Our extensive experiments substantiate the Junk DNA Hypothesis across a diverse range of model sizes, tasks, and datasets. While the overarching notion that·"more challenging downstream tasks permit less room for pruning" may not come as a surprise, our study unveils **several more subtle, often unexpected findings**:

- Moving beyond the nebulous distinction between simple and complex tasks, the various conceptions of task difficulty **uniquely** defined above by us, both within and across tasks, appear to align closely with the behavior of pruning fragility. This suggests a practical desire to estimate the task-dependent achievable degree of LLM sparsity. In certain tasks, even a modest reduction in low-magnitude pre-trained weights (e.g., 10-20%) results in a significant drop in accuracy, underscoring their pivotal role in handling intricate tasks.

- Unlike pruning, we found that quantization fail to effectively capture the same "monotonically" impairing effect of gradual compression when task complexity varied from difficult to simple.

- Moreover, we confirm that for difficult tasks, the essential knowledge resides within the pre-trained weight values and uninformed pruning can do "irreversible" damage for challenging tasks, i.e., the "damage" caused by overly pruning pre-trained weights cannot be un-done by continual downstream fine-tuning.

- Junk DNA Hypothesis holds true when transitioning from unstructured to structured N:M pruning, and even extends to other weight importance-based LLMs pruning techniques (SparseGPT, Wanda). Interestingly, we found the aforementioned methods tend to benefit simple tasks more compared to challenging tasks.

2. Experimental Settings

In this section, we provide a summary of our experimental settings used to validate Junk DNA hypothesis. Table 1 provides details on dataset, model configuration, learning setting, pruning techniques, and difficulty spectrum for various task settings used to quantitatively control task difficulty with and across tasks. We first experiment with popular pruning methods (allow multiple data points to validate our argument): One-shot Magnitude, SparseGPT, and Wanda to comprehensively validate the **monotonic impairment**

---

[1]Approximately 2% of human genome encodes proteins, leaving the remaining portion seemingly superfluous (Carey, 2015). This non-coding section of the genome has earned the moniker "Junk DNA" (Ohno, 1972), positing that large genomes would inevitably harbor non-coding sequences, passively accumulated over millennia, devoid of any protein-coding capacity. Yet over the past decade, it has become evident that at least some of these seemingly extraneous DNAs play essential roles in cellular function. For example, these regions of DNA contain vital sequences that act as regulatory elements (Zheng et al., 2010). In this paper, we borrow the "Junk DNA" nickname to refer to those "unsung heroes" in LLM pre-trained weights - whose functionalities have been under-recognized in previous LLM compression literature.

Task Setting	Dataset	Model	Learning	Pruning	Difficulty Variation	Across Task
Setting 1: Data Adequacy	GLUE benchmark	RoBERTa-Base	Fine-tuning	Magnitude, N:M	6	No
Setting 2: MCQ Option Count	MMLU	Vicuna-7B	None	Magnitude, SparseGPT, Wanda, N:M	3	No
Setting 3: Context Length	TriviaQA	Vicuna-7B	None	Magnitude, SparseGPT, Wanda, N:M	5	No
Setting 4: k-shot examples	MMLU	Vicuna-7B	Few-shot	Magnitude, SparseGPT, Wanda, N:M	4	No
Setting 5: Human-Centric	PIQA, HellaSwag, OpenBookQA	Vicuna-13B	None	Magnitude, SparseGPT, Wanda, N:M	3	Yes
Setting 6: Factoid v.s. MCQ	FreebaseQA, MMLU	Vicuna-7B	None	Magnitude, SparseGPT, Wanda	2	Yes

Table 1. Our experimental task settings to quantitatively control task difficulty within tasks and across tasks.

claim on our curated difficulty spectrum across various task settings. Next, in Section 4, we will draw parallel between pruning and quantization to illustrate how the later fails to capture this monotonic trend, thus unable to facilitate demarcation across task difficulty.

Sparsity: We consider two types of sparsities: (1) *Unstructured Sparsity*: individual weights in the model are zeroed out independently, leading to irregular zero patterns (LeCun et al., 1990; Han et al., 2016); (2) *Structured N:M Sparsity*: a fine-grained sparsity pattern in which only *N* weights are non-zero for every continuous *M* weights (Nvidia, 2020; Zhou et al., 2021). We report the results of *M*=8 (*N* ranges from 7 to 1). Note that we intentionally focus on one-shot magnitude pruning, to isolate the effect of small-magnitude weights as the sole "delta" between the two models, and due to its recently observed promise to retain LLM performance (Jaiswal et al., 2023b). We additionally include two more SoTA LLM pruning methods: SparseGPT (Frantar & Alistarh, 2023), and Wanda (Sun et al., 2023) to confirm whether the validity of our hypothesis can generalize to other weight importance criteria as well.

Dataset: We explore a wide variety of datasets while defining task difficulty to convincingly establish the validity of Junk DNA Hypothesis. For our experiments, we use GLUE benchmark (Wang et al., 2018), MMLU benchmark covering 50+ subjects (Hendrycks et al., 2020), TriviaQA (Joshi et al., 2017), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), and FreebaseQA (Jiang et al., 2019). More information about the aforementioned datasets is available in Appendix D.

Normalized Performance Drop: Model performance depends on task category and difficulty level. To fairly illustrate the impact of compression across task category, task difficulty, and pruning techniques, our experiments present normalized performance drop to facilitate ease in perceiving conclusion. Let f_d^t and f_p^t be dense and pruned model performance on task t, normalized performance drop is calculated as $(f_p^t - f_d^t)/f_d^t$.

To quantitatively capture the **monotonic impairment and trend** across the task difficulty spectrum, we make use of two popular statistical measures:

1. Spearman's rank correlation: It assesses how well the ranking relationship between two variables (X & Y) can be described using a monotonic function. A Spearman

correlation of zero indicates that there is no tendency for Y to either increase or decrease when X increases.

2. Theil–Sen estimator: It is a robust estimation of linear slopes (treated as a surrogate of "how fast Y changes with X") from many (potentially noisy) sample points (X & Y). It works by choosing the median of the slopes of all possible lines that go through any pair of points. Higher slopes by Theil–Sen estimator indicate faster changes of Y w.r.t. X.

3. How Pruning Illicit Monotonic Impairment Across Difficulty Spectrum?

3.1. Case study of difficulty spectrum within same task

3.1.1. TASK DIFFICULTY SETTING 1: *Varying the Adequacy of Target Domain Data*

Rationale: The difficulty of learning a task is commonly thought to be influenced by the number of available training examples: fewer data points typically imply more challenges to learn well. To quantitatively control task difficulty within a single task, we manually manipulate the *volume of data used for fine-tuning* by randomly sampling certain ratios of data from the target domain dataset. This allows us to *disentangle task difficulty from the task type*.

Method: To examine the influence of small-magnitude weights, we conduct a comparative analysis between two models: one starting from the pre-trained model with small-magnitude weights, and the other without. The former is commonly referred to as task-specific fine-tuning on downstream tasks, denoted as **Dense Transfer** in this paper. The latter model, named **Sparse Transfer**, differs from dense transfer in the way that we first perform magnitude pruning on the pre-trained model, creating a sparse model. We then fine-tune on downstream tasks while keeping the sparse mask fixed. Figure 1 summarizes our results for this setting across unstructured and structured sparsity patterns.

3.1.2. TASK DIFFICULTY SETTING 2: *Varying the Option Count in Multiple-choice QA Setting*

Rationale: This setting proposed in our work, extends the idea that given a task of multiple-choice question answering with a fixed dataset, the difficulty of the task is proportional to the number of available options to select the correct answer. To quantitatively control task difficulty, we manually
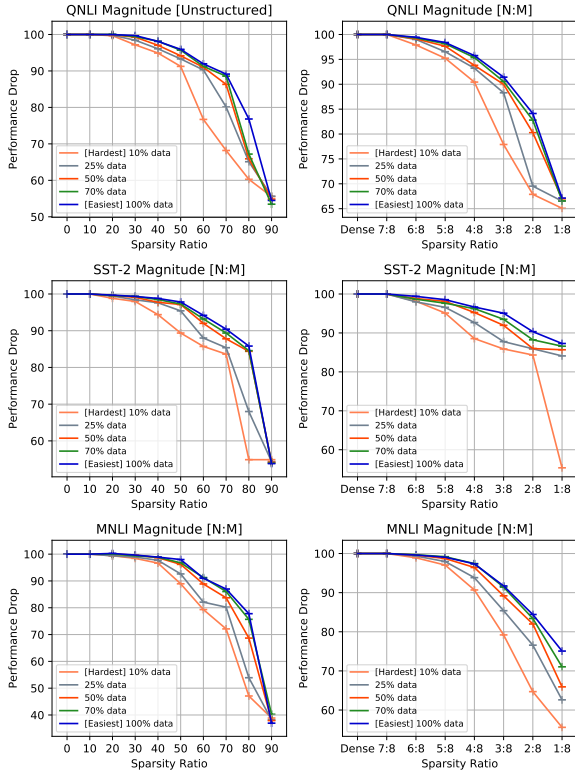
Figure 1. **Task Difficulty Setting 1: Varying target domain data adequacy**: Dense Transfer vs. Sparse Transfer using RoBERTa-Base on various downstream tasks. Task difficulty is measured by the training data volume.

Figure 2. **Task Difficulty Setting 2: Varying option count in multiple-choice QA**. Dense v.s. Sparse subnetwork performance of Vicuna-7B using MMLU. Task difficulty is measured by the number of available choices in a multiple-choice QA setting.

manipulate the option count *for each question from* [2 − 4] which provide a random guess success rate from 50% (2 options) to 25% (4 options). This setting uniquely allows us to control the task difficult for a given task.

Method: To understand the influence of small-magnitude weights, in this setting, we exploit in-context learning setting with a natural prompting approach to present the question and answer options to the LLMs jointly, and have it output the symbol (*e.g.*, "A") associated with its chosen answer option. We fixed 5 in-context question-answer example pairs while . Here, we attempt to understand how the pruned model can effectively reason and successfully associate the correct answer among the given answer options with symbols that represent them, using knowledge remaining within them with increasing difficulty by varying option counts. Figure 2 summarizes our results for this setting across unstructured and structured sparsity patterns.

3.1.3. TASK DIFFICULTY SETTING 3: *Varying context length for Retrieval-Augmented QA*

Rationale: In this task setting, we exploit a highly industry-relevant task of Retrieval-Augmented QA in which given a context, LLMs are prompted to answer questions based on the context. To define a quantitative measure of difficulty
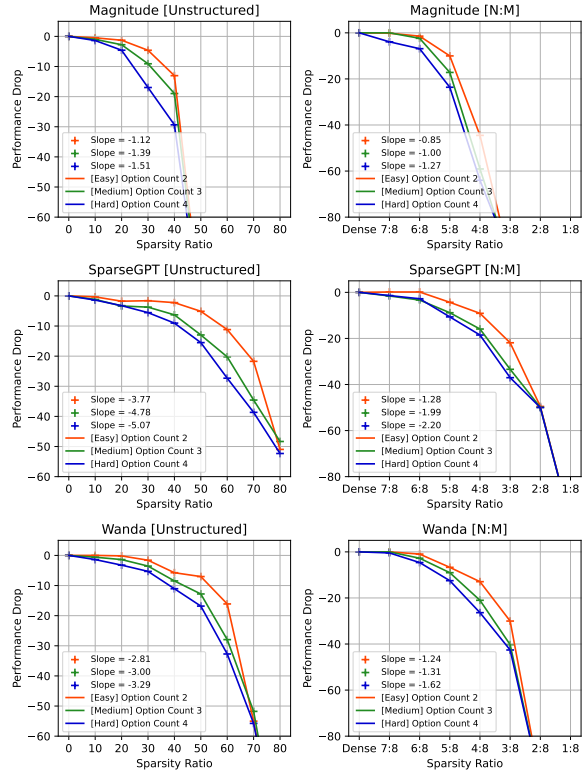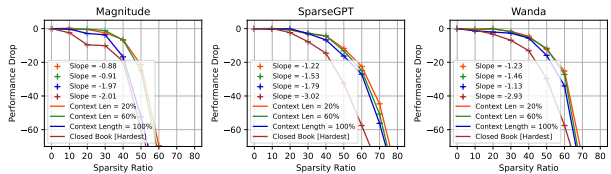

Figure 3. **Task Difficulty Setting 3:Varying context length in Retrieval-Augmented QA**. Dense v.s. Sparse subnetwork performance of Vicuna-7B pruned using TrivaQA Benchmark. Task difficulty is measured by the number of tokens provided in context.

for this task, we propose to *vary the context length ensuring that the correct answer still resides within the provided context*. Retrieval-augmented QA requires LLMs to possess sufficient ability to synthesize long in-context knowledge provided within input prompts, and locate and retrieve correct answers within it. Our novel strategy to **control token counts** within context helps us to define a smooth relation with the task difficulty.

Method: Our evaluation system using TriviaQA includes two high-level components: ❶ *document selection*, selecting the set of documents upon which to condition; and ❷ *document reading*, determining how to incorporate the selected documents into the LLM answer generation process, which requires extracting correct answer phrases from con-
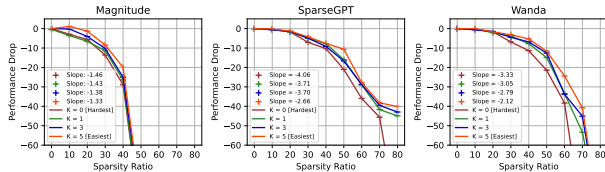
Figure 4. **Task Difficulty Setting 4:Few-shot In-context Learning**. Dense v.s. Sparse subnetwork performance of Vicuna-7B pruned using MMLU Benchmark. Task difficulty is measured by the number of K shot in-context demonstration examples provided to assist multiple-choice QA.

ditioned documents. Given the ground truth, we select x% of tokens around it in the context document from the document selection step, to ensure that the context is complete (*e.g.,* no split sentences). We evaluate the performance of pruned LLMs to investigate their ability to generate answers using the provided x% tokens. We additionally added a closed-book (Ram et al., 2023) setting at the end of the difficult spectrum (highest) because in this setting, LLMs are not provided with any external information and need to rely on their own internal knowledge ingested during pretraining (Jaiswal et al., 2023a).

3.1.4. TASK DIFFICULTY SETTING 4: *Varying number of k-shot examples for in-context learning*

Rationale: Modern-day LLMs have shown a unique ability for out-of-domain generalization using in-context learning (ICL). ICL adapts a model to a task by conditioning it with few-shot demonstrations specified purely via text interaction with the model and it has been observed to significantly improve performance on the downstream application (Brown et al., 2020; Mosbach et al., 2023). In this task setting, we propose to *control the amount of example demonstrations to quantitatively measure the difficulty of the task.* Based on our experimental finding that providing more demonstrations leads to better performance of the dense model, this setting defines the difficulty spectrum directly on the number of example demonstrations provided to the pruned model during the conversation using natural language prompts.

Method: Similar to task setting 2, we use in-context learning to present k example demonstration to the compressed model. For our evaluation, we use MMLU benchmark, and use $k \in \{0, 1, 3, 5\}$ examples from the validation set to control the task difficulty where 0 indicates no example was shown to the model (hardest) while 5 indicate the model generated answer observing 5 examples (easiest). Figure 4 summarizes our results for this setting.

3.1.5. MAIN RESULTS: VALIDATING THE JUNK DNA

❶ **Removal of small-magnitude weights is viable to some extent for easier tasks in comparison to difficult tasks:** In all four task setting's results (Figure 1, 2, 3, and 4), we find that it is feasible to discard around 10-15% of small-

magnitude at once without significantly compromising the performance in unstructured sparsity. This indicates that *for simple tasks, the knowledge encoded in high-magnitude weights is sufficient to handle the task.* For instance, in task setting 4 (Figure 4a), we found that for the easiest task (with 5 ICL demonstration), we can remove small weights up to 14% without any drop in performance in comparison with dense counterpart, which is **not true** for the hardest task (with 0 ICL demonstration).

❷ **Eliminating small weight presents a monotonic impairment relationship with increasing task difficulty:** Across all our experiments, we found a smooth monotonic impairment relationship between removing small weights and the difficulty of the task. As we gradually move to the hard end of the difficulty spectrum, the impact of pruning small weights keeps increasing which shows that the "useless" small weights are imperative to encode crucial knowledge necessary to solve more challenging downstream tasks. The *monotonically increasing theil–sen estimator slope w.r.t. task difficulty provides a quantitative estimate to measure "monotonically impairing" caused by pruning* low-magnitude weights.

❸ **Junk DNA persists in both N:M sparsity and unstructured sparsity, and beyond the magnitude criteria** Both N:M sparsity and unstructured sparsity yield similar observations, indicating the presence of Junk DNA in both settings. We further experiment with two other popular LLM-pruning methods: SparseGPT and Wanda which consider weight importance as a criterion for pruning. Our observations can be summarized as: (1) both these methods (including their N:M setting) corroborate with JunkDNA Hypothesis, (2) JunkDNA can be viewed as a benchmark to develop better pruning algorithms which can minimize the performance gap between easy and hard task, (3) interesting, we observe that these *new sophisticated weight importance based pruning methods tend to benefit simple task more in comparison to challenging tasks*, indicating a need to carefully evaluate true merits of modern compression methods.

3.2. Case study of difficulty spectrum across tasks

3.2.1. TASK DIFFICULTY SETTING 5: *Estimating LLM-facing Task Difficulty by Normalized Human-LLM Performance Gap*

Rationale and Method: We propose a method to gauge complexity by juxtaposing the performance of deep learning models with that of human counterparts. Specifically, we define task difficulty as the disparity in performance between humans and models, normalized by human performance. A more pronounced positive performance gap (for instance, where humans outperform the machine to a greater extent) would signify a higher level of difficulty for the model in handling the given task. Conversely, in cases where the ma-

chine outperforms humans, a larger gap indicates an easier task. The resulting assessment of across-task difficulty is outlined in Table 3. Specifically, we choose a range of downstream tasks, including PIQA, HellaSwag, OpenBookQA and our experiments are conducted using Vicuna-13B.
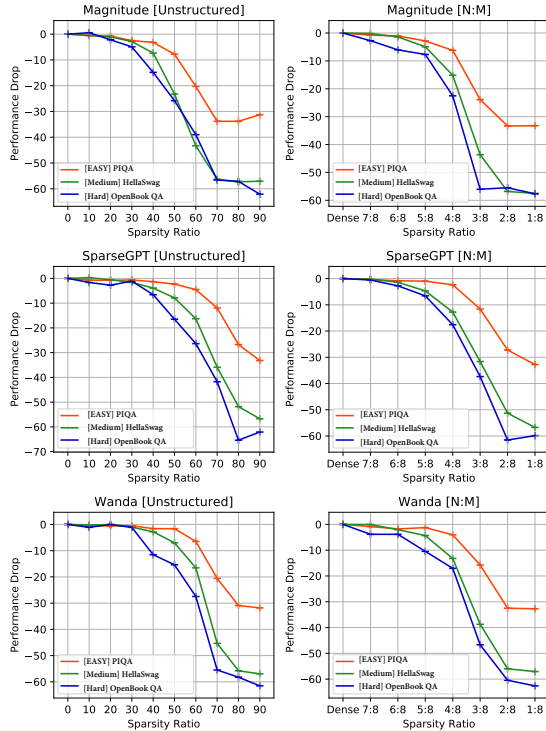

Figure 5. **Across-Task Difficulty via Normalized Human-LLM Performance Gap:** Dense v.s. Sparse subnetwork performance of Vicuna-13B. Task difficulty is measured by Human-LLM Performance gap **normalized** by the dense performance.

3.2.2. TASK DIFFICULTY SETTING 6: *Factoid-based v.s. Multiple-choice QA*

Rationale and Method: In this setting, we compare two popular QA settings: Factoid-based QA and Multiple-Choice QA. A typical Factoid-QA task aims to answer natural language questions using facts, i.e., entities or attributes of knowledge ingested within them during pre-training. On the other hand, multiple-choice QA setting present the question and answer options to the LLMs jointly, and have it output the symbol (*e.g.*, "A") associated with its chosen answer option. Unlike multiple-choice QA, factoid-based QA requires LLMs' ability to answer natural language questions using facts using its own knowledge and possess the ability to generate full answers unlike output a symbol, which makes it inherently a more challenging task.

3.2.3. MAIN RESULTS: VALIDATING THE JUNK DNA HYPOTHESIS

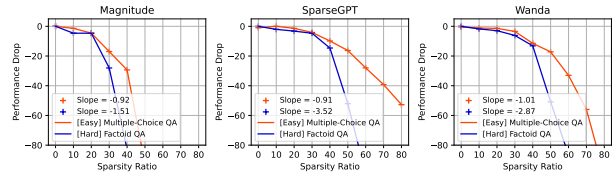The findings depicted in Figure 5, 6 echo the conclusions drawn in Section 3.1, once more providing robust support

Figure 6. **Across-Task Difficulty for Factoid-based QA and Multiple-Choice QA::** Dense v.s. Sparse subnetwork performance of Vicuna-7B. Task difficulty is measured by Human-LLM Performance gap **normalized** by the dense performance.

for the validity of the *Junk DNA hypothesis across a broad spectrum of task categories*. While it may be feasible to remove small-magnitude weights without significant repercussions in simpler tasks, these pre-trained small weights contain vital downstream knowledge essential for tackling challenging tasks and thus are no longer dispensable suggesting their importance -manifested as monotonic relationships between the amount of pruned weights and the difficult level of tasks that start to be impaired.

4. Does Quantization Also Yield Monotonic Impairment Across Difficulty Spectrum?

Pruning and *quantization* are two popularly used remedies to reduce the overheads of billions of parameters in current LLMs. Pruning techniques primarily shrink network sizes by removing specific weights from the model – essentially setting them to zero. Quantization methods aim to quantize parameters into low bit-precision to reduce compute and memory budget. Across recently developed quantization methods for LLMs (Lin et al., 2023a; Frantar et al., 2022a; Dettmers et al., 2023), we adopted GPTQ for the quantization of all linear layers (exactly similar to our pruning settings) to study the monotonic impairment induced across the task difficulty spectrum. In this section, we firstly ask: *How does the ratio of compression using quantization reflect on the performance of the downstream tasks? Does quantization strength dictate any performance pattern across the difficulty spectrum of the underlying task?*

To this end, Figure 7 presents our head-to-head comparison for one-shot low-magnitude pruning and GPTQ quantization (Frantar et al., 2022b) various compression strengths. We adopted retrieval-augmented QA (task setting 3) in which task difficulty is controlled by varying the context length (token counts) accommodating the correct answer phrase, and variable option count in multiple-choice QA (task setting 2) for our experiments.

Based on our experimental results, we summarize: ❶ while pruning shows a monotonic performance degradation with increasing compression ratio, quantization does not exhibit the smooth impairing impact of compression; ❷ removal of low magnitude weights using pruning tends to have a monotonically increasing impairing impact if we move along the
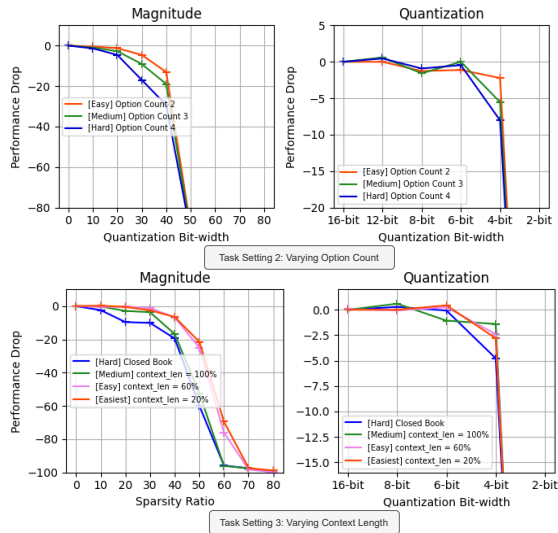
Figure 7. **How is pruning special?** Performance comparison of pruning and quantization with varying compression ratios on our task difficulty spectrum. We can observe the **monotonic impairment** of pruning across task difficulty and pruning ratio. On the contrary, quantization **fail** to capture this monotonic behavior across task difficulty and compression ratio.

extreme end of our task difficulty spectrum, which is again not convincingly demarcated either by quantization. It is interesting to observe that quantization to a certain degree tends to benefit the overall performance but again without any clear pattern across the difficulty spectrum.

How Is Pruning Special? It is interesting to see the unique ability of pruning to identify a critical subset of weights that can independently complete specific set of tasks the same way as the dense model - but not quantization. This naturally reminds us the Lottery Ticket Hypothesis (LTH) (Frankle et al., 2019; Chen et al., 2020), stating that there exists a pruned subnetwork that can match the performance of its dense counterpart; meanwhile no similar conclusion has been drawn for quantized networks at scale. Such discrepancy seems to echo the two methods' different impairment behaviors (monotonic or not across task difficulty levels). We additionally experimented with low-rank compression (Yu et al., 2017) (not the low-rank weight updates used in parameter-efficient tuning (Hu et al., 2021)) and found its behavior closer to pruning than quantization (Appendix E).

The classical LTH (with iterative pruning and re-training) has not yet been studied for modern-scale LLMs due to its prohibitive computational costs. Previous work in smaller models did find reusable sparse substructures that contain sufficient information to handle a few similar tasks altogether (Chen et al., 2020; Yu et al., 2020). Recent attempt (Jaiswal et al., 2023b) demonstrated a similar property on LLMs for a range of tasks, using cheap one-shot pruning without re-training; but the authors also recognized the "sweetpoint sparsity" was correlated to the difficulty of

Task Setting	Difficulty (↓)	Pruning	Quantization
Task Setting 3	Closed Book	0.996	0.799
	Context_len = 100%	0.996	0.794
	Context_len = 60%	0.975	0.399
	Context_len = 20%	0.987	0.316
Task Setting 2	Option Count 4	0.979	0.872
	Option Count 3	1.0	0.799
	Option Count 2	1.0	0.666

Table 2. Spearman's rank correlation of pruning and quantization for task with varying difficulty levels. Unlike quantization high Spearman correlation for pruning uniquely captures monotonic impairment induced by pruning.

downstream tasks involved (although they did not provide a concrete way to quantify such). As a modern LLM can perform an infinite range of downstream tasks in zero- or few-shot, Junk DNA Hypothesis can be viewed as a scaled extension of LTH, which suggests that: if the task difficulty spectrum (much broader than before) is taken into consideration, then there might not exist a single "lottery" subset of weight, that can be removed without causing irreparable damage for the performance of *all tasks* - although one might still be able to find a "lottery" for *some (easy) tasks*.

5. Are Pre-trained Magnitude Values Indeed the True Gem?

Having recognized the pivotal role of small weights in downstream adaptation, particularly in relation to task difficulty, our next objective is to delve into the foundational factors contributing to the crucial function of small weights and whether they can be recovered with fine-tuning. More specifically, we intend to validate if pruning leads to **irreparable** loss of knowledge. Our primary research inquiries are:

• *Which one holds greater significance: the knowledge (weight-values) stored in pre-trained small-magnitude weights, or the potential to adjust these weights through fine-tuning?*

• *Is it possible to recover the knowledge embedded in pre-trained small-magnitude weights if we prune them and allow them to grow freely during fine-tuning?*

Method: We explore four comparison methods: (1) **Dense Transfer**: as described in Section 3.1.1; (2) **Dense Transfer with (Partial) Freezing**: a dense model where small-magnitude weights remain fixed during fine-tuning; (3) **Sparse Transfer**: as in Section 3.1.1; (4) **Sparse to Dense Transfer**: small-magnitude weights are initially pruned after pre-training, and subsequently during fine-tuning, those pruned weights are allowed to gradually regain non-zero values. This approach also aids in determining whether the knowledge within small-magnitude pre-trained weights is essential for performance or if their adaptability during fine-tuning takes precedence. We pick Task Setting 1 (within the same task) and Task Setting 5 (across different task), to report their performance using RoBERTa-Large, on MNLI, QNLI, SST-2, as well as CSQA and WinoGrande. The difficulty spectrum normalized by human performance for the
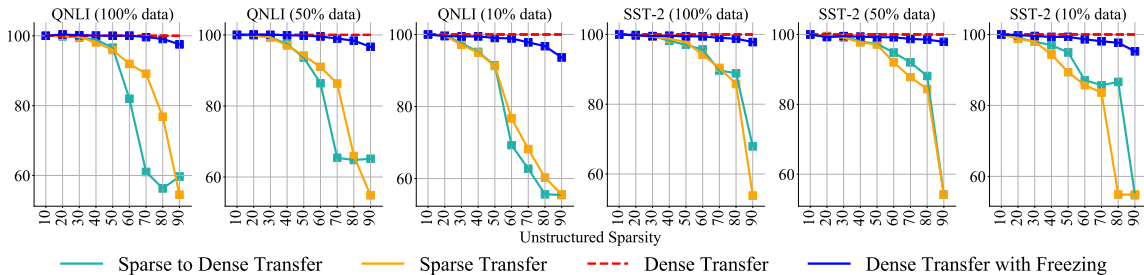
Figure 8. **Varying target domain data adequacy**: Four different fine-tuning settings with RoBERTa-Base on various downstream tasks. All performance is normalized by the one of Dense Transfer.
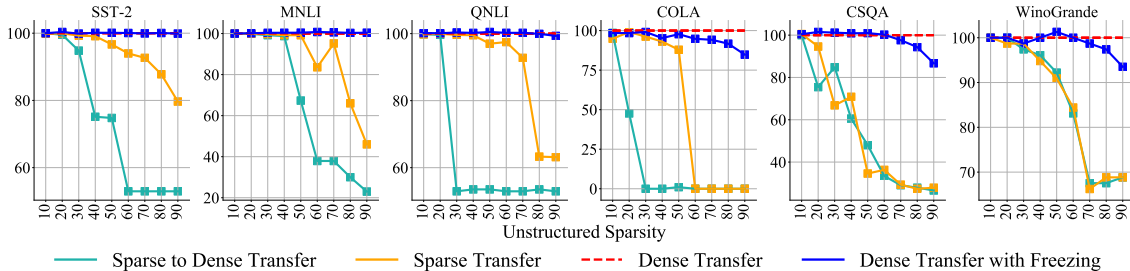

Figure 9. **Across-Task Difficulty via Normalized Human-LLM Performance Gap**: Four different fine-tuning settings with RoBERTa-Large on various downstream tasks. All performance is normalized by the one of Dense Transfer.

aforementioned tasks is presented in Appendix B.

Table 3. Measuring the Across-Task Difficulty by the Performance Difference between humans and models (normalized by human performance): the larger (positive) margin, the more difficult for the model.

	PIQA	HellaSwag	OpenBookQA
Human	95.0	95.6	92.0
Vucuna13B	78.35	59.68	36.40
"Task Difficulty" (%)	[Easy] 17.53	[Medium] 37.57	[Hard] 60.43

Results: The outcomes for both within-task difficulty and across-task difficulty are illustrated in Figure 8 and Figure 9, respectively. Below, we outline our key observations:

❶ **Pre-trained small weights harbor vital downstream knowledge, beyond mere free parameters.** Across both task-difficulty metrics, it becomes evident that settings preserving the pre-trained values of small weights—namely, Dense Transfer and Dense Transfer with Freezing—achieve superior performance when compared to the other two settings. The removal of small-magnitude weights from pre-trained models results in significant performance degradation, even when we permit the pruned weights to regenerate during fine-tuning. This observation strongly bolsters the Junk DNA Hypothesis, indicating that small-magnitude weights are far from redundant; rather, they house sophisticated knowledge crucial for downstream adaptation. This knowledge proves challenging to re-gain through fine-tuning, if these initial pre-trained weights are eliminated.

❷ **Freezing without updating yields commendable results.** Remarkably, on simpler tasks like SST-2, MNLI, and QNLI, maintaining an overwhelmingly large portion (90%) of small-magnitude weights in a frozen state leads

to equally impressive performance without any loss. Even on more intricate tasks such as COLA, CSQA, and Wino-Grande, freezing up to 70% of small-magnitude weights results in no discernible performance dip. This suggests that for easier tasks, the knowledge embedded in pre-trained small-magnitude weights is already more than sufficient. However, for more challenging tasks, allowing for moderate updates to all pre-trained weights remains essential.

6. Conclusion and Future Work

In this study, we embark on an exploration to validate the prevailing belief that deep network weights are excessively redundant, allowing for a substantial pruning of parameters without compromising performance. Our research presents a compelling counter-argument by unearthing the previously overlooked yet intricate role of small-magnitude weights, closely tied to the difficulty level of downstream tasks. We found small-magnitude pre-trained weights encode vital knowledge essential for tackling difficult downstream tasks - manifested as monotonic relationship between increasing task difficulty and impairment induced by removing small weights. Moreover, we argue that these seemingly inconsequential weights can result in irreparable loss of knowledge and performance degradation in difficult tasks, even if downstream continual training is allowed. Meanwhile, quantization fails to exhibit similar effects and hence does not as convincingly disentangle this task- difficulty information as pruning. As the number of parameters in deep networks continues to grow exponentially, our findings prompt the exploration of directions such as task-complexity-dependent dynamic inference and network self-slimmable properties.

Impact Statement

This study presents the "Junk DNA Hypothesis," a nuanced perspective on the role of small-magnitude weights in large language models (LLMs), challenging the traditional view of their redundancy. Our findings highlight the importance of these weights in complex tasks and suggest the need for further exploration into dynamic inference and network slimming techniques based on task complexity. This could lead to improved methods for more efficient inference, and contribute to the "GreenAI" goal.

Acknowledgments

References

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Carey, N. *Junk DNA: a journey through the dark matter of the genome*. Columbia University Press, 2015.

Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Wang, Z., and Carbin, M. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems*, 33:15834–15846, 2020.

Dettmers, T., Svirschevski, R., Egiazarian, V., Kuznedelev, D., Frantar, E., Ashkboos, S., Borzunov, A., Hoefler, T., and Alistarh, D. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *ArXiv*, abs/2306.03078, 2023. URL https://api.semanticscholar.org/CorpusID:259076379.

Dong, X., Chen, S., and Pan, S. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in Neural Information Processing Systems*, 30, 2017.

Evci, U., Gale, T., Menick, J., Castro, P. S., and Elsen, E. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning*, pp. 2943–2952. PMLR, 2020.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Frankle, J., Dziugaite, G. K., Roy, D. M., and Carbin, M. The lottery ticket hypothesis at scale. *arXiv preprint arXiv:1903.01611*, 2019.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Frantar, E. and Alistarh, D. Sparsegpt: Massive language models can be accurately pruned in one-shot, 2023.

Frantar, E., Kurtic, E., and Alistarh, D. M-fac: Efficient matrix-free approximations of second-order information. *Advances in Neural Information Processing Systems*, 34: 14873–14886, 2021.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ArXiv*, abs/2210.17323, 2022a. URL https://api.semanticscholar.org/CorpusID:253237200.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022b.

Gale, T., Elsen, E., and Hooker, S. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

Hajimolahoseini, H., Rezagholizadeh, M., Partovinia, V., Tahaei, M., Awad, O. M., and Liu, Y. Compressing pre-trained language models using progressive low rank decomposition. *Advances in Neural Information Processing Systems*, 2021.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016.

Hassibi, B. and Stork, D. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5, 1992.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jaiswal, A., Gan, Z., Du, X., Zhang, B., Wang, Z., and Yang, Y. Compressing llms: The truth is rarely pure and never simple. *arXiv preprint arXiv:2310.01382*, 2023a.

Jaiswal, A., Liu, S., Chen, T., and Wang, Z. The emergence of essential sparsity in large pre-trained models: The weights that matter. *arXiv preprint arXiv:2306.03805*, 2023b.

Jaiswal, A. K., Ma, H., Chen, T., Ding, Y., and Wang, Z. Training your sparse neural network better with any mask. In *International Conference on Machine Learning*, pp. 9833–9844. PMLR, 2022.

Jaiswal, A. K., Liu, S., Chen, T., Ding, Y., and Wang, Z. Instant soup: Cheap pruning ensembles in a single pass can draw lottery tickets from large models. In *International Conference on Machine Learning*, pp. 14691–14701. PMLR, 2023c.

Jiang, H., Zhan, K., Qu, J., Wu, Y., Fei, Z., Zhang, X., Chen, L., Dou, Z., Qiu, X., Guo, Z., et al. Towards more effective and economic sparsely-activated model. *arXiv preprint arXiv:2110.07431*, 2021.

Jiang, K., Wu, D., and Jiang, H. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. pp. 318–323. Association for Computational Linguistics, June 2019. doi: 10.18653/v1/N19-1028. URL https://aclanthology.org/N19-1028.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Kim, S., Gholami, A., Yao, Z., Mahoney, M. W., and Keutzer, K. I-bert: Integer-only bert quantization. In *International conference on machine learning*, pp. 5506–5518. PMLR, 2021.

Kundu, S., Nazemi, M., Beerel, P. A., and Pedram, M. Dnr: A tunable robust pruning framework through dynamic network rewiring of dnns. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, pp. 344–350, 2021.

Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., and Alistarh, D. The optimal bert surgeon: Scalable and accurate second-order pruning for large language models. *arXiv preprint arXiv:2203.07259*, 2022.

Lagunas, F., Charlaix, E., Sanh, V., and Rush, A. M. Block pruning for faster transformers. *arXiv preprint arXiv:2109.04838*, 2021.

LeCun, Y., Denker, J. S., and Solla, S. A. Optimal brain damage. In *Advances in neural information processing systems*, pp. 598–605, 1990.

Li, Z., You, C., Bhojanapalli, S., Li, D., Rawat, A. S., Reddi, S. J., Ye, K., Chern, F., Yu, F., Guo, R., et al. Large models are parsimonious learners: Activation sparsity in trained transformers. *arXiv preprint arXiv:2210.06313*, 2022.

Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration. *ArXiv*, abs/2306.00978, 2023a. URL https://api.semanticscholar.org/CorpusID:258999941.

Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., and Han, S. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023b.

Liu, S., Yin, L., Mocanu, D. C., and Pechenizkiy, M. Do we actually need dense over-parameterization? in-time over-parameterization in sparse training. *arXiv preprint arXiv:2102.02887*, 2021.

Liu, S., Chen, T., Chen, X., Shen, L., Mocanu, D. C., Wang, Z., and Pechenizkiy, M. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. *arXiv preprint arXiv:2202.02643*, 2022.

Liu, S., Chen, T., Zhang, Z., Chen, X., Huang, T., Jaiswal, A., and Wang, Z. Sparsity may cry: Let us fail (current) sparse neural networks together! *arXiv preprint arXiv:2303.02141*, 2023.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lv, X., Zhang, P., Li, S., Gan, G., and Sun, Y. Lightformer: Light-weight transformer using svd-based weight transfer and parameter sharing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10323–10335, 2023.

Ma, X., Fang, G., and Wang, X. Llm-pruner: On the structural pruning of large language models. *arXiv preprint arXiv:2305.11627*, 2023.

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

Mocanu, D. C., Mocanu, E., Stone, P., Nguyen, P. H., Gibescu, M., and Liotta, A. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9 (1):2383, 2018.

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

Mosbach, M., Pimentel, T., Ravfogel, S., Klakow, D., and Elazar, Y. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*, 2023.

Mostafa, H. and Wang, X. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, 2019.

Mozer, M. C. and Smolensky, P. Using relevance to reduce network size automatically. *Connection Science*, 1(1): 3–16, 1989.

Nangia, N. and Bowman, S. R. Human vs. muppet: A conservative estimate of human performance on the glue benchmark. *arXiv preprint arXiv:1905.10425*, 2019.

Nvidia. Nvidia a100 tensor core gpu architecture. *https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf*, 2020.

Ohno, S. So much "junk" dna in our genome. *Brookhaven symposia in biology*, 23:366–70, 1972.

Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., and Shoham, Y. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*, 2023.

Sanh, V., Wolf, T., and Rush, A. Movement pruning: Adaptive sparsity by fine-tuning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20378–20389. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/eae15aabaa768ae4a5993a8a4f4fa6e4-Paper.pdf.

Sharma, P., Ash, J. T., and Misra, D. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. *arXiv preprint arXiv:2312.13558*, 2023.

Singh, S. P. and Alistarh, D. Woodfisher: Efficient second-order approximation for neural network compression. *Advances in Neural Information Processing Systems*, 33: 18098–18109, 2020.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Wang, C., Grosse, R., Fidler, S., and Zhang, G. Eigendamage: Structured pruning in the kronecker-factored eigenbasis. In *International Conference on Machine Learning*, pp. 6566–6575. PMLR, 2019.

Xu, D., Yen, I. E., Zhao, J., and Xiao, Z. Rethinking network pruning–under the pre-train and fine-tune paradigm. *arXiv preprint arXiv:2104.08682*, 2021.

Yin, L., Li, G., Fang, M., Shen, L., Huang, T., Wang, Z., Menkovski, V., Ma, X., Pechenizkiy, M., and Liu, S. Dynamic sparsity is channel-level sparsity learner. *arXiv preprint arXiv:2305.19454*, 2023.

Yu, H., Edunov, S., Tian, Y., and Morcos, A. S. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. In *8th International Conference on Learning Representations*, 2020.

Yu, X., Liu, T., Wang, X., and Tao, D. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7370–7379, 2017.

Yuan, G., Ma, X., Niu, W., Li, Z., Kong, Z., Liu, N., Gong, Y., Zhan, Z., He, C., Jin, Q., et al. Mest: Accurate and fast memory-economic sparse training framework on the edge. *Advances in Neural Information Processing Systems*, 34: 20838–20850, 2021.

Zafrir, O., Larey, A., Boudoukh, G., Shen, H., and Wasserblat, M. Prune once for all: Sparse pre-trained language models. *arXiv preprint arXiv:2111.05754*, 2021.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zeng, W. and Urtasun, R. Mlprune: Multi-layer pruning for automated neural network compression. 2018.

Zhang, Q., Zuo, S., Liang, C., Bukharin, A., He, P., Chen, W., and Zhao, T. Platon: Pruning large transformer models with upper confidence bound of weight importance. In *International Conference on Machine Learning*, pp. 26809–26823. PMLR, 2022.

Zheng, Y., Josefowicz, S., Chaudhry, A., Peng, X. P., Forbush, K., and Rudensky, A. Y. Role of conserved non-coding dna elements in the foxp3 gene in regulatory t-cell fate. *Nature*, 463(7282):808–812, 2010.

Zhou, A., Ma, Y., Zhu, J., Liu, J., Zhang, Z., Yuan, K., Sun, W., and Li, H. Learning n: m fine-grained structured sparse neural networks from scratch. *arXiv preprint arXiv:2102.04010*, 2021.

A. Related Work

A.1. Classical Pruning and Sparse Neural Networks

Pruning removes specific parts of a deep neural network, such as weights, neurons, or filters. The initial purpose of pruning is retrospectively to accelerate the model at inference time (a.k.a., post-training sparsification (Mozer & Smolensky, 1989; LeCun et al., 1990). Post-training sparsification has been well studied and results in various mature criteria that can be generally categorized into zero-order methods (Han et al., 2016; Gale et al., 2019), first-order methods (Molchanov et al., 2016; Sanh et al., 2020; Jiang et al., 2021), and second-order methods (LeCun et al., 1990; Hassibi & Stork, 1992; Dong et al., 2017) - the last usually achieve higher performance but also are more expensive due to the Hessian calculation, leading to the development of many Hessian approximation approaches (Zeng & Urtasun, 2018; Wang et al., 2019; Singh & Alistarh, 2020; Kurtic et al., 2022). The Lottery Ticket Hypothesis (LTH) (Frankle & Carbin, 2019) utilizes iterative magnitude pruning (IMP) to identify a subnetwork at initialization that can be re-trained independently to the original dense network's performance. Sparse training (Mocanu et al., 2018; Jaiswal et al., 2022; Mostafa & Wang, 2019; Evci et al., 2020; Liu et al., 2021; Yuan et al., 2021; Yin et al., 2023; Kundu et al., 2021), on the other hand, starts with a (random) sparse network and updates network connectivity during training to search for good sparse neural network without any pre-training and dense training steps.

A.2. Sparsity in Large-Scale Models

The advent of large-scale pre-trained models has led to the development of advanced post-training pruning methods, aiming to enhance the cost-effectiveness of these expansive models (Sanh et al., 2020; Chen et al., 2020; Jaiswal et al., 2023c; Zafrir et al., 2021; Kurtic et al., 2022; Xu et al., 2021; Lagunas et al., 2021; Zhang et al., 2022; Frantar et al., 2021; Jaiswal et al., 2023b; Ma et al., 2023). Among them, Frantar et al. (2021) extend second-order pruning to the BERT-level scale, enabling the pruning of blocks of weights and achieving state-of-the-art results for sparse BERT. Frantar & Alistarh (2023) introduce SparseGPT for pruning large language models (LLMs) in a single shot without requiring re-training or fine-tuning. They leverage column-wise second-order pruning, and successfully remove 100B weights from OPT-175B without a significant increase in perplexity. More recently, Sun et al. (2023) propose a straightforward pruning method that takes both weights and activations into account, demonstrating comparable performance to Frantar & Alistarh (2023). Li et al. (2022) reveal that activation sparsity is a prevalent phenomenon in Transformers (90% of intermediate output), yielding another opportunity for acceleration. Liu et al. (2023) introduce a large-scale SMC-Bench, indicating that state-of-the-art magnitude- and/or gradient-based sparse algorithms fall short when applied out-of-the-box to larger-scale models and a selected of complex downstream tasks. Our study is inspired by Liu et al. (2023), but with significantly expanded experiment scales, versatile task choices, concrete task difficulty definitions, and richer insights.

B. Across-Task Difficulty by the Performance Difference between humans and models.

Table 4. Measuring the Across-Task Difficulty by the Performance Difference between humans and models (normalized by human performance): the larger (positive) margin, the more difficult for the machine. Human performance is obtained from Nangia & Bowman (2019). The more challenging task is marked in bold.

	Single Sentence		Sentence Similarity		Natural Language Inference		Commonsense Reasoning	
	SST-2	COLA	QQP	STS-B	QNLI	RTE	WinoGrande	CSQA
Human	97.8	66.4	80.4	92.7	91.2	93.6	94.0	89.0
RoBERTa-Large	96.2	64.9	91.8	92.2	94.4	84.3	78.1	72.1
"Task Difficulty" (%)	1.64	**2.26**	-14.18	**0.54**	-3.51	**9.94**	16.91	**18.99**

We acknowledge the limitations in our approach, such as when both humans and LLMs perform poorly on a task, potentially indicating high difficulty not reflected in our 'relative' gap metric. We hope our exploration can inspire more efforts in a more general and rigorous assessment of cross-task difficulty.

C. Small-Magnitude Weights Contribute to Loss Basin Preservation

We also investigate the potential reasons behind the substantial performance drop resulting from the removal of small-magnitude weights on harder tasks. Our analysis revolves around the loss landscape, and we conjecture that small-magnitude weights play a significantly more crucial role in preserving the loss basin of the dense model on harder tasks compared to
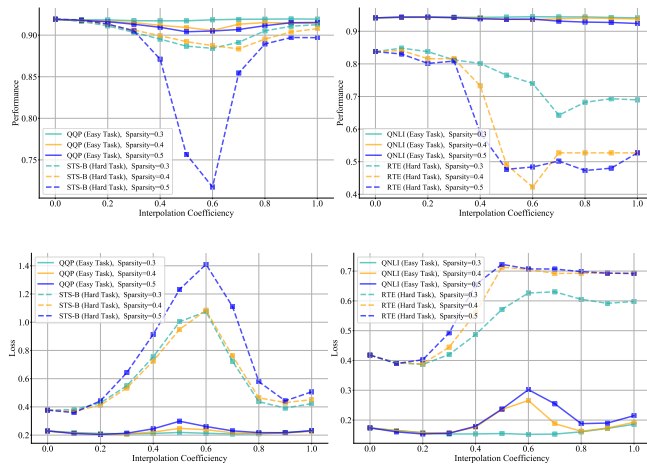
Figure 10. Linear interpolation from the Dense Transfer (**Left**) model to its corresponding Sparse Transfer models (**Right**) on easy and harder tasks (in terms of across-task difficulty).

easier ones. Consequently, the absence of these small weights disrupts the optimal basin, leading to a considerable loss of performance.

To test our conjecture, we utilize the linear mode connectivity (LMC) metric proposed by (Frankle et al., 2020) between the solution produced by Sparse Transfer and that of Dense Transfer. Specifically, we perform linear interpolation between the fine-tuned model of Dense Transfer (θ_d) and the fine-tuned model of Sparse Transfer (θ_s), denoted as $\tilde{\theta} = \alpha\theta_s + (1-\alpha)\theta_d$. We conduct two sets of comparisons, namely QQP vs. STS-B and QNLI vs. RTE, and report the performance and loss in Figure 5. Our findings reveal that both sparse and dense models remain linearly connected, with minimal or no increase in loss barrier for easy tasks (i.e., QQP and RTE) when a certain portion of small-magnitude weights is removed. However, a significant increase in the loss barrier is observed when the same number of weights is removed for harder tasks. This observation strongly supports the concept of "Junk DNA", emphasizing the vital role of small-magnitude weights in ensuring that the fine-tuned model resides in the optimal basin. In contrast, the removal of these weights would lead to the destruction of this optimal basin, which is challenging to fix through fine-tuning, causing a notable decline in performance.

D. Dataset Details

D.1. Massive Multitask Language Understanding (MMLU)

MMLU (Hendrycks et al., 2020) is a new benchmark designed to measure knowledge acquired during pretraining by evaluating models exclusively in zero-shot and few-shot settings. This makes the benchmark more challenging and more similar to how we evaluate humans. The benchmark covers 57 subjects across STEM, the humanities, the social sciences, and more. It ranges in difficulty from an elementary level to an advanced professional level, and it tests both world knowledge and problem solving ability. Subjects range from traditional areas, such as mathematics and history, to more specialized areas like law and ethics. The granularity and breadth of the subjects makes the benchmark ideal for identifying a model's blind spots.

D.2. Freebase Factoid Question Answering

FreebaseQA (Jiang et al., 2019) is a data set for open-domain QA over the Freebase knowledge graph. The question-answer pairs in this data set are collected from various sources, including the TriviaQA data set and other trivia websites (QuizBalls, QuizZone, KnowQuiz), and are matched against Freebase to generate relevant subject-predicate-object triples that were further verified by human annotators. As all questions in FreebaseQA are composed independently for human contestants in various trivia-like competitions, this data set shows richer linguistic variation and complexity than existing QA data sets, making it a good test-bed for emerging KB-QA systems.

D.3. TrviaQA Dataset

TriviaQA (Joshi et al., 2017) is a popular reading comprehension dataset which includes 95K question-answer pairs authored by trivia enthusiasts and independently gathered evidence documents, six per question on average, that provide

14

high-quality distant supervision for answering the questions. TriviaQA consists of fairly complex compositional questions with considerable syntactic and lexical variability between questions and corresponding answer-evidence sentences, making it a challenging testbed for our evaluation.

D.4. Glue Benchmark

General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) is a collection of nine natural language understanding tasks, including single-sentence tasks CoLA and SST-2, similarity and paraphrasing tasks MRPC, STS-B and QQP, and natural language inference tasks MNLI, QNLI, RTE and WNLI. GLUE is designed to favor and encourage models that share general linguistic knowledge across tasks.

D.5. OpenbookQA Benchmark

OpenBookQA (Mihaylov et al., 2018) is a new kind of question-answering dataset modeled after open book exams for assessing human understanding of a subject. It consists of 5,957 multiple-choice elementary-level science questions (4,957 train, 500 dev, 500 test), which probe the understanding of a small "book" of 1,326 core science facts and the application of these facts to novel situations. For training, the dataset includes a mapping from each question to the core science fact it was designed to probe. Answering OpenBookQA questions requires additional broad common knowledge, not contained in the book. The questions, by design, are answered incorrectly by both a retrieval-based algorithm and a word co-occurrence algorithm. Additionally, the dataset includes a collection of 5,167 crowd-sourced common knowledge facts, and an expanded version of the train/dev/test questions where each question is associated with its originating core fact, a human accuracy score, and a clarity score.

D.6. HellaSwag

HellaSWAG (Zellers et al., 2019) is a dataset for studying grounded commonsense inference. It consists of 70k multiple choice questions about grounded situations: each question comes from one of two domains - activitynet or wikihow - with four answer choices about what might happen next in the scene. The correct answer is the (real) sentence for the next event; the three incorrect answers are adversarially generated and human verified, so as to fool machines but not humans.

E. Low-Rank Compression

An alternative way of compressing the model is reducing the rank of weight matrices by retaining the top k components identified through Singular Value Decomposition (SVD) (Lv et al., 2023; Hajimolahoseini et al., 2021; Sharma et al., 2023). In our study, we specifically modify the rank of k using SVD decomposition across all linear layers in the transformer block of the Vicuna-7B model, ranging from 4000 to 100 ranks.. We then evaluate the compressed model's performance in two task difficulty settings: ❶ Settings 2: *Varying the Option Count in Multiple-choice QA Setting* and Settings 3:❷ *Varying Context Length for Retrieval-Augmented QA*.
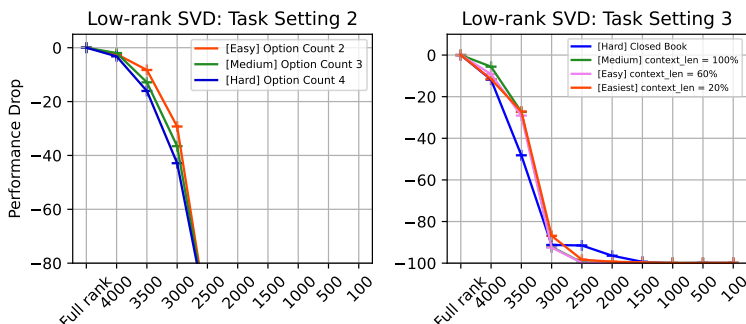

Figure 11. Low-Rank Compression using SVD.

We noticed the concurrent work (Sharma et al., 2023) suggesting layer-selective low-rank compression of weights often improves LLM reasoning and generalization, without needing no re-training needed. We however note that requires careful

Table 5. Low-Rank Compression using SVD.

	Context Length				Option Count		
	Closed Book	100%	60%	20%	2	3	4
Spearman's Rank	0.899	0.942	0.905	0.912	0.933	0.956	0.956

selection of layer & rank. In comparison, we apply "uniform" SVD to all linear weight layers (except embedding), to make fair comparisons to pruning/quantization. We leave the examination of more sophisticated methods such as (Sharma et al., 2023) as future work.