Depth as a Scaling Vector: Simple Pruning and Evaluation of Emergent Abilities in Pruned LLMs

Chang Liu[†], **Arjun Choudhry**[†], **Yifu Cai, Nina Żukowska, Mononito Goswami, Artur Dubrawski**[*]
Auton Lab, Carnegie Mellon University
{changl8,awd}@andrew.cmu.edu

Abstract

The evolving lifecycle of large language models (LLMs) calls for effective strategies for scaling them down for deployment without sacrificing core capabilities. In this work, we investigate depth as a primary architectural scaling vector, introducing simple methods for pruning layers of LLMs, and systematically evaluate how such scaling affects the emergent abilities of LLMs. Our evaluations demonstrate that these methods offer a practical path to facilitate LLM deployment, significantly reducing computational demands while retaining the emergent abilities that make these models powerful and attractive in a wide range of applications.

1 Introduction

The widespread success of large language models (LLMs) has introduced new challenges throughout their evolving lifecycle, from initial training to widespread deployment. One primary bottleneck is caused by the immense computational and memory footprints of LLMs, which limit their accessibility and practicality in resource-constrained environments, when performance is also critical.

Strategically removing groups of parameters, pruning has emerged as a key method to address this challenge [1]. Much of the early focus was on width pruning, which removes redundant weights or neurons within layers [2, 3]. More recently, depth pruning – the removal of entire layers – has been explored as a promising alternative that can offer hardware-agnostic acceleration [4, 5, 6]. However, previous work rarely prunes LLMs to less than 50% of their original sizes and lacks systematic and unified evaluations of the emergent abilities of the pruned model in complex reasoning tasks.

In this paper, we investigate whether simple, direct depth pruning methods can preserve emergent abilities even under aggressive compression rates, where fewer than half of the layers are retained. Our contributions are threefold: (1) We introduce simple, problem-space-agnostic depth pruning methods to scale down LLMs for deployment. (2) We perform a systematic and unified evaluation to determine how the emergent abilities crucial for complex reasoning are affected by aggressive architectural compression using a wide range of state-of-the-art pruning strategies. (3) We demonstrate that these simple pruning methods can preserve key emergent capabilities even at high model sparsities, offering a practical pathway to scale down LLMs for effective deployment. We open-source our code in `https://github.com/ChangLiu-DrPatient/SimpleDepthPruning`.

2 Related work

Model pruning offers a practical strategy for scaling LLMs down for deployment. Early attempts focus on width pruning, which sparsifies LLMs by removing weights within layers. For example,

[1]Corresponding author.
[2]These authors contributed equally.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling.

`Wanda` [2] removes weights with the smallest magnitudes multiplied by their corresponding input activations. `SparseGPT` [3] prunes LLMs one-shot and treats pruning as a large-scale sparse regression problem. More recently, depth pruning presents an alternative scaling vector by removing entire layers, effectively providing more uniform and hardware-agnostic acceleration. `ShortGPT` [4] selects layers based on layer importance scores that measure the similarity between each layer's input and output. `Shortened Llama` [5] removes layers based on the influence of each layer on the model output perplexity on a set of calibration examples. `LaCo` [6] reduces the size of the model by merging multiple subsequent layers into a single preceding layer, adding the parameter differences of the later layers to the earlier one in an iterative process guided by output similarity. However, these methods have been evaluated on different sets of benchmarks, which hinders direct comparison. Our work proposes simpler alternatives to these pruning methods and provides a systematic evaluation of how such pruning affects the emergent abilities that define modern LLMs.

3 Quite simple depth pruning methods

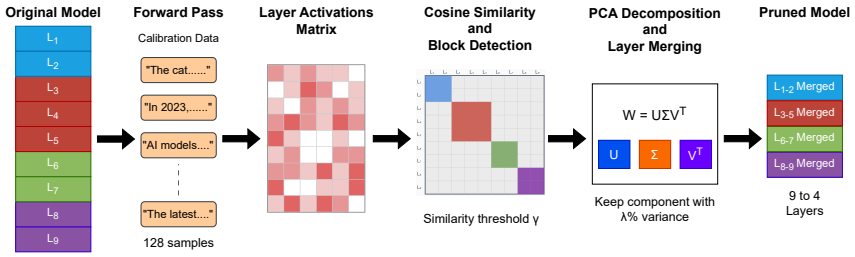

Figure 1: Overview of our `LayerMergeAct` depth pruning method.

We introduce simple depth pruning methods that effectively scale down model sizes to facilitate deployment. In all our methods, we only modify the decoder layers of an LLM, keeping the other components (e.g., embedding layers) intact. Specifically, with $\{L_1, L_2, \cdots, L_m\}$ being the original LLM decoder layers, we output a pruned model with decoder layers $\{\tilde{L}_1, \tilde{L}_2, \cdots, \tilde{L}_n\}$ where $n < m$. For aggressive pruning, we focus on cases where $n \leq \frac{m}{2}$. Note that we may carry out the pruning algorithm on a subset of layers $\{L_a, L_{a+1}, \cdots, L_b\}$ where $[a, b] \subseteq [1, m]$, and inherit the layers $\{\tilde{L}_1, \cdots, \tilde{L}_{a-1}\}$ and $\{\tilde{L}_{b+1}, \cdots, \tilde{L}_m\}$ if either of them is non-empty.

In addition, for all our methods, we determine the output decoder layers based on the parameters of a fixed module type of the original decoder layers. The set of module types include T =['self_attn.k_proj', 'self_attn.v_proj', 'self_attn.q_proj', 'self_attn.o_proj', 'mlp.gate_proj', 'mlp.up_proj', 'mlp.down_proj', 'input_layernorm', 'post_attention_layernorm']. We denote the parameters of module type $t \in T$ in layer L_i by W_i^t.

`LayerCluster`: layer selection via k-medoids clustering. For a selected range $[a, b]$ and module type t, we perform k-medoids clustering on the parameters $[W_a^t, W_{a+1}^t, \cdots, W_b^t]$, yielding k cluster centers that correspond to the indices $\{j_1, \cdots, j_k\} \subseteq \{a, a+1, \cdots, b\}$. The resulting pruned model will sequentially inherit layers $\{L_1, \cdots, L_{a-1}, L_{j_1}, \cdots, L_{j_k}, L_{b+1}, \cdots, L_m\}$ of the original model.

`LayerMerge`: layer merging via Principal Component Analysis. For a selected range $[a, b]$ and each module type t, we perform Principal Component Analysis (PCA) on the stacked parameters $[W_a^t, W_{a+1}^t, \cdots, W_b^t]$. We pick the first k principal components for each module type to form k layers $\tilde{L}_{j_1}, \cdots, \tilde{L}_{j_k}$ that replace the original set of layers, where the parameters of each module type for $\tilde{L}_{j_r}$ correspond to the r-th principal component. The resulting pruned model will consist sequentially of layers $\{L_1, \cdots, L_{a-1}, \tilde{L}_{j_1}, \cdots, \tilde{L}_{j_k}, L_{b+1}, \cdots, L_m\}$.

`LayerMergeAct`: layer merging via adaptive grouping and PCA. Unlike the previous methods, which prune the LLM based solely on its parameters, `LayerMergeAct` leverages a small set of calibration examples $\mathcal{D}$ (128 random samples from the C4 dataset [7] training split, ODC-By license). As shown in Figure 1, for each example $x \in \mathcal{D}$, we perform a single forward pass through the original LLM and record the output activations of each layer $\{z_1^x, z_2^x, \cdots, z_m^x\}$. We then compute the

cosine similarity between each pair of outputs (z_i^x, z_j^x) and organize them into a cosine similarity matrix $S(x)$. We then average the similarity matrix across all samples in $\mathcal{D}$ to obtain the aggregated similarity matrix $S = \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} S(x)$. Using a user-defined threshold γ, we recursively find diagonal blocks of S where (i) all values are above γ and (ii) the minimum value in the block is largest across all available blocks that satisfy (i). More details can be found in Algorithm 1 in the Appendix.

After retrieving the block indices $\mathcal{I} = \{(s_1, e_1), (s_2, e_2), \cdots, (s_r, e_r)\}$, we perform the `LayerMerge` algorithm described in Section 3 for each pair of block indices in $\mathcal{I}$. Note that we slightly modify the algorithm by adding a global hyperparameter λ, where we keep the principal components that retain $\lambda\%$ variance for each pair of block indices instead of keeping a fixed number of principal components. The hyperparameters γ and λ determine the sparsity of the pruned LLM.

4 Experiments

To evaluate how scaling down LLMs along their depth affects key emergent abilities, we utilize the `Open LLM Leaderboard v2` (implemented in the Language Model Evaluation Harness [8], MIT license), which evaluates reasoning abilities in multiple aspects, including *mathematics and logic*, *multistep soft reasoning*, *natural language understanding*, and *graduate level domain knowledge*, etc. To facilitate rapid evaluation, we curate `Leaderboard-Lite`, a subset of 15 `Open LLM Leaderboard v2` challenges (Table 3) with the highest variance in performance when running the `LayerMergeAct` pruning algorithm under 4 different configurations that yield 50% sparsity on the `llama-2-7b-hf` base model (i.e., reduced to half of its original depth).

We compare our simple pruning methods with more complex state-of-the-art width pruning (`Wanda` [2] and `SparseGPT` [3]) and depth pruning methods (`ShortGPT` [4], `Shortened Llama` [5], and `LaCo` [6]) using `llama-2-7b-hf` and `llama-3.1-8b` as base models and a single NVIDIA Tesla V100-SXM2-32GB GPU.

Accuracy / Sparsity / Method	0.5 k=16	0.53 k=15	0.5625 k=14	0.625 k=12	0.7188 k=9	0.7813 k=7	0.8438 k=5	0.9062 k=3
`Wanda`	0.3126	0.3098	0.3101	**0.3091**	**0.3173**	**0.3121**	0.2751	0.3088
`SparseGPT`	**0.3195**	**0.3205**	**0.3155**	0.3036	0.2974	0.2907	**0.3019**	**0.3101**
`ShortGPT`	<u>0.2920</u>	0.3029	**0.3118**	**0.3088**	**0.3064**	0.2922	0.2949	<u>0.2999</u>
`Shortened Llama`	**0.2967**	0.2994	0.2915	<u>0.3014</u>	0.2793	0.2915	<u>0.3105</u>	<u>0.2999</u>
`LaCo`	0.2900	<u>0.3061</u>	<u>0.3108</u>	0.2997	<u>0.3004</u>	0.2850	0.2888	0.2798
`LayerCluster`	0.2793	0.2907	0.2689	0.2843	0.2870	0.2728	0.2822	0.2711
`LayerMerge`	0.2768	0.2793	0.2810	0.2885	0.2987	**0.3073**	**0.3113**	0.2967
`LayerMergeAct`	0.2897	**0.3091**	0.2987	<u>0.3014</u>	0.2999	<u>0.2924</u>	0.2987	**0.3062**

Table 1: The performance of pruning methods on `Leaderboard-Lite` with different LLM sparsity, using `llama-2-7b-hf` as the base model (accuracy 0.3152). "k" denotes the number of decoder layers of the depth-pruned model corresponding to the given sparsity. For each listed sparsity, the performance of the best-performing depth pruning method is marked in **bold** while the second-best is <u>underlined</u>. The best-performing width pruning method is also marked in **bold**.

Accuracy / Sparsity / Method	0.5 k=16	0.53 k=15	0.5625 k=14	0.625 k=12	0.7188 k=9	0.7813 k=7	0.8438 k=5	0.9062 k=3
`Wanda`	0.3158	0.3168	**0.3203**	**0.3185**	0.3029	**0.3021**	**0.3143**	0.2843
`SparseGPT`	**0.3277**	**0.3230**	0.3061	0.3106	**0.3051**	0.2964	0.2987	**0.3131**
`ShortGPT`	0.2838	<u>0.2992</u>	<u>0.2982</u>	**0.3091**	0.2934	**0.3049**	0.2894	**0.3004**
`Shortened Llama`	0.2927	0.2937	0.2885	0.2758	0.2920	0.2833	<u>0.2919</u>	**0.3004**
`LaCo`	<u>0.3041</u>	0.2982	**0.3071**	<u>0.3021</u>	<u>0.2969</u>	<u>0.2967</u>	0.2885	0.2862
`LayerCluster`	0.2880	**0.2994**	0.2867	0.2880	0.2751	0.2860	0.2537	0.2602
`LayerMerge`	0.2835	0.2820	0.2817	0.2830	0.2872	0.2865	0.2897	<u>0.2984</u>
`LayerMergeAct`	**0.3071**	0.2574	0.2897	0.2731	**0.3054**	0.2922	**0.2974**	0.2825

Table 2: The performance of pruning methods on `Leaderboard-Lite` with different sparsity of the pruned LLM, using `llama-3.1-8b` as the base model, which achieves an accuracy of 0.3850.

As shown in Tables 1 and 2, although simpler, our pruning methods achieve comparable and sometimes superior `Leaderboard-Lite` performance to the baselines: In particular, for the `llama-2-7b-hf` base model, our methods secured the first place among all depth pruning methods on the three largest sparsities (k = 3, 5, 7) and achieved performance comparable to that of the

base model. However, for both base models, the performance of depth pruning methods generally lags behind that of width pruning methods.

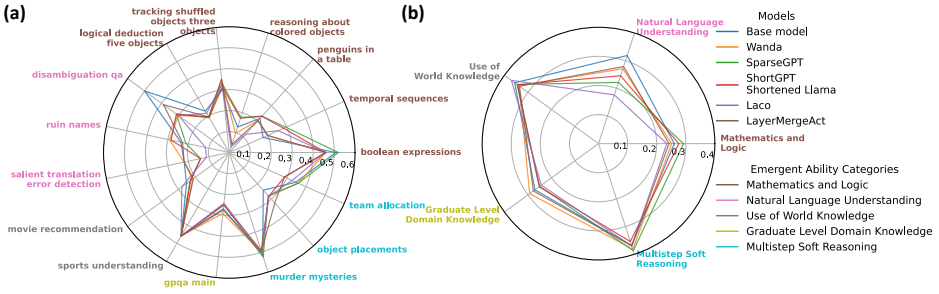

Figure 2: The performance of different pruning methods on `Leaderboard-Lite` (`llama-2-7b-hf` as the base model) with the largest model sparsity (k=3). **(a)** presents task-wise performance while **(b)** aggregates task performance by their corresponding category of emergent abilities. The layer selection methods `ShortGPT` and `Shortened Llama` resulted in identical models.

We then examine how different emergent abilities are affected when we aggressively scale down the model depth to only 3 layers. We first find that pruning strategies affect emergent abilities differently (Figure 2): While both width pruning methods outperform the depth pruning methods and the base model on *multistep soft reasoning*, `Wanda` excels in *graduate level domain knowledge* and `SparseGPT` leads in *mathematics and logic*. For depth pruning, layer selection methods (`Shortgpt` and `Shortened Llama`) slightly outperform the base model in terms of *mathematics and logic*. On the other hand, our layer merging approach `LayerMergeAct`, achieving the best overall `Leaderboard-Lite` performance (Table 1) among depth pruning methods, is the most successful in retaining emergent abilities in *graduate level domain knowledge* and *multistep soft reasoning*. Lastly, while all methods incur a noticeable drop in *natural language understanding* abilities, with the largest loss in the 'disambiguation qa' challenge, `LayerMergeAct` is the most successful in retaining this emergent ability.

We also find that there is no significant trend between depth and how well each emergent ability is retained (Figure 3). For `LayerMergeAct`, a depth of 3 layers achieves superior or comparable performance to all deeper configurations, suggesting that aggressively scaling down depth to the extreme can be viable for deployment with both resource and performance requirements.

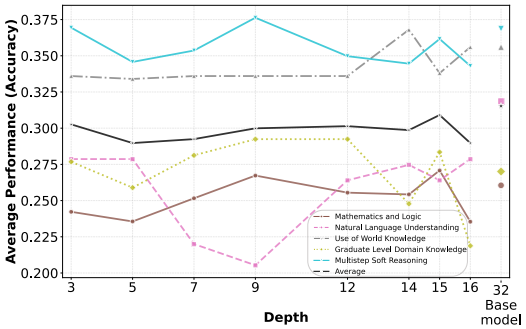

Figure 3: Emergent abilities of `LayerMergeAct`-pruned models with different depth (using `llama-2-7b-hf`).

5 Discussion

In this paper, we presented simple and direct pruning methods that reduce LLM complexities for deployment and conducted a unified and systematic evaluation of state-of-the-art pruning methods. Our evaluation showed that our methods, while aggressively shrinking the size of the model, can effectively retain key emergent abilities that empower LLMs, suggesting a practical pathway for managing the deployment of LLMs and their evolving lifecycle in general.

Although our methods achieve comparable or superior performance to the baselines, all methods suffer from a larger gap from the base model when using `llama-3.1-8b` (Table 2), while the performance of the pruned model under the same sparsity does not improve. This suggests that `llama-3.1-8b` may be less "prunable" than `llama-2-7b-hf`. As more advanced LLMs are being developed, it is interesting that they may be harder to scale down for deployment.

Furthermore, while all the evaluated methods lack a trend between performance and depth retention beyond 50% pruning (Tables 1 and 2, Figure 3), this phenomenon may also be due to the fact that task-agnostic pruning methods may be unable to retain specific emerging capabilities. In future work,

we will develop targeted, task-specific methods for scaling down LLMs and expect to gain more compression at a lower loss of performance. We will also develop more systematic benchmarks that evaluate emergent abilities at finer resolutions to carefully guide the development of such methods.

References

[1] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

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

[3] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR, 2023.

[4] Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*, 2024.

[5] Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. Shortened llama: A simple depth pruning for large language models. *arXiv preprint arXiv:2402.02834*, 11:1, 2024.

[6] Yifei Yang, Zouying Cao, and Hai Zhao. Laco: Large language model pruning via layer collapse. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6401–6417, 2024.

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[8] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

A Algorithm for finding diagonal blocks

Algorithm 1 Recursive Block Finder

Input:
Raw similarity matrix $S \in \mathbb{R}^{n \times n}$
Binarized matrix M where $M_{ij} = 1$ if $S_{ij} \geq \gamma$ and 0 otherwise, for a threshold γ.
Output:
A list of non-overlapping block indices $\mathcal{I} = [(s_1, e_1), (s_2, e_2), \dots]$

```
procedure FINDBLOCKS(S, M)
    n ← number of rows in M
    if n = 0 then return []
    end if
    if n = 1 then return [(0, 0)]
    end if
    B ← []                                   ▷ List to store potential blocks and their scores
    for s ← 0 to n − 1 do
        if M[s, s] = 1 then
            e ← s + 1
            while e < n and submatrix M[s : e, s : e] is all ones do
                e ← e + 1
            end while
            s_min ← min(S[s : e − 1, s : e − 1])
            Append ((s, e − 1), s_min) to B
        end if
    end for
    Sort B descending by block size (e − s), then by s_min
    (s*, e*) ← indices of the first block in sorted list B
    M_before ← M[0 : s* − 1, 0 : s* − 1]
    M_after ← M[e* + 1 : n − 1, e* + 1 : n − 1]
    I_before ← FindBlocks(S, M_before)
    I_after ← FindBlocks(S, M_after)
    Shift all indices (i_s, i_e) in I_after by e* + 1
    return I_before ∪ [(s*, e*)] ∪ I_after
end procedure
```

B Challenges in `Leaderboard-Lite`

Domain	Challenge
Mathematics and Logic	leaderboard_bbh_boolean_expressions leaderboard_bbh_temporal_sequences leaderboard_bbh_penguins_in_a_table leaderboard_bbh_reasoning_about_colored_objects leaderboard_bbh_tracking_shuffled_objects_three_objects leaderboard_bbh_logical_deduction_five_objects
Natural Language Understanding	leaderboard_bbh_disambiguation_qa leaderboard_bbh_ruin_names leaderboard_bbh_salient_translation_error_detection
Use of World Knowledge	leaderboard_bbh_movie_recommendation leaderboard_bbh_sports_understanding
Graduate Level Domain Knowledge	leaderboard_gpqa_main
Multistep Soft Reasoning	leaderboard_musr_murder_mysteries leaderboard_musr_object_placements leaderboard_musr_team_allocation

Table 3: Challenges in Leaderboard-Lite grouped by the measured emergent ability of the LLM.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: They accurately reflect the paper's contributions and scope, i.e., providing simple depth pruning methods and evaluating the emergent abilities of pruned LLMs

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: They are presented in the Discussion section, where it is stated that the task-agnostic methods may not be able to retain specific emerging capabilities, whereas tas-specific methods should be developed to remedy this issue.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The code, algorithm, and the hardware setting to facilitate reproducibility are all presented in this paper

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with

the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provided a link to the code/data at the end of the Introduction section.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: They are all presented at the beginning of the Experiments section.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The results of evaluating the pruned LLMs on the `Leaderboard-Lite` are deterministic, without the need for error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: They are provided at the start of the Experiments section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: Every aspect of this research conform to the Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Though pruning facilitates LLM deployment, the paper presents foundational research that is not tied to any particular downstream application. We do not see any direct path towards negative applications of our task-agnostic pruning methods and evaluations.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper uses an established leaderboard and poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use the C4 dataset (ODC-By license) and the OpenLLM Leaderboard via LM Evaluation Harness (MIT license) as assets and properly credited the creators.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The assets are open-sourced in the URL link provided at the end of the Introduction section.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.