

Pruning Laws for Large Language Models

Ayan Sengupta^{1*}, Siddhant Chaudhary^{1*}, Tanmoy Chakraborty²

¹Department of Electrical Engineering, Indian Institute of Technology Delhi, India,

²Yardi School of Artificial Intelligence, Indian Institute of Technology Delhi, India
ayan.sengupta@ee.iitd.ac.in, urssidd@gmail.com, tanchak@iitd.ac.in

Abstract

Scaling up model parameters and training data consistently improves the performance of large language models (LLMs), but at the cost of rapidly growing memory and compute requirements, which makes deployment on resource-limited hardware infeasible. *Model pruning*, a widely used compression technique, reduces inference costs by removing redundant parameters. However, its impact on downstream performance remains unpredictable and is typically assessed only through costly empirical sweeps. To address this gap, we introduce *pruning laws* – simple and interpretable scaling relations that connect a pruned LLM’s post-pruning performance to its unpruned performance and pruning ratio. Across ten LLMs (1.3B–30B parameters), a 20B mixture-of-experts model, three pruning strategies (unstructured, width, and depth), and eight diverse tasks, we show that pruning laws achieve strong predictive accuracy (average extrapolation error < 7%), reliably quantify performance degradation, and identify critical pruning thresholds beyond which recovery is infeasible. Moreover, we demonstrate that the functional form transfers across dense and mixture-of-experts architectures, pruning methods, and unseen models in zero-shot and one-shot setups, with task- and method-specific coefficients that vary in interpretable ways. These results provide both researchers and practitioners with a principled framework to select pruning strategies, estimate safe pruning ratios without exhaustive tuning, and deploy LLMs efficiently under real-world compute and latency constraints.

1 Introduction

The rapid evolution of Large Language Models (LLMs) has been guided largely by *neural scaling laws* (Kaplan et al., 2020; Hoffmann et al., 2022), which predict that increasing model size and data volume consistently improves performance.

However, this pursuit of scale has led to prohibitive computational costs, making deployment on resource-limited hardware increasingly infeasible (Faiz et al., 2024). To address this, *model pruning* has emerged as a critical technique for compressing LLMs. Whether through unstructured weight removal (Frantar and Alistarh, 2023) or structured approaches like depth and width pruning (Yang et al., 2024; Ashkboos et al., 2024), pruning aims to eliminate redundancy without sacrificing capability. Yet, despite its widespread adoption, pruning remains an empirical art; there is currently no systematic framework to predict how a specific pruning strategy will impact downstream performance across different models and tasks.

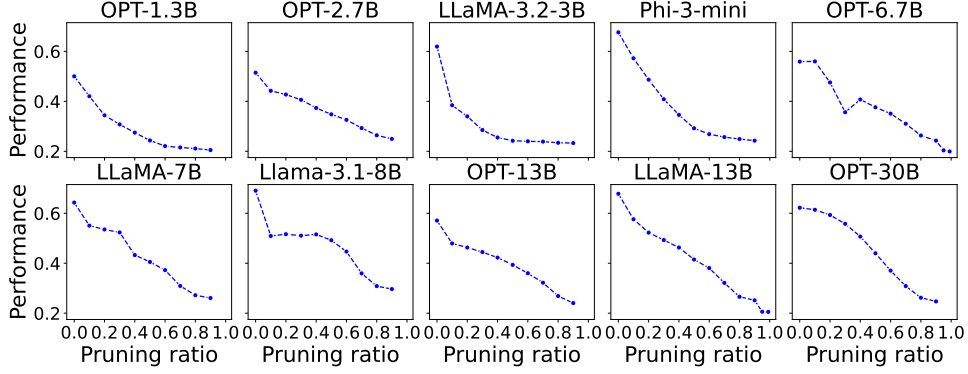
To bridge this gap, we introduce **Pruning Laws** – a principled analytical framework that models post-pruning performance as a function of the pruning ratio. By formalizing these relationships, we move beyond costly trial-and-error sweeps to answer four fundamental questions:

- **RQ1.** How does pruning affect downstream performance across task categories?
- **RQ2.** How do pruning strategies trade off between efficiency and accuracy?
- **RQ3.** Can pruning laws accurately predict performance for unseen models?
- **RQ4.** How can pruning laws guide principled pruning decisions?

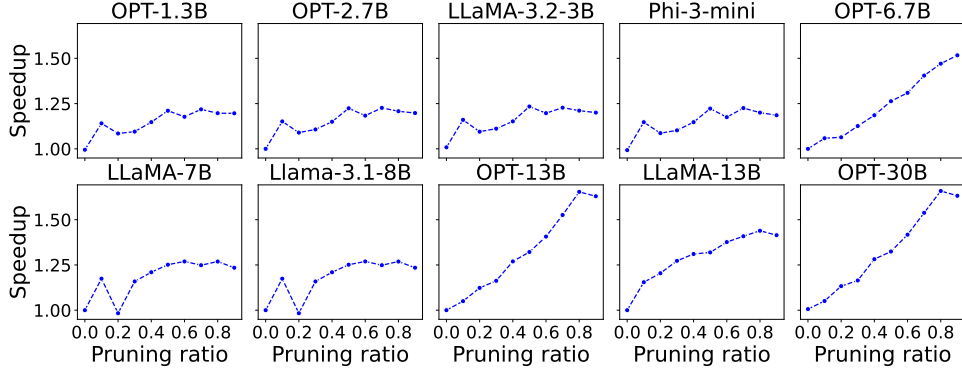
Our study fits these laws across a diverse landscape of ten LLMs (including OPT, LLaMA-2/3, and Phi-3), three distinct pruning strategies, and eight downstream tasks. We define the pruning law using a power function, $\mathcal{L} = \mathcal{L}_0 P_0 (1 - r)^\alpha$, which connects the pruned model’s performance $\mathcal{L}$ to its unpruned baseline $\mathcal{L}_0$ and the retention ratio $(1 - r)$ (with r being the pruning ratio). Our key empirical observations are summarized below:

RQ1. Impact on downstream performance. We find that performance degradation follows a

*Both authors contribute equally.



(a) Downstream model performance at different pruning ratios



(b) Inference speedup of pruned LLMs

Figure 1: **A.** Average downstream performance of different LLMs at varying pruning ratios. The different performance degradation patterns emphasize the need for a systematic evaluation of pruning effects on different LLMs of different scale and architectural family. Full result highlighted in Figure 5 of Appendix B. **B.** Inference speedup of different models at varying pruning ratios on an autoregressive language modeling benchmark.

predictable power law, but the sensitivity varies drastically by task. As shown in Figure 1a, *reasoning tasks* are remarkably resilient, characterized by a low decay exponent ($\alpha \approx 0.22$) that allows models to retain strong capabilities even at 50% pruning. In stark contrast, *QA tasks* are highly fragile ($\alpha \approx 2.42$), suffering sharp collapses at moderate compression ratios. Language modeling falls in between ($\alpha \approx 0.73$). Furthermore, we show that recovery fine-tuning (RFT) significantly mitigates this decay, reducing the sensitivity of language tasks from $0.73 \rightarrow 0.47$ and lowering test errors across the board.

RQ2. Computational benefits. Our analysis quantifies the "no free lunch" trade-off in pruning. *Depth pruning* offers the highest reward, achieving super-linear speedups ($> 5\times$ at 90% pruning) reflected by high scaling coefficients ($\alpha \approx 0.72$), though it suffers from higher variance in reliability. *Unstructured pruning* is the safest for accuracy but offers negligible speedups ($\sim 1.2\times$) due to its flat runtime scaling ($\alpha \approx 0.01$). *Width pruning*

occupies the middle ground, providing stable, predictable speedups of $\sim 1.4\times$ alongside moderate accuracy retention.

RQ3. Generalization to newer models. A critical feature of pruning laws is that the same functional form is reusable across setups. When applied zero-shot to unseen architectures like LLaMA-3.1 and Phi-3, our laws predict performance with low error (0.09–0.12). For out-of-distribution pruning methods, a simple one-shot calibration, requiring just a single evaluation, further refines these predictions (e.g., reducing SlimGPT prediction error from $0.13 \rightarrow 0.05$). The form also holds on a mixture-of-experts model, GPT-OSS-20B, where removing a routed block restructures the active-parameter graph rather than narrowing a dense stack, yielding coefficients ($\alpha = 0.48$, $P_0 = 0.89$) inside the range spanned by the dense models. What transfers is the form; the coefficients themselves remain task- and method-specific.

RQ4. Practical implications. These laws provide actionable guidelines for practitioners. Instead of guessing, one can use our coefficients to plan safe pruning thresholds – aggressive ratios (60–70%) for reasoning-heavy applications, but conservative ones ($< 40\%$) for QA. This framework transforms pruning from an ad-hoc procedure into a predictable engineering discipline.¹

2 Related Work

Model Pruning for Efficiency. Large Language Models have demonstrated remarkable capabilities, yet their deployment is often bottlenecked by immense computational requirements. Model pruning addresses this by reducing parameter counts, generally falling into two categories. *Unstructured pruning* removes individual weights (Frantar and Alistarh, 2023; Sun et al., 2023); while effective at preserving accuracy, it often requires specialized hardware to realize speedups. In contrast, *structured pruning* removes entire components such as layers or channels (Ashkboos et al., 2024; Yang et al., 2024), offering direct latency gains on standard hardware. Recent work has also explored calibration-free methods (Sengupta et al., 2025) to simplify the compression pipeline. However, despite the variety of methods, practitioners currently lack a unified framework to predict how these strategies will impact performance without running exhaustive benchmarks.

Neural Scaling Laws. The study of scaling laws has traditionally focused on pre-training. Foundational works like *Kaplan scaling laws* (Kaplan et al., 2020) and *Chinchilla scaling laws* (Hoffmann et al., 2022) established power-law relationships linking test loss to model size and data volume. Later studies have nuanced these findings, with Caballero et al. (2023) noting that monotonic laws may fail to capture emergent behaviors in Transformers. More recently, research has pivoted toward *parameter-efficient scaling*. Busbridge et al. (2025) introduced distillation scaling laws to optimize compute allocation between teacher and student models. Similarly, Kumar et al. (2024) proposed precision-aware scaling laws, demonstrating that low-precision training can improve scaling efficiency. Chen et al. (2024) examined scaling in the context of recovery fine-tuning for structured

pruning. While these works offer valuable insights into training and recovery dynamics, they do not address the fundamental degradation characteristics of the pruning process itself.

Positioning Pruning Laws. Our work introduces a distinct framework that complements existing compression theories. Unlike distillation scaling laws, which depend on specific teacher-student pairings and extensive retraining, or quantization scaling laws (Cao et al., 2024), which are often tightly coupled to hardware backends, our **Pruning Laws** are architecture-agnostic and require no retraining. By modeling performance as a direct function of the pruning ratio, our framework allows practitioners to identify *critical pruning thresholds* – the point beyond which recovery becomes infeasible. This capability generalizes across diverse model families and pruning methods (including unseen ones), making it a robust tool for resource-aware deployment planning.

3 Methodology

3.1 Parametrization of the LLM pruning law

With pruning laws, we propose a series of analytical methods for estimating the performance of LLM post-pruning on a variety of downstream tasks. For all downstream tasks, we assume that the performance of a model on a task is captured by a metric, which is bounded (e.g., model accuracy); in other words, higher the performance of a model, the better it is. Building on prior studies (Kaplan et al., 2020; Hoffmann et al., 2022) that establish scaling laws for LLM pre-training, we formulate a relationship between the performance of a pruned model and its corresponding base model through a law defined by two key parameters: the performance of the base model on a task (denoted by $\mathcal{L}_0$) and the pruning ratio used to prune the model (denoted by $r \in (0, 1)$); we denote the relationship by $\mathcal{L} := \mathcal{L}(\mathcal{L}_0, r)$, where $\mathcal{L}$ represents the performance of the pruned model. Our proposed pruning law can be used to determine the optimal value of r needed to obtain a well-performing pruned model, maximizing the performance retainment post-pruning.

The functional form of our parametrization is described by the following equation:

$$\mathcal{L}(\mathcal{L}_0, r) = \mathcal{L}_0 P_0 (1 - r)^\alpha \quad (1)$$

where α and P_0 are real numbers. The exponent α controls how quickly performance decays as the

¹Additional data, source code and dataset have been open-sourced at https://github.com/parmanu-lcs2/pruning_laws.

pruning ratio r increases. The coefficient P_0 is a *bias term* that encodes the boundary conditions of the law: although $(1 - r)$ tends to 1 as r approaches 0, the performance of the pruned model at extremely small pruning ratios may not exactly match the base model’s performance. Factors such as evaluation noise, subtle architecture-dependent effects and differences among pruning methods can introduce a systematic offset at $r \rightarrow 0$. Consequently, P_0 captures these offsets and depends on the particular model, pruning method and task being considered.

Levels of fit. In our analysis, we fit the pruning law at several levels of aggregation:

1. **Task level:** we fit a single (α, P_0) pair for each task, pooling data across all models and pruning methods. In this case, P_0 reflects the average boundary-condition bias for that task.
2. **Method–task level:** we fit the law separately for each pruning method and task across all models. Here P_0 absorbs both task-specific and method-specific biases (e.g., unstructured vs. structured pruning may yield different offsets near $r = 0$).
3. **Model–task level:** we fit the law for each model and task, pooling across methods. P_0 then captures architecture-specific bias for that task, indicating how resilient a given model is on that task when lightly pruned.

This choice of functional form is motivated by the following *feasibility conditions* for a pruning law:

- We formulate the pruning law as a power law with respect to the retention ratio. Unlike some pre-training scaling laws, we require the law to be *scale invariant* with respect to $(1 - r)$, i.e., $\mathcal{L}(\mathcal{L}_0, r)$ must be a homogeneous function of $(1 - r)$. Such a form allows us to derive optimal decision regions for choosing r given a target performance drop.
- The post-pruning performance should decrease as the pruning ratio r increases, i.e., $\frac{\partial \mathcal{L}}{\partial r} < 0$. In Equation 1, this behavior is captured by requiring $\alpha > 0$.
- The functional form should capture the *iterative nature* of many pruning methods: pruning a model by r_1 followed by further pruning by r_2 should yield a performance proportional to a single pruning with overall pruning ratio $1 - (1 - r_1)(1 - r_2)$ (Sengupta et al., 2025; Yang et al., 2024).

Here, we show that our proposed functional form in Equation 1 satisfies the last feasibility condition for iterative pruning methods. So, let $r_1 \in (0, 1)$ and $r_2 \in (0, 1)$ be two pruning ratios. Suppose a model is first pruned at ratio r_1 , followed by a pruning by the same method at ratio r_2 . It is easy to see that the overall pruning ratio (i.e., a pruning ratio achieving the total pruning in one step) is $r_1 + r_2 - r_1 r_2$. Then, the performance obtained by the multi-step pruning is

$$\begin{aligned} \mathcal{L}_{\text{mutlistep}} &= \mathcal{L}(\mathcal{L}(\mathcal{L}_0, r_1), r_2) \\ &= \mathcal{L}(P_0 \mathcal{L}_0 (1 - r_1)^\alpha, r_2) \\ &= P_0^2 \mathcal{L}_0 (1 - r_1)^\alpha (1 - r_2)^\alpha \\ &= P_0 \mathcal{L}(\mathcal{L}_0, r_1 + r_2 - r_1 r_2) \\ &= P_0 \mathcal{L}_{\text{singlestep}} \end{aligned}$$

Hence, our functional form attains proportional performance between single-step and multi-step pruning for iterative methods. Through extensive experiments (see Section 5), we find that the proposed form in Equation 1 satisfies remaining of the feasibility conditions.

3.2 Fitting the pruning law: Ordinary least squares for linear regression

Taking logarithms on both sides of Equation 1 gives:

$$\log \mathcal{L} = \log \mathcal{L}_0 + \alpha \log(1 - r) + \log P_0, \quad (2)$$

Therefore, fitting the pruning law reduces to a linear regression in log space. In this formulation, α is the slope (multiplying $\log(1 - r)$) and $\log P_0$ is the intercept, capturing the boundary-condition bias discussed above. The regression is performed on the transformed variables $r' := 1 - r$ with $\log P_0$ as the intercept. To learn α and P_0 , we use the standard ordinary least squares (OLS) method (Zdaniuk, 2014), assuming additive Gaussian noise. The regression model can thus be stated as:

$$\log \mathcal{L} = \log \mathcal{L}_0 + \alpha \log(1 - r) + \log P_0 + \epsilon_{\text{noise}}, \quad (3)$$

where $\epsilon_{\text{noise}} \sim \mathcal{N}(0, 1)$.²

4 Experimental Setup

To develop the proposed pruning laws, we compress and evaluate a range of LLMs, including

²A similar implementation with robust estimation Huber loss (Huber, 1992) is also performed, but due to very marginal differences omitted from this paper.

OPT (Zhang et al., 2022) at 1.3B, 2.7B, 6.7B, 13B and 30B parameter scales, and LLaMA-2 (Touvron et al., 2023b) at 7B and 13B. Moreover, for evaluating the universality of the pruning laws on more recent architectures, we consider LLaMA-3.1-8B, LLaMA-3.2-3B (Dubey et al., 2024) and Phi-3-mini-4K-Instruct (Abdin et al., 2024) models. All pre-trained model checkpoints are obtained from Hugging Face³. We experiment with three pruning methods: (1) **SparseGPT** (Frantar and Alistarh, 2023) for unstructured weight pruning, (2) **LaCo** (Yang et al., 2024) for structured depth pruning (layer collapse), and (3) **SliceGPT** (Ashkboos et al., 2024) for structured width pruning. Additional testing is conducted with two further depth pruning methods – **SlimGPT** (Ling et al., 2024) and **LLM Pruner** (Ma et al., 2023) and two width pruning methods including **SVD-LLM** (Wang et al., 2024) and **PruneNet** (Sengupta et al., 2025). To test the laws beyond dense decoder-only architectures, we further run a full depth pruning sweep on **GPT-OSS-20B** (Agarwal et al., 2025), an open-weights mixture-of-experts (MoE) model, using **ShortGPT** (Men et al., 2024), which ranks layers by a Block-Influence score. For each method, we apply a comprehensive range of sparsity levels with pruning ratios of $\{10\%, 20\%, \dots, 90\%\}$. We evaluate the pruned models on eight downstream tasks using the LM Evaluation Harness (Gao et al., 2024)⁴, performing all evaluations in a zero-shot setting. The tasks span three categories:

- **Reasoning:** PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2021), Hel-laSwag (Zellers et al., 2019), ARC-e and ARC-c (Clark et al., 2018).
- **Question-answering:** CoQA (Reddy et al., 2019)
- **Language modeling:** WikiText (Merity et al., 2017) and LAMBADA (Paperno et al., 2016).

For classification-style tasks, including reasoning and QA, we report *accuracy*. For language modeling tasks, we compute *perplexity*, which is unbounded $(1, \infty)$; thus, we report the inverse log-perplexity (i.e., $1/\log(\text{ppl})$) to map the metric to a $(0, 1)$ range for consistency with other evaluations. In addition to these individual tasks, we calculate the average task performance, denoted as ‘**Average**’.

³<https://huggingface.co/models>

⁴Task descriptions are provided in Appendix A.

Model	Reasoning	QA	Language	Average
OPT-1.3B	0.53	0.45	0.44	0.50
OPT-2.7B	0.52	0.51	0.49	0.51
LLaMA-3.2-3B	0.66	0.64	0.52	0.62
Phi-3-mini	0.74	0.68	0.53	0.68
OPT-6.7B	0.57	0.55	0.54	0.56
LLaMA-7B	0.64	0.64	0.64	0.64
LLaMA-3.1-8B	0.69	0.68	0.69	0.69
OPT-13B	0.58	0.56	0.56	0.57
LLaMA-13B	0.67	0.66	0.69	0.68
OPT-30B	0.64	0.56	0.60	0.62

Table 1: Results obtained by the unpruned models on different tasks.

Evaluating pruning laws. For evaluating the fitted pruning laws on in-distribution data points (e.g., same model/pruning method used in training and testing), we use *rolling-style* test data, where for each pruning ratio $r \in [20\%, 90\%]$, we train the pruning laws on pruning ratios $\{10\%, \dots, r\%\}$ and test the pruning laws on $\{r\% + 10\%, \dots, 90\%\}$. Finally, we calculate the average root mean square error (RMSE) between the predicted and ground truth over all the test datasets. For evaluating the extrapolation capability of the pruning laws in out-of-distribution setup, we adopt two extrapolation evaluation strategies – *zero-shot extrapolation*, where we use the fitted parametric functions to test on unseen data, and *one-shot extrapolation*, where we use the α coefficient from the fitted parametric function, but the bias term P_0 is re-estimated from pruning the model on a single pruning ratio. Precisely, let α and P_0 denote the fitted coefficients. For a given pruning ratio r , we prune the LLM, obtain its performance $\mathcal{L}_r$, and re-estimate the bias term $\hat{P}_0 = \frac{\mathcal{L}_r}{\mathcal{L}_0 \times (1-r)^\alpha}$. While zero-shot extrapolation setup is intended to verify the universality of the pruning laws, one-shot extrapolation is particularly useful for verifying the flexibility of pruning laws for testing out-of-distribution models or pruning strategies, when a single point enumeration is practically possible.

5 Experimental Results

5.1 Unpruned LLM Performance

Table 1 reports the accuracy of ten unpruned models across reasoning, QA, and language modeling tasks. As expected, performance generally improves with scale: OPT-2.7B achieves an average of 0.51, whereas OPT-13B and LLaMA-13B improve to 0.57 and 0.68, respectively. The LLaMA series consistently outperforms OPT at similar pa-

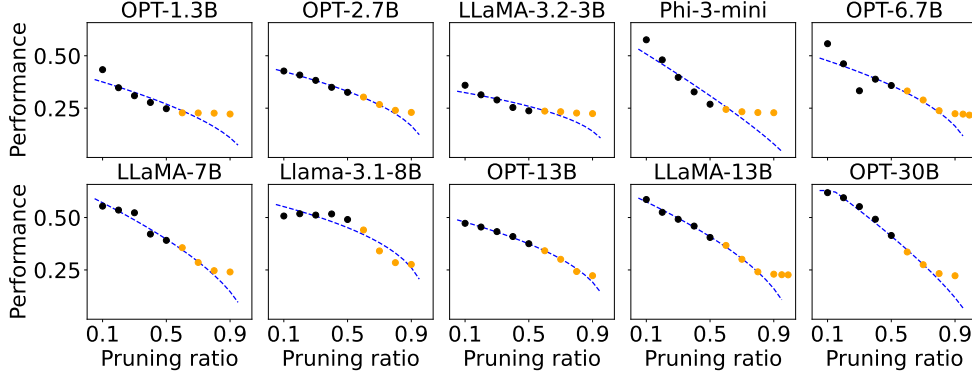


Figure 2: Fitted pruning laws for average downstream performance of pruned LLMs. Task-specific visualizations reported in Figure 6 of Appendix B. **Black** and **orange** points highlight the training and testing data points, respectively, with black line indicating the scores predicted by our pruning laws.

parameter scales, underscoring architectural and training advantages. Notably, Phi-3-Mini-4K-Instruct matches the average of LLaMA-13B (0.68) despite being smaller in size, and LLaMA-3.1-8B surpasses both (0.69). These results confirm that newer families, even at comparable or smaller parameter counts, can rival or outperform older architectures. These baseline numbers provide the $\mathcal{L}_0$ values against which pruning performance is evaluated.

5.2 Pruned LLM Results

Impact of pruning on model performance.

Pruning exerts a clear and systematic influence on downstream performance, with degradation patterns varying significantly across model scales and architectures. As illustrated in Figure 1a, larger models such as LLaMA-13B and OPT-30B exhibit notable resilience, maintaining competitive accuracy from high baselines (see Table 1) up to 30-40% pruning. In contrast, smaller but highly efficient models like LLaMA-3.2-3B and Phi-3-mini degrade much more rapidly. Despite LLaMA-3.2-3B achieving a high initial average score of 0.62, Figure 1a reveals a sharp performance drop even at low pruning ratios ($< 20\%$), confirming that while larger models offer greater tolerance due to parameter redundancy (Frantar and Alistarh, 2023), highly optimized efficient architectures possess significantly less redundant capacity.

Regarding task sensitivity, reasoning capabilities tend to be more robust compared to QA or language modeling. Larger models like LLaMA-7B manage to retain relatively strong reasoning scores even at 50% pruning, whereas QA and language modeling capabilities erode faster. However, regard-

less of model size, performance consistently collapses across all architectures once pruning ratios exceed 70-80%, with average scores approaching random baselines. These observations suggest that reasoning-oriented tasks degrade more gracefully than linguistic knowledge tasks, implying that pruning disproportionately impacts the retrieval of specific linguistic information over logical problem-solving abilities.

Pruning laws for downstream performance.

Table 2 demonstrates pruning laws that characterize how downstream performance scales with pruning ratios. Across tasks (cf. Table 2a), the exponent α is highest for QA ($\alpha \approx 2.42$) and smallest for reasoning ($\alpha \approx 0.22$). This indicates that QA accuracy decays sharply with pruning, whereas reasoning remains relatively stable even under high compression. Language tasks fall in between ($\alpha \approx 0.73$), showing gradual degradation.

The bias term P_0 , which modulates retained performance, is close to one for reasoning tasks ($P_0 \approx 0.95$), reflecting robustness. In contrast, it is substantially larger for QA ($P_0 \approx 1.45$), indicating disproportionately large performance drops. The deviation of P_0 from unity arises as a fitting artifact when a power law models “cliff-like” degradation curves: fragile tasks often maintain stability at low pruning ratios before collapsing sharply. To capture this steep degradation, the regression extrapolates a theoretical intercept ($P_0\mathcal{L}_0$) that exceeds the true baseline. Thus, $P_0 \gg 1$ signals a phase-transition-like behavior rather than super-compressibility. We also observe deviations at very small pruning ratios ($r < 10\%$), where empirical performance remains near $\mathcal{L}_0$ while the power law predicts slightly higher values. This indicates that

Task	α	P_0	R^2	F	Error
QA	2.42 ± 0.1	1.45 ± 0.1	0.89	348.5	0.04
Reasoning	0.22 ± 0.0	0.95 ± 0.0	0.79	170.8	0.06
Language	0.73 ± 0.0	0.76 ± 0.0	0.96	947.9	0.03
Average	0.35 ± 0.0	0.85 ± 0.0	0.88	316.4	0.05

(a) Task-level

Method	α	P_0	R^2	F	Error
Unstr	0.47 ± 0.0	1.22 ± 0.0	0.90	370.2	0.15
Depth	0.13 ± 0.0	0.54 ± 0.0	0.20	12.2	0.10
Width	0.39 ± 0.0	0.83 ± 0.0	0.70	103.1	0.10

(b) Method-task level

Model	α	P_0	R^2	F	Error
OPT-1.3B	0.26 ± 0.1	0.70 ± 0.1	0.57	11.5	0.10
OPT-2.7B	0.3 ± 0.0	0.83 ± 0.0	0.92	87.7	0.05
LLaMA-3.2-3B	0.18 ± 0.0	0.48 ± 0.1	0.56	11.3	0.08
Phi-3-mini	0.4 ± 0.1	0.64 ± 0.1	0.62	14.0	0.12
OPT-6.7B	0.19 ± 0.0	0.74 ± 0.1	0.65	19.7	0.11
LLaMA-7B	0.43 ± 0.0	0.85 ± 0.1	0.89	66.8	0.06
Llama-3.1-8B	0.35 ± 0.0	0.81 ± 0.0	0.88	59.8	0.09
OPT-13B	0.37 ± 0.0	0.86 ± 0.0	0.97	235.2	0.02
LLaMA-13B	0.23 ± 0.0	0.69 ± 0.1	0.70	24.1	0.11
OPT-30B	0.53 ± 0.1	0.99 ± 0.1	0.90	70.0	0.07

(c) Model-task level

Table 2: Coefficients of the (a) task-level, (b) method-task-level, and (c) model-task level pruning laws of the form $\mathcal{L} = \mathcal{L}_0 P_0 (1 - r)^\alpha$. We report the coefficients estimated with ordinary least square (OLS) fits, along with the standard errors. To evaluate the goodness-of-fit, we calculate adjusted R^2 , F statistic. We calculate the Error as the average root mean square error on the test data. Pruning laws with recovery fine-tuned LLMs are reported in Table 4 of Appendix B. We further highlight the results with other functional forms in Table 7, justifying the power-law functional form defined in Eq. 1.

the pruning law primarily characterizes the degradation regime rather than the stable low-sparsity regime.

Dependence of model architecture on effectiveness of pruning. Across model families (cf. Table 2c), the pruning law maintains strong predictive accuracy, though sensitivity varies with architecture and scale. Larger models such as LLaMA-13B and OPT-30B exhibit stable bias terms ($P_0 \approx 0.69$ – 0.99), indicating strong performance retention at the onset of pruning. In contrast, smaller efficient models like LLaMA-3.2-3B show lower bias terms ($P_0 \approx 0.48$), reflecting early performance collapse at low pruning ratios ($< 20\%$) before entering a smoother decay regime. While larger OPT models exhibit higher α values (≈ 0.53), this reflects decay from a higher baseline rather than increased fragility. The goodness-of-fit also correlates with

scale; larger models achieve higher adjusted R^2 scores (e.g., 0.97 for OPT-13B), confirming that the pruning law reliably captures architecture-specific behaviors. Overall, larger models offer stronger buffers against pruning by avoiding the catastrophic early drops observed in smaller architectures.

Does the pruning method matter? Method-level coefficients reported in Table 2b reveal clear differences among pruning strategies. Unstructured pruning shows the strongest adherence to the pruning law, with adjusted $R^2 = 0.90$ and a large F -statistic of 370.20. This strong fit likely arises from the fine-grained nature of weight removal, which produces a smooth reduction in model capacity that aligns well with power-law scaling. Depth pruning performs poorly in comparison, with adjusted $R^2 = 0.20$, a weak F -statistic of 12.21, and a low bias term ($P_0 \approx 0.54$). This behavior stems from the coarse-grained removal of entire layers, which introduces high variance depending on which layer is removed. Although the fit is weaker, empirical comparisons of functional forms (Table 7) confirm that the power law still yields lower test error (0.05) than linear or polynomial alternatives (0.06–0.10). Width pruning lies between the two, with moderate α (≈ 0.39), stable P_0 (≈ 0.83), and adjusted $R^2 = 0.70$. These results indicate that the pruning law generalizes across strategies while distinguishing their behavior.

The weak depth pruning fit is, however, a property of the *pruning operator* rather than of the framework. LaCo collapses layers according to a merge schedule that is discontinuous in r : the merge plan changes abruptly as r crosses block-boundary thresholds, so the realized degradation does not interpolate smoothly between adjacent ratios and a smooth power law absorbs the jaggedness as residual variance. Consistent with this reading, the one depth-pruning fit we obtain with a smoother operator – ShortGPT (Men et al., 2024), which ranks layers by a continuous Block-Influence score – reaches an average-task adjusted R^2 of 0.65 (Table 6), against 0.20 for LaCo. The two are not matched in scope – the ShortGPT fit covers one model, the LaCo row pools ten – so this is indicative rather than a controlled ablation. It nonetheless suggests that the fit quality of the law inherits the smoothness of the operator, which is a useful diagnostic when selecting a pruner.

Influence of model pruning on inference speed. Pruning substantially improves inference speedup,

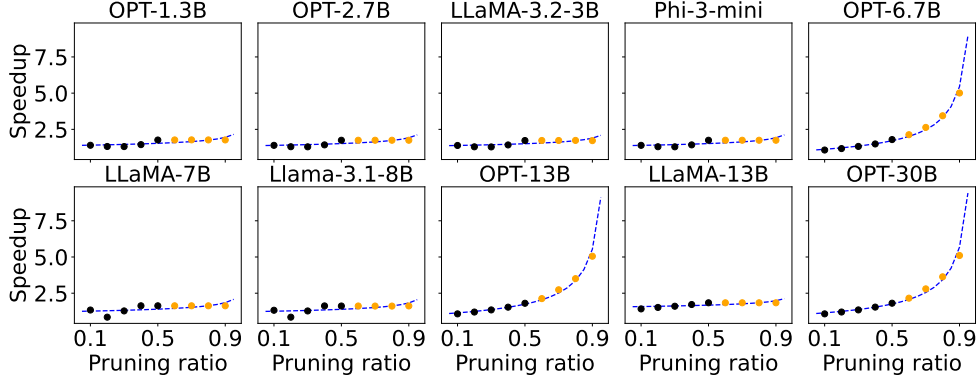


Figure 3: Fitted pruning laws for inference speedup of pruned LLMs for depth pruning. We report the remaining visualizations for other pruning methods in Figure 7 of Appendix B.

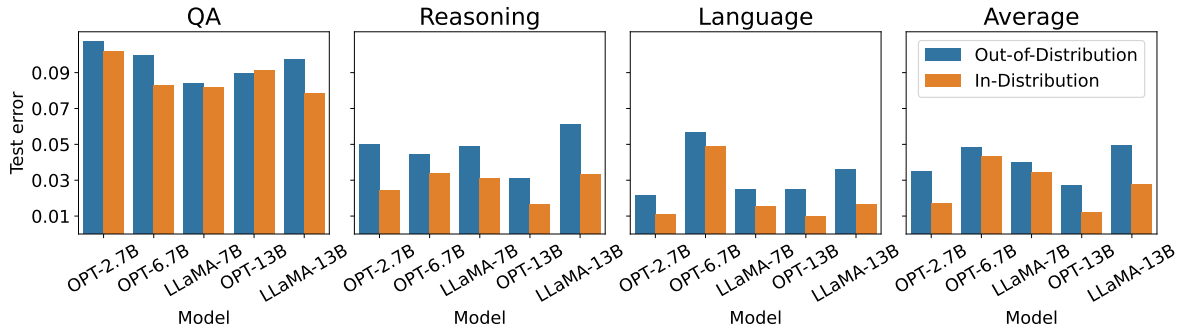


Figure 4: In-distribution (ID) and out-of-distribution (OOD) test error for different pruned LLMs. In the ID setup a model (e.g., LLaMA-7B) is tested while a different model from the same family (e.g., LLaMA-13B) is used in training. In the OOD setup, training and testing are performed on different model families.

though gains depend on pruning ratio and strategy (cf. Figure 3 and Table 5 of Appendix B). Depth pruning delivers the largest benefits, with super-linear speedups at higher pruning ratios. This behavior is reflected in relatively high α values (e.g., 0.72 ± 0.03 for OPT-13B and 0.73 ± 0.03 for OPT-30B). In contrast, unstructured pruning yields smaller but stable improvements, consistent with its near-zero $\alpha \approx 0.01$ (e.g., 0.01 ± 0.0 for LLaMA-7B and OPT-6.7B). Width pruning falls between the two, with $\alpha \approx 0.14$ – 0.15 for larger models (e.g., 0.15 ± 0.04 for LLaMA-13B). Goodness-of-fit metrics reinforce these trends: unstructured pruning shows strong adjusted R^2 values (0.94–0.97) and minimal test error (0–0.01), whereas depth pruning exhibits weaker fits and higher variance in test error (0.03–0.34). Overall, pruning laws confirm that depth pruning can yield dramatic inference acceleration, albeit with less stability, while unstructured and width pruning provide steadier improvements.

These wall-clock gains are hardware-conditional. Depth and width pruning produce dense smaller sub-networks and realize their gains directly on

standard accelerators. Unstructured pruning does not: fine-grained sparsity is not exploited by conventional dense kernels, and gains require N:M structured-sparse support (Mishra et al., 2021), so its near-flat $\alpha \approx 0.01$ reflects the backend we measure on rather than the sparsity itself. The two families of laws should therefore be read differently – the performance law of Eq. 1 is hardware-independent, whereas the speedup law is conditional on the deployment kernel.

5.3 Universality of Pruning Laws

Our results indicate that the same functional form $\mathcal{L}_1 = \mathcal{L}_0 (1 - r)^\alpha P_0$ transfers across *models*, *architectures*, and *pruning methods* with moderate testing error. Figure 4 clearly illustrates the consistently low in-distribution (ID) (when a model from same family is used in fitting the functions) test errors, as compared to out-of-distribution (OOD) (when none of the same-family models used in fitting) test errors. Particularly, for reasoning and language modeling tasks with larger models (13B sized), having same family models in the training

Method Type	Pruner	0-shot error	1-shot error
Depth	LLM Pruner	0.10	0.06
	SlimGPT	0.13	0.05
Width	SVD-LLM	0.13	0.16
	PruneNet	0.04	0.03

Table 3: OOD extrapolation results.

data (in-distribution) reduces the test error by 50%.

We further test universality by applying our fitted pruning laws to completely new pruning methods not used during training (Table 3). We obtain the performances of LLaMA-1-7B model (Touvron et al., 2023a) on reasoning tasks for depth pruning methods LLM Pruner (Ma et al., 2023) and SlimGPT (Ling et al., 2024) and two width pruning methods – SVD-LLM (Wang et al., 2024) and PruneNet (Sengupta et al., 2025), all numbers obtained from the corresponding papers. In the zero-shot setup, extrapolation errors remain small, ranging from 0.04 for PruneNet to 0.13 for SlimGPT and SVD-LLM. In the one-shot setup, where only $\hat{P}_0$ is recalibrated, errors decrease for most methods (e.g., LLM Pruner: 0.10 $\rightarrow$ 0.06, SlimGPT: 0.13 $\rightarrow$ 0.05, PruneNet: 0.04 $\rightarrow$ 0.03). Overall, these results reinforce the universality of pruning laws: they transfer across unseen pruning algorithms in zero-shot mode, while one-shot calibration may improve performance when the method aligns well with the retained elasticity.

Transfer to a mixture-of-experts model. All results so far concern dense decoder-only LLMs. To test whether the law survives a change of architecture, we fit it on GPT-OSS-20B, in which each block routes tokens to a subset of experts, so that removing a block restructures the active-parameter graph rather than narrowing a dense stack. Equation 1 applies without modification, yielding $\alpha = 0.48 \pm 0.13$ and $P_0 = 0.89 \pm 0.11$ on the average task – both inside the range spanned by the dense models – and the task ordering carries over (reasoning 0.30, language 0.98, QA 0.99; Table 6 of Appendix B). The sweep also separates two QA styles the main study could not: CoQA reproduces the cliff-like collapse discussed above ($\alpha = 3.53$, $P_0 \gg 1$), whereas CommonsenseQA decays gradually ($\alpha = 0.52$, $P_0 = 0.54$). The form covers both, with coefficients differing by an order of magnitude.

Practically, this means practitioners can *predict* pruned performance for new model families and

pruning strategies with zero data, and, if desired, achieve better estimation with a single-point calibration of P_0 . We stress that what transfers is the *functional form*, not a single set of coefficients: across the architectures, methods and tasks we study, $\mathcal{L} = \mathcal{L}_0 P_0 (1 - r)^\alpha$ remains the right two-parameter description, while (α, P_0) themselves remain task- and method-specific and vary in interpretable ways.

5.4 Practical Implications of Pruning Laws

Rather than relying on ad-hoc trial and error, our pruning laws provide a principled roadmap for model compression. Practitioners *seeking speed can turn to depth pruning*, those *prioritizing accuracy to unstructured pruning*, and those *balancing both to width pruning*. When *shifting to new models or pruning methods*, *zero-shot reuse of our coefficients enables immediate deployment*, while *one-shot calibration offers sharper estimates if a single evaluation is possible*. By aligning pruning choices with task sensitivity, *aggressive ratios for reasoning*, *conservative ones for QA and language modeling*, practitioners can achieve reliable efficiency gains without sacrificing performance. We provide more detailed practical guidelines in Appendix C for the readers to better synthesize our results for practical real-life deployment situations.

6 Conclusion

In this paper, we introduced pruning laws for LLMs that characterize how downstream performance scales with pruning ratio across tasks, architectures, and pruning strategies. Our analysis reveals systematic relationships between sparsity and performance, enabling predictive modeling of the accuracy–efficiency trade-off during model compression. Empirically, pruning sensitivity increases with model scale, while recovery tuning can mitigate performance degradation even with limited data. Among the evaluated methods, depth pruning yields the largest inference speedups, whereas unstructured pruning methods provide stronger robustness for downstream performance. Overall, our work establishes a quantitative foundation for understanding pruning behavior in LLMs and motivates future research on adaptive-hybrid compression strategies that balance computational efficiency with performance stability.

Limitations

While our proposed pruning laws provide a robust framework for predicting the performance of pruned LLMs, we acknowledge certain limitations that bound the scope of our findings. First, our empirical validation is conducted on models ranging from 1.3B to 30B parameters, together with a single 20B mixture-of-experts model. While we observe consistent scaling behaviors across this range, confirming these laws on significantly larger foundation models (e.g., 70B+ dense or MoE architectures) remains a subject for future study due to the high computational cost of such validations: a sweep at our resolution (9 ratios $\times$ multiple methods $\times$ 8 tasks) on a 70B+ model is beyond the compute budget available to us. Furthermore, regarding inference acceleration, our study highlights that unstructured pruning offers negligible speedups on standard hardware without specialized kernels. While the pruning laws accurately predict the *potential* theoretical efficiency, the realization of these gains in practice is heavily dependent on the underlying hardware and software stack.

When pruning laws fail. Our results also delineate the boundary conditions of the framework, which we consolidate here as four failure modes together with the mitigation we would recommend in each case.

1. **Discontinuous pruning operators.** When the pruning schedule jumps as r crosses structural thresholds, as in LaCo’s layer-merging plan, degradation does not interpolate smoothly between adjacent ratios and the residuals become jagged (adjusted $R^2 = 0.20$ for depth pruning in Table 2b). A smoother importance criterion appears to recover the fit – ShortGPT reaches an average-task adjusted R^2 of 0.65 on GPT-OSS-20B – so the law should be fitted per operator, or per layer-block segment when the operator is known to be discontinuous.
2. **Collapse regions at high pruning ratios.** Beyond $r \approx 70\text{--}80\%$ performance approaches the random baseline on most tasks, and a power law cannot describe a floor. Predictions in this regime should be treated as extrapolations only; in practice we recommend restricting the fit to ratios below the observed collapse onset.
3. **Cliff-like task degradation.** Tasks that stay flat and then collapse, such as CoQA, are fitted with an inflated intercept ($P_0 \gg 1$) and a

large exponent. This is a signature of a phase transition rather than of super-compressibility, and pooling such tasks with gradually decaying ones (e.g., CommonsenseQA, $\alpha = 0.52$) distorts both coefficients. Task-level coefficients should be reported separately when the task family is heterogeneous.

4. **Calibration for out-of-distribution methods.** Zero-shot transfer to an unseen pruner carries higher error (up to 0.13 in Table 3). One-shot re-estimation of P_0 narrows this gap for most methods but not all – it worsens SVD-LLM ($0.13 \rightarrow 0.16$) – so it helps only when the calibration ratio is representative of the deployment regime. Re-estimating α as well requires at least two evaluations at well-separated ratios, and is worthwhile only when that budget is available.

A further scope restriction concerns recovery fine-tuning. We fine-tune on WikiText-2 only, and report the resulting coefficients (Appendix B.2) as an illustration that recovery shifts α and P_0 in interpretable ways rather than as a theory of recovery. The scaling behavior of post-pruning recovery as a function of fine-tuning data and model size is treated directly by Chen et al. (2024), whose results are complementary to ours: our law characterizes the pre-recovery starting point, theirs the recovery trajectory.

Ethical Statement

The primary goal of this work is to advance the efficiency and accessibility of Large Language Models. By providing a principled method to predict post-pruning performance, our work contributes to the field of “Green AI,” enabling practitioners to reduce the energy consumption and carbon footprint associated with deploying LLMs. Furthermore, by identifying effective compression strategies, we aim to lower the hardware barrier for entry, allowing researchers and developers with limited computational resources to utilize capable models.

However, we acknowledge that the democratization of efficient, high-performance LLMs carries potential dual-use risks. The same techniques that make models more accessible for beneficial applications could also facilitate the deployment of malicious systems (e.g., for generating disinformation) on consumer-grade hardware. Additionally, while pruning removes redundant parameters, the impact of compression on the safety alignment and

bias of the model is not fully understood. It is possible that pruning could disproportionately affect safety guardrails or amplify latent biases present in the training data. We encourage practitioners to rigorously evaluate pruned models for safety and fairness before deployment in sensitive user-facing applications.

Acknowledgment

We acknowledge the support of the IBM-IITD AI Horizons network. T. Chakraborty would like to acknowledge the support of the Yardi School of Artificial Intelligence Travel Grant.

References

- Marah Abidin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. [Phi-3 technical report: A highly capable language model locally on your phone](#). *Preprint*, arXiv:2404.14219.
- Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). *arXiv preprint arXiv:2508.10925*.
- Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoeffler, and James Hensman. 2024. [SliceGPT: Compress large language models by deleting rows and columns](#). In *The Twelfth International Conference on Learning Representations*.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.
- Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. 2025. [Distillation scaling laws](#). In *Forty-second International Conference on Machine Learning*.
- Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. 2023. [Broken neural scaling laws](#). *arXiv preprint*. ArXiv:2210.14891.
- Zeyu Cao, Cheng Zhang, Pedro Gimenes, Jianqiao Lu, Jianyi Cheng, and Yiren Zhao. 2024. Scaling laws for mixed quantization in large language models. *arXiv preprint arXiv:2410.06722*.
- Xiaodong Chen, Yuxuan Hu, Jing Zhang, Xiaokang Zhang, Cuiping Li, and Hong Chen. 2024. Scaling law for post-training after model pruning. *arXiv preprint arXiv:2411.10272*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Ahmad Faiz, Sotaro Kaneda, Ruhan Wang, Rita Osi, Prateek Sharma, Fan Chen, and Lei Jiang. 2024. [Llmcarbon: modeling the end-to-end carbon footprint of large language models](#). *arXiv preprint*. ArXiv:2309.14393.
- Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.
- Elias Frantar and Dan Alistarh. 2023. [Sparsegpt: Massive language models can be accurately pruned in one-shot](#). *Preprint*, arXiv:2301.00774.
- 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, and 5 others. 2024. [A framework for few-shot language model evaluation](#).
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. [Training compute-optimal large language models](#). *arXiv preprint*. ArXiv:2203.15556.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Peter J. Huber. 1992. [Robust Estimation of a Location Parameter](#), pages 492–518. Springer New York, New York, NY.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *arXiv preprint*. ArXiv:2001.08361.
- Tanishq Kumar, Zachary Ankner, Benjamin F. Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. 2024. [Scaling laws for precision](#). *arXiv preprint*. ArXiv:2411.04330.

- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR’12, page 552–561. AAAI Press.
- Gui Ling, Ziyang Wang, and Qingwen Liu. 2024. Slimgpt: Layer-wise structured pruning for large language models. *Advances in Neural Information Processing Systems*, 37:107112–107137.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*.
- Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. [Shortgpt: Layers in large language models are more redundant than you expect](#). *Preprint*, arXiv:2403.03853.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. [Pointer sentinel mixture models](#). In *International Conference on Learning Representations*.
- Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378*.
- Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambda dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.
- Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Ayan Sengupta, Siddhant Chaudhary, and Tanmoy Chakraborty. 2025. [You only prune once: Designing calibration-free model compression with policy learning](#). In *The Thirteenth International Conference on Learning Representations*.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2024. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*.
- Yifei Yang, Zouying Cao, and Hai Zhao. 2024. [Laco: Large language model pruning via layer collapse](#). *Preprint*, arXiv:2402.11187.
- Bozena Zdaniuk. 2014. [Ordinary Least-Squares \(OLS\) Model](#), pages 4515–4517. Springer Netherlands, Dordrecht.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [Opt: Open pre-trained transformer language models](#). *Preprint*, arXiv:2205.01068.

A Dataset Descriptions

The WikiText corpus (Merity et al., 2017) is a standard benchmark for language modeling, consisting of human-curated articles that are stylistically polished, factually reliable, and written from a neutral perspective. Two variants exist – WikiText-2 and WikiText-103, with our experiments making use of WikiText-2. The LAMBADA dataset (*Language Modeling Broadened to Account for Discourse Aspects*) (Paperno et al., 2016) contains narrative passages designed to test a model’s ability to predict the final word, a task that requires integrating broad context and maintaining discourse-level coherence. PiQA (Bisk et al., 2020) evaluates physical common-sense reasoning in everyday situations, often with unconventional but practical solutions. Each instance poses an instruction-based problem (e.g., how to construct, bake, or manipulate objects) and provides two candidate solutions, of which exactly one is correct, framing the task as multiple-choice QA. WinoGrande (Sakaguchi et al., 2021), derived from the Winograd Schema Challenge (Levesque et al., 2012), is a large-scale benchmark for pronoun resolution, a task trivial for humans yet challenging for AI systems. HelLaSwag (Zellers et al., 2019) focuses on common-sense natural language inference: given a context, models must select the most plausible continuation. The AI2 Reasoning Challenge (Clark et al., 2018) provides grade-school science exam questions that require both knowledge and reasoning to solve. Finally, CoQA (Reddy et al., 2019) is a conversational QA dataset containing 127k question-answer pairs across 8k dialogues spanning seven domains. Its questions are conversational in style, and answers are free-form text supported by highlighted evidence in the passage, making it a benchmark for building dialogue-based reading comprehension systems.

B Results

In this section, we present the detailed task-level results for the pruned LLMs. These results complement the main paper by offering a fine-grained view of performance degradation under pruning, highlighting differences across architectures, pruning ratios, and downstream tasks.

B.1 Pruned LLM Performance

Figure 5 provides a comprehensive visual breakdown of downstream performance for all ten evalu-

ated LLMs across varying pruning ratios.

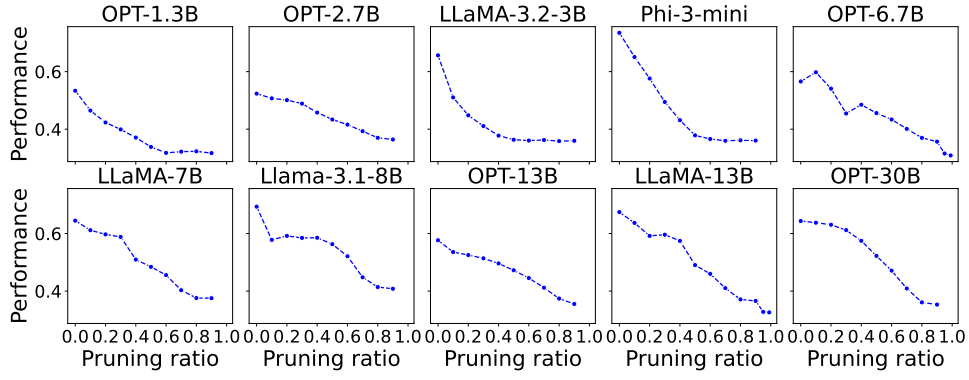
Reasoning tasks exhibit the most graceful degradation. Larger models like **OPT-30B** and **LLaMA-13B** maintain a remarkably flat performance trajectory up to 50-60% pruning before showing significant decline. Even smaller, highly efficient models like **Phi-3-mini** and **LLaMA-3.2-3B** start with high baselines (> 0.6) and decay relatively smoothly. This visual trend corresponds directly to the low decay exponents ($\alpha \approx 0.22$) reported in the main text, suggesting that logical reasoning relies on distributed representations that are robust to sparse removal.

In sharp contrast, the post-pruning results reveals a “cliff-like” failure mode for Question Answering. Almost every model, regardless of scale, suffers a catastrophic collapse once pruning ratios exceed 40-50%. For instance, **OPT-1.3B** and **LLaMA-3.2-3B** see their accuracy plummet to near-random levels almost immediately after the 20% pruning mark. This brittle behavior visually justifies the high bias terms ($P_0 > 1$) and steep decay coefficients ($\alpha > 2.0$) found in our scaling laws, indicating that the knowledge retrieval circuits required for QA lack the redundancy found in reasoning pathways.

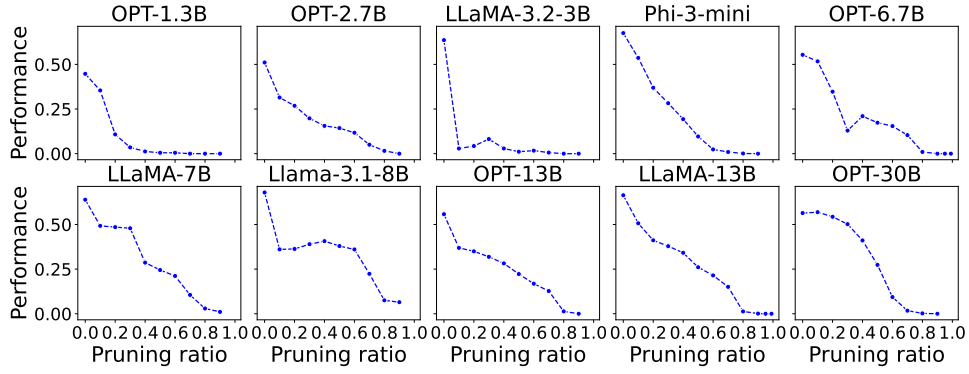
Language modeling occupies the middle ground. The curves are steeper than Reasoning but lack the sudden drop-offs of QA. We observe a quasi-linear decay in log-perplexity (mapped to normalized accuracy here) for models like **LLaMA-7B** and **LLaMA-3.1-8B**. This suggests that while linguistic coherence degrades predictably with parameter removal, it does not vanish as abruptly as factual recall.

Comparing the degradation curves across columns highlights the interaction between model scale and architectural efficiency:

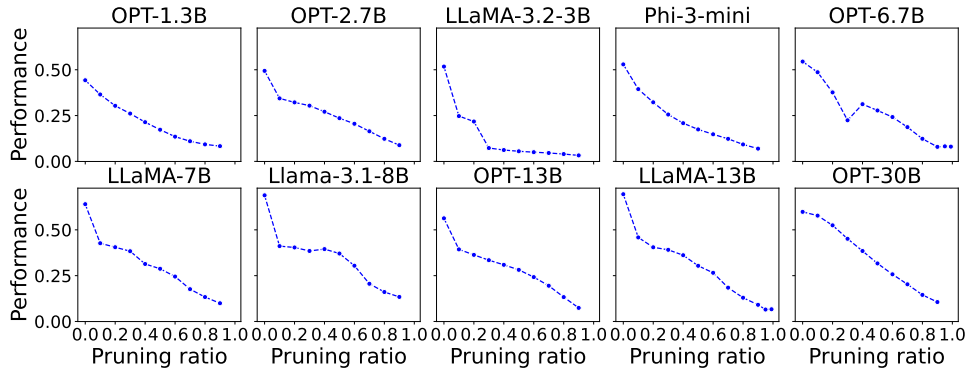
- **Redundancy in Scale:** Larger models (rightmost columns, e.g., OPT-30B) exhibit wider “safe zones” – plateaus where performance remains unchanged despite pruning. This empirically confirms the “Lottery Ticket” hypothesis (Frankle and Carbin, 2018) at scale, where massive models possess significant sparse subnetworks capable of solving the task.
- **Fragility of Efficient Models:** Newer, parameter-efficient models like **LLaMA-3.2-3B** and **Phi-3-mini** show high initial performance but steeper initial slopes compared to older, larger models like OPT-13B. This supports our



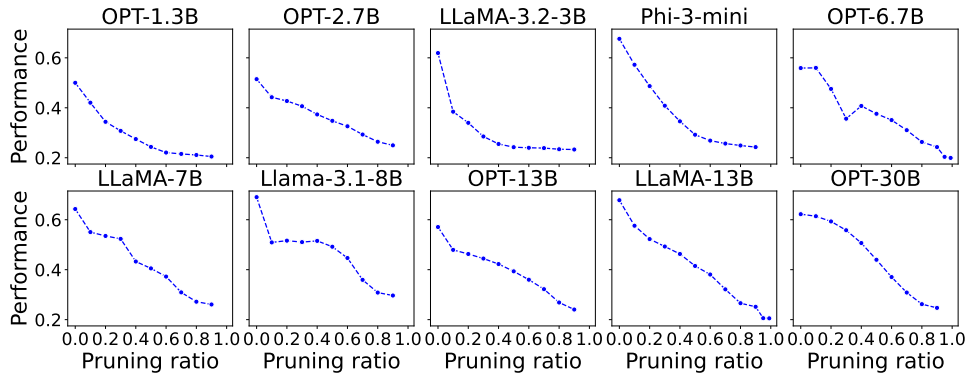
(a) Reasoning tasks



(b) QA tasks



(c) Language modeling tasks



(d) Average performance

Figure 5: Average downstream performance of different LLMs at varying pruning ratios for all tasks.

finding that highly optimized architectures have less “slack” or redundancy; they are efficient by design, leaving less room for pruning without immediate penalties.

B.2 Pruning Laws with Recovery Fine-Tuning

While our main analysis focuses on the direct impact of pruning (zero-shot), practitioners often apply recovery fine-tuning (RFT) to restore lost capabilities. Table 4 reports the pruning law coefficients fitted on models after they have undergone recovery fine-tuning on the WikiText-2 dataset.

Task-level robustness shifts. Comparing these RFT coefficients with the zero-shot coefficients in the main text (Table 2a), we observe distinct shifts in task sensitivity:

- **Language Modeling gains the most resilience.** The decay exponent α for language tasks drops significantly from ≈ 0.73 (zero-shot) to ≈ 0.47 (post-RFT). This indicates that fine-tuning is highly effective at repairing the specific probabilistic distributions required for next-token prediction, effectively “flattening” the degradation curve.
- **Hierarchy of sensitivity remains invariant.** Despite the improvements, the relative ordering of task fragility remains unchanged. QA tasks remain the most brittle ($\alpha \approx 2.39$, comparable to the zero-shot $\alpha \approx 2.42$), while Reasoning remains the most robust ($\alpha \approx 0.24$, similar to zero-shot $\alpha \approx 0.22$). This suggests that RFT helps recover general linguistic patterns (Language modeling) but struggles to recover the specific, “cliff-like” knowledge retrieval mechanisms required for QA once they are pruned.

Method-level consistency. At the method level (Table 4c), Unstructured Pruning continues to exhibit the best fit ($Adj R^2 \approx 0.88$), though its α coefficient decreases, reflecting improved retention. Interestingly, Depth Pruning sees a slight improvement in goodness-of-fit ($Adj R^2$ rises from $0.20 \rightarrow 0.29$), suggesting that fine-tuning smooths out some of the “step-wise” variance caused by removing discrete layers, though it remains the noisiest method to predict.

B.3 Pruning Laws on a Mixture-of-Experts Model

Table 6 reports the coefficients fitted on GPT-OSS-20B, pruned with ShortGPT at ratios $\{10\%, \dots, 80\%\}$ and evaluated on seven tasks

spanning all three task families. Two observations are worth isolating.

First, the fit is obtained without any change to Equation 1. This is not guaranteed a priori: in an MoE model each block routes tokens to a subset of experts, so removing a block both reduces depth and restructures which experts a token can reach, and one might expect the degradation curve to break from the dense case. It does not. The average-task coefficients ($\alpha = 0.48 \pm 0.13$, $P_0 = 0.89 \pm 0.11$) sit inside the range spanned by the dense models in Table 2c, and the task ordering is preserved.

Second, the inclusion of CommonsenseQA separates two degradation styles that were confounded in the main study, where CoQA was our only QA task. CoQA is fitted with $\alpha = 3.53$ and $P_0 = 38.86$, the signature of a curve that stays flat and then collapses; CommonsenseQA is fitted with $\alpha = 0.52$ and $P_0 = 0.54$, a gradual decay closer to the reasoning tasks. Both are described by the same form, but their coefficients differ by an order of magnitude, and the pooled QA group sits between them at an adjusted R^2 of 0.55. This supports reporting task-level rather than family-level coefficients whenever a task family is heterogeneous.

B.4 Justification of the Functional Form

A core contribution of our work is the specific power-law formulation $\mathcal{L} = \mathcal{L}_0 P_0 (1 - r)^\alpha$. To validate this choice, we compared it against seven alternative mathematical models, including standard power laws without bias terms, exponential decays, and rational functions. The results are summarized in Table 7.

Necessity of the Bias Term (P_0). The comparison highlights the critical role of the bias term P_0 . The standard power law without bias, $\mathcal{L} = \mathcal{L}_0 (1 - r)^\alpha$, yields a test error of 0.10, which is double the error of our proposed form (0.05). This confirms our hypothesis that pruning introduces an initial “offset” or phase transition (captured by P_0) that a rigid origin-bound power law cannot model accurately.

Superiority over Exponential and Polynomial forms. Our formulation also outperforms exponential decay models ($e^{-\beta r^\alpha}$), which tend to underpenalize aggressive pruning, resulting in higher test errors (0.06 – 0.09). Similarly, polynomial expansions ($1 - \alpha r - \beta r^2$) fail to capture the asymptotic

Task	α	P_0	Adj R^2	F Statistic	Test Error
QA	2.39 ± 0.11	1.32 ± 0.15	0.91	350.23	0.03
Reasoning	0.24 ± 0.02	0.9 ± 0.02	0.84	231.12	0.05
Language	0.47 ± 0.01	0.86 ± 0.02	0.96	1129.47	0.01
Average	0.27 ± 0.01	0.89 ± 0.02	0.89	351.41	0.03

(a) Task-level

Model	α	P_0	Adj R^2	F Statistic	Test Error
OPT-1.3B	0.25 ± 0.04	0.65 ± 0.03	0.65	24.19	0.06
OPT-2.7B	0.21 ± 0.01	0.89 ± 0.02	0.96	209.1	0.02
LLaMA-3.2-3B	0.21 ± 0.04	0.51 ± 0.02	0.61	37.03	0.04
Phi-3-mini	0.37 ± 0.06	0.69 ± 0.03	0.71	49.31	0.05
OPT-6.7B	0.28 ± 0.02	0.93 ± 0.03	0.95	139.87	0.05
LLaMA-7B	0.29 ± 0.03	0.87 ± 0.03	0.92	98.4	0.04
Llama-3.1-8B	0.35 ± 0.03	0.83 ± 0.04	0.89	71.08	0.05
OPT-13B	0.24 ± 0.01	0.88 ± 0.01	0.99	1289.02	0.01
LLaMA-13B	0.32 ± 0.03	0.88 ± 0.03	0.94	133.19	0.01
OPT-30B	0.52 ± 0.04	0.99 ± 0.02	0.94	92.11	0.05

(b) Model-task level

Method	α	P_0	Adj R^2	F Statistic	Test Error
Unstructured	0.34 ± 0.02	1.13 ± 0.02	0.88	309.55	0.10
DepthPruning	0.13 ± 0.03	0.65 ± 0.03	0.29	18.61	0.06
WidthPruning	0.31 ± 0.03	0.88 ± 0.03	0.76	141.88	0.06

(c) Method-task level

Table 4: Pruning laws of LLMs post-recovery-fine-tuning. All the models are fine-tuned on the WikiText2 dataset with a subsample size of 1000 and a LoRA (Hu et al., 2022) rank of 8.

Model	Method	α	P_0	Adj R^2	F Statistic	Test Error
OPT-1.3B	Depth	0.15 ± 0.05	0.72 ± 0.05	0.52	9.62	0.28
OPT-2.7B		0.14 ± 0.05	0.73 ± 0.05	0.51	9.27	0.2
LLaMA-3.2-3B		0.14 ± 0.05	0.73 ± 0.05	0.5	8.97	0.31
Phi-3-mini		0.14 ± 0.05	0.73 ± 0.05	0.51	9.27	0.31
OPT-6.7B		0.72 ± 0.03	0.96 ± 0.03	0.99	677.7	0.03
LLaMA-7B		0.17 ± 0.1	0.81 ± 0.11	0.21	3.14	0.26
LLaMA-3.1-8B		0.17 ± 0.1	0.81 ± 0.11	0.21	3.16	0.34
OPT-13B		0.72 ± 0.03	0.94 ± 0.03	0.99	589.92	0.04
LLaMA-13B		0.1 ± 0.04	0.64 ± 0.04	0.48	8.35	0.15
OPT-30B		0.73 ± 0.03	0.94 ± 0.04	0.98	478.49	0.04
OPT-1.3B	Unstr	0.01 ± 0.0	0.78 ± 0.0	0.33	4.9	0.02
OPT-2.7B		0.01 ± 0.0	0.78 ± 0.0	0.5	9.15	0.01
LLaMA-3.2-3B		0.01 ± 0.0	0.78 ± 0.0	0.5	9.15	0.02
Phi-3-mini		0.01 ± 0.0	0.78 ± 0.0	0.31	4.58	0.02
OPT-6.7B		0.01 ± 0.0	0.84 ± 0.0	0.96	176.74	0
LLaMA-7B		0.01 ± 0.0	0.81 ± 0.0	0.95	167.97	0
LLaMA-3.1-8B		0.01 ± 0.0	0.81 ± 0.0	0.95	167.97	0
OPT-13B		0.01 ± 0.0	0.82 ± 0.0	0.94	118.19	0.01
LLaMA-13B		0.02 ± 0.0	0.87 ± 0.0	0.97	294.22	0
OPT-30B		0.01 ± 0.0	0.82 ± 0.0	0.94	118.19	0.01
OPT-1.3B	Width	-0.0 ± 0.02	1.15 ± 0.02	-0.14	0.01	0.36
OPT-2.7B		-0.0 ± 0.02	1.15 ± 0.02	-0.14	0.01	0.08
LLaMA-3.2-3B		-0.0 ± 0.02	1.14 ± 0.02	-0.14	0.01	0.36
Phi-3-mini		-0.0 ± 0.02	1.15 ± 0.02	-0.14	0.01	0.36
OPT-6.7B		0.08 ± 0.02	1.08 ± 0.03	0.57	11.44	0.11
LLaMA-7B		0.02 ± 0.02	1.02 ± 0.02	-0.06	0.56	0.09
LLaMA-3.1-8B		0.01 ± 0.02	1.01 ± 0.02	-0.09	0.31	0.23
OPT-13B		0.14 ± 0.04	1.02 ± 0.05	0.55	10.58	0.19
LLaMA-13B		0.15 ± 0.04	0.95 ± 0.04	0.69	18.91	0.18
OPT-30B		0.14 ± 0.04	1.03 ± 0.05	0.55	10.91	0.21

Table 5: Coefficients of pruning law for inference speedup of the form $\mathcal{L} = \mathcal{L}_0 P_0 (1 - r)^\alpha$.

Task	α	P_0	R^2	F	Error
QA	0.99 ± 0.32	1.10 ± 0.27	0.55	9.7	0.09
Reasoning	0.30 ± 0.07	0.85 ± 0.06	0.70	17.6	0.12
Language	0.98 ± 0.14	0.79 ± 0.12	0.87	48.9	0.05
Average	0.48 ± 0.13	0.89 ± 0.11	0.65	14.0	0.13

(a) Task-group level

Task	α	P_0	R^2	F	Error
ARC-c	0.41 ± 0.17	0.69 ± 0.14	0.42	6.0	0.10
ARC-e	0.61 ± 0.11	0.87 ± 0.09	0.82	32.8	0.11
PIQA	0.20 ± 0.04	0.94 ± 0.03	0.81	31.5	0.10
WinoGrande	0.12 ± 0.04	0.86 ± 0.03	0.53	8.8	0.12
CoQA	3.53 ± 0.71	38.86 ± 1.67	0.77	24.8	0.05
CSQA	0.52 ± 0.24	0.54 ± 0.20	0.36	4.9	0.13
LAMBADA	0.98 ± 0.14	0.79 ± 0.12	0.87	48.9	0.05

(b) Task level

Table 6: Coefficients of the pruning law fitted on the mixture-of-experts model GPT-OSS-20B (Agarwal et al., 2025), pruned with ShortGPT (Men et al., 2024) at ratios $\{10\%, \dots, 80\%\}$. The sweep additionally covers CommonsenseQA (CSQA) (Talmor et al., 2019), widening the QA task family beyond conversational reading comprehension.

Functional Form	Train Error	Test Error
$L = L_0 * P_0 * (1 - r)^\alpha$ (Ours)	0.02	0.05
$L = L_0 * (1 - r)^\alpha$	0.05	0.10
$L = L_0 * (1 - \beta * r^\alpha)$	0.03	0.05
$L = L_0 * e^{-\beta * r^\alpha}$	0.03	0.06
$L = L_0 * (1 - \beta * (1 - e^{-\alpha * r}))$	0.03	0.09
$L = L_0 / (1 + \beta * r^\alpha)$	0.03	0.07
$L = L_0 / (1 + e^{\beta * (r - \alpha)})$	0.03	0.06
$L = L_0 * (1 - \alpha * r - \beta * r^2)$	0.03	0.06

Table 7: Results with pruning laws of other functional forms. Among all the functional forms investigated, our original functional form generates the lowest training and testing error, affirming the validity of the fitted pruning laws.

tail behavior of degradation as effectively. The proposed form achieves the lowest combined training error (0.02) and test error (0.05) among all candidates, balancing model complexity with predictive fidelity across the wide range of pruning ratios (10% – 90%).

C Practitioner Guidelines for Selecting Pruning Strategies

To make our findings actionable, we summarize them as practical rules of thumb for real-world adoption of pruning strategies. These guidelines allow practitioners to quickly decide which pruning strategy best matches their constraints and to use the pruning laws for principled planning rather than

ad hoc trial-and-error:

1. Selecting the pruning method:

- Use *depth pruning* when runtime speedup is the primary objective, as it yields the largest accelerations (up to $5\times$), though with higher variance in accuracy.
- Use *unstructured pruning* when performance preservation is critical, since it provides the lowest test errors and the most stable predictions, albeit with modest speedups.
- Use *width pruning* for a balanced trade-off between accuracy and speed, offering moderate but consistent improvements.

2. When facing unseen models:

- Start with *zero-shot mode* by directly reusing fitted coefficients (α , P_0), which achieves low extrapolation error even across families.
- If limited resources are available to probe the new model, perform *one-shot calibration*: keep α fixed and re-estimate only P_0 from a single pruning experiment. This is beneficial when accuracy at specific pruning ratios is critical.

3. When using new pruning methods:

- Begin with zero-shot predictions using coefficients from similar pruning categories.
- Apply one-shot calibration of P_0 if the method is highly different (e.g., algorithmic pruning heuristics not studied in our experiments) and one evaluation run is feasible.

4. When targeting different tasks:

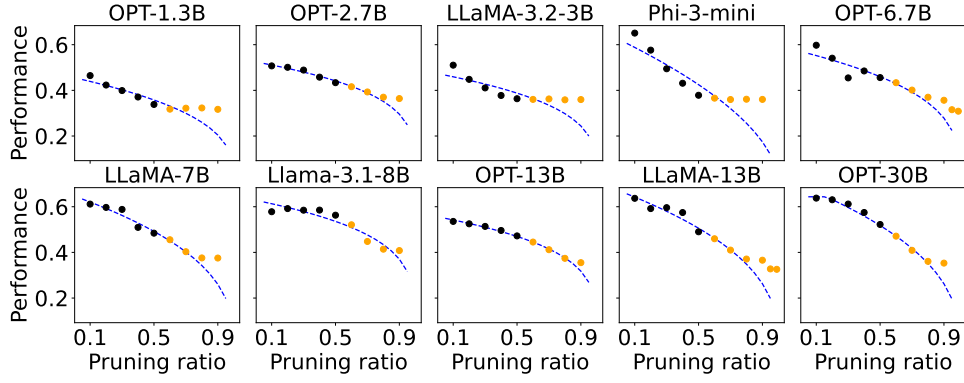
- For *reasoning tasks*, higher pruning ratios can be tolerated, as they are more robust (smaller α).
- For *QA and language modeling*, adopt more conservative pruning ratios, since these tasks degrade faster with pruning.

5. Compression planning:

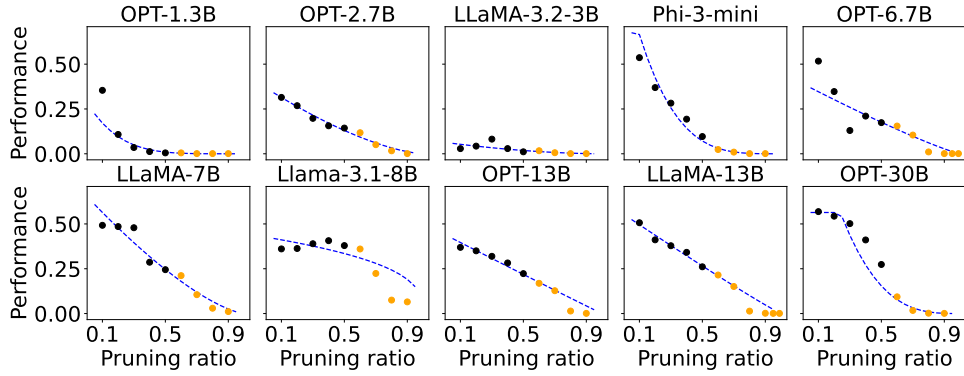
- If multiple evaluations are possible, test at a few pruning ratios (preferably $< 50\%$) and fit the pruning law locally to refine estimates.
- If evaluations are infeasible (e.g., resource-constrained scenarios), reuse our published coefficients (model-wise or task-wise) to approximate trade-offs with reasonable accuracy.

In summary, practitioners should use zero-shot pruning laws for fast deployment across new models, tasks, or methods, and fall back to one-shot

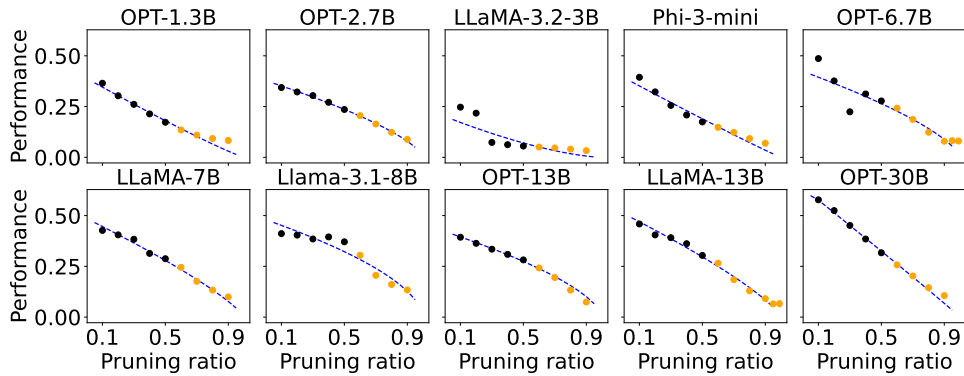
calibration of P_0 when some evaluation budget is available. This ensures a principled, universal, and resource-aware approach to model pruning.



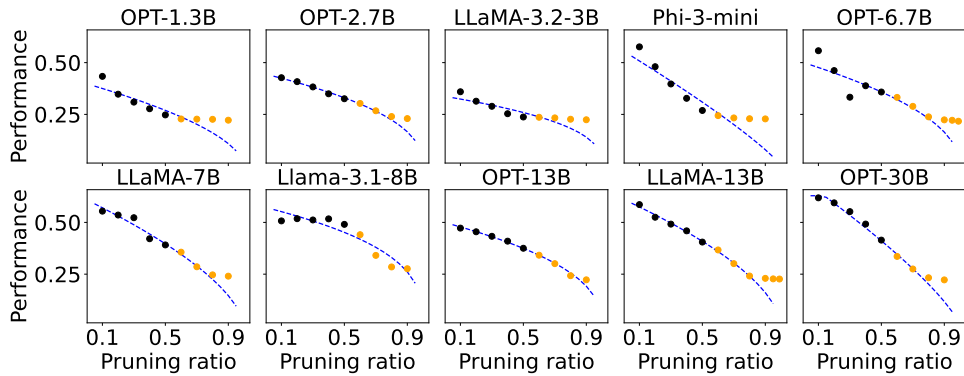
(a) Reasoning tasks



(b) QA tasks

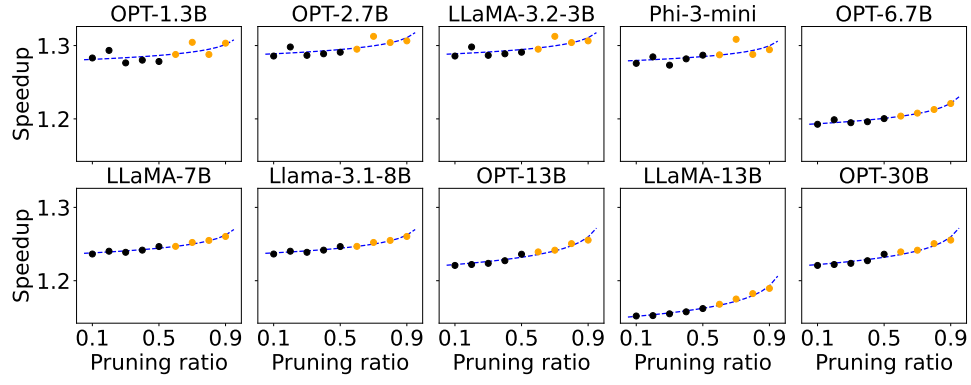


(c) Language modeling tasks

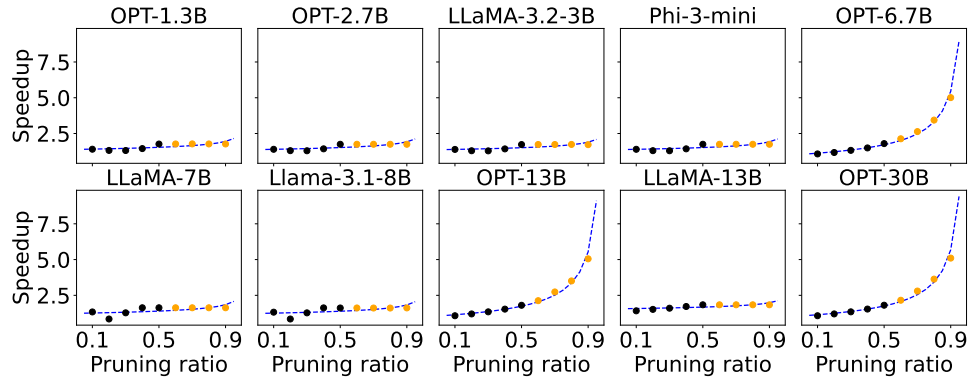


(d) Average task performance

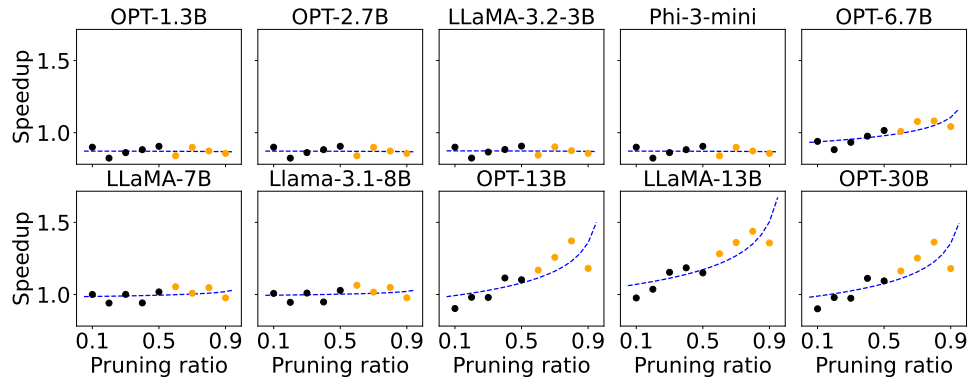
Figure 6: Fitted pruning laws for downstream performance of pruned LLMs. **Black** and **orange** points highlight the training and testing data points, respectively, with black line indicating the scores predicted by our pruning laws.



(a) Unstructured Pruning



(b) Depth Pruning



(c) Width Pruning

Figure 7: Fitted pruning laws for inference speedup of pruned LLMs for different pruning methods.