P^2 Law: Scaling Law for Post-Training After Model Pruning

Xiaodong Chen[2,3*], **Yuxuan Hu**[2,3*], **Xiaokang Zhang**[2,3], **Yanling Wang**[4]
Cuiping Li[1,2], **Hong Chen**[1,2], **Jing Zhang**[1,2†]
[1]Engineering Research Center of Database and Business Intelligence, MOE, China
[2]School of Information, Renmin University of China,Beijing, China
[3]Key Laboratory of Data Engineering and Knowledge Engineering, MOE, China
[4] Zhipu AI, China
{chenxiaodong,huyuxuan1999,zhang-jing,zhang2718,licuiping,chong}@ruc.edu.cn
wangyl@zgclab.edu.cn

Abstract

Pruning has become a widely adopted technique for reducing the hardware requirements of large language models (LLMs). To recover model performance after pruning, post-training is commonly employed to mitigate the resulting performance degradation. While post-training benefits from larger datasets, once the dataset size is already substantial, increasing the training data provides only limited performance gains. To balance post-training cost and model performance, it is necessary to explore the optimal amount of post-training data. Through extensive experiments on the Llama-3 and Qwen-2.5 series models, pruned using various common pruning methods, we uncover the scaling **Law** for **P**ost-training after model **P**runing, referred to as the P^2 Law. This law identifies four key factors for predicting the pruned model's post-training loss: the model size before pruning, the number of post-training tokens, the pruning rate, and the model's loss before pruning. Moreover, P^2 Law can generalize to larger dataset sizes, larger model sizes, and higher pruning rates, offering valuable insights for the post-training of pruned LLMs.

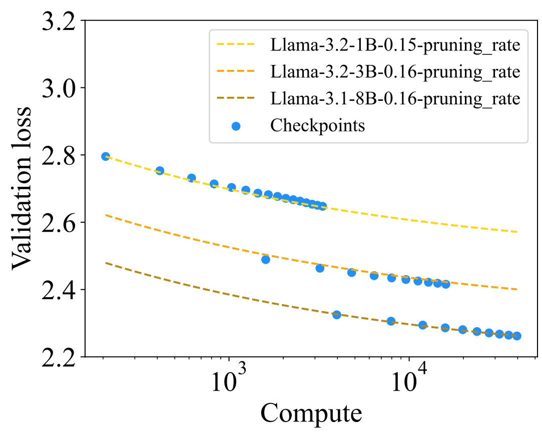


Figure 1: Loss curves derived by P^2 Law and the actual checkpoints of Llama-3 series models pruned by depth pruning with a pruning rate of approximately 15%. Compute (C) denotes the computational cost, which is calculated by $C = 6ND$ (Kaplan et al., 2020), where N denotes the model size after pruning, and D denotes the number of post-training tokens.

1 Introduction

Large language models (LLMs) based on the Transformer architecture (Vaswani et al., 2017) have been applied across diverse domains and tasks. However, as LLMs grow in size, their hardware demands increase substantially, limiting their practical deployment in real-world scenarios. To address this challenge, researchers have focused on developing compact models through model pruning techniques (Han et al., 2016) that maintain high performance while reducing hardware requirements.

Model pruning can be broadly categorized into unstructured pruning (Frantar and Alistarh, 2023; Zhang et al., 2024; Sun et al., 2024) and structured pruning (Chen et al., 2024; Hu et al., 2024; Liu et al., 2024; Muralidharan et al., 2024; Ma et al., 2023; Ashkboos et al., 2024; Men et al., 2024). Unstructured pruning removes individual elements from weight matrices, producing sparse matrices while preserving satisfactory model performance. However, the introduced structural irregularities make this approach hardware-unfriendly and hinder its ability to accelerate computation. To mitigate this problem, semi-structured pruning, a variant of unstructured pruning, leverages specific hardware support (Mishra et al., 2021) to achieve acceleration but may result in greater performance degradation compared to unstructured pruning. In contrast, structured pruning removes entire components, such as attention heads or layers, effectively reducing the model size but often with a higher performance drop compared to other pruning methods.

To effectively leverage hardware-friendly mod-

[*]Xiaodong Chen and Yuxuan Hu have equal contribution.
[†]Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5668–5686

els pruned using semi-structured or structured pruning methods, post-training (Ashkboos et al., 2024; Chen et al., 2024; Yang et al., 2024; Ma et al., 2023; Kim et al., 2024) serves as an essential step after model pruning to mitigate the performance degradation. For example, LLM-Pruner (Ma et al., 2023) utilizes 50,000 instruction data samples for fine-tuning, whereas Shortened Llama (Kim et al., 2024) uses 627B tokens of pre-training data for continual pre-training of the pruned LLMs. In general, compared to fine-tuning with a small dataset, continual pre-training with a large dataset is a more effective way to fully recover performance, but it demands substantial hardware resources. Given the significant hardware demands, a question is raised: **is it truly necessary to use a vast amount of data for performance recovery?** LLM-Streamline (Chen et al., 2024) answers the question by demonstrating that using large amounts of data for post-training only slightly improves performance compared to using a suitably sized amount. Hence, this raises another question: **whether a scaling law can be established to predict the optimal amount of post-training data required after model pruning for resource efficiency?**

To address the problem, we conduct pilot experiments on the Llama-3 (Dubey et al., 2024) and Qwen-2.5 series models (Team, 2024), applying both typical structured and semi-structured pruning methods. In specific, we observe several trends in the post-training loss curves, allowing us to identify the necessary conditions that the scaling **Law** for **P**ost-training after model **P**runing (P^2 Law) must satisfy. Building on the Chinchilla scaling law (Hoffmann et al., 2022) proposed for pre-training and the identified conditions, we define multiple parameterizations of our P^2 Law and select the most suitable parameterization. To assess the fit of different parameterizations to P^2 Law, we introduce a new metric named Average Slope Difference (ASD). As scaling laws are used to find the suitable training data size by balancing cost and performance, focusing on the slope of the predicted loss curve rather than the predicted loss values, the ASD metric is designed to measure the slope discrepancy between predicted and actual loss curves. Finally, P^2 Law is parameterized as,

$$\mathcal{L}(N_0, D, \rho, \mathcal{L}_0) = \mathcal{L}_0 + (\frac{1}{\rho})^{\gamma}(\frac{1}{N_0})^{\delta}(\frac{N_C}{N_0^{\alpha}} + \frac{D_C}{D^{\beta}} + E) \tag{1}$$

where N_C, D_C, E, α, β, γ, δ are constants, N_0 denotes the model size before pruning, D denotes the number of post-training tokens, ρ denotes the pruning rate, $\mathcal{L}_0$ denotes the model's loss before pruning, and $\mathcal{L}$ denotes the pruned model's post-training loss.

In this paper, we conduct a series of experiments to validate the P^2 Law. Taking Llama-3 series models pruned by depth pruning with a pruning rate of approximately 15% as an example, Figure 1 illustrates that P^2 Law accurately fits the actual post-training losses of the pruned model checkpoints, where compute (C) represents the computational cost calculated as $C = 6ND$ (Kaplan et al., 2020), N is the model size after pruning, and D is the number of post-training tokens. Utilizing the post-training loss curves derived by P^2 Law, we can accurately predict that the computational cost required for the post-training loss of Llama-3.2-1B to start decreasing gently is approximately 10^4. This predicted size of post-training data provides a good balance between cost and performance. Furthermore, we evaluate the generalization ability of the P^2 Law, demonstrating that P^2 Law can effectively generalizes to larger dataset sizes, larger model sizes, and higher pruning rates.

Overall, this work makes the following contributions:

- We conduct extensive studies to uncover the P^2 Law, the first scaling law for post-training after pruning, helping balance post-training cost and pruned LLM performance.
- We propose ASD, an effective metric for the evaluation of parameterizations of scaling laws for the post-training of pruned LLMs.
- We demonstrate that the P^2 Law generalizes effectively to larger dataset sizes, larger models, and higher pruning rates, offering valuable insights for optimizing pruned LLMs across diverse settings.

2 Preliminary

In this section, we present the preliminary of this work, including various pruning methods and the post-training method.

2.1 Pruning

We utilize three common pruning methods to prune LLMs, including two structured pruning methods (depth pruning (Chen et al., 2024; Song et al., 2024; Gromov et al., 2024) and width pruning (Ashkboos et al., 2024; Hu et al., 2024; Liu et al., 2024)) and a hardware-friendly variant of unstructured pruning method known as 2:4 semi-structured pruning (Sun et al., 2024; Frantar and Alistarh, 2023; Zhang et al., 2024).

Depth Pruning. Depth pruning is a structured pruning method that removes entire Transformer layers from LLMs. Specifically, depth pruning involves estimating the importance of each Transformer layers in LLMs and then removing those layers with the lowest importance.

Width Pruning. Width pruning is another structured pruning method that reduces the number of embedding channels in LLMs. This method involves measuring the importance of embedding channels and pruning the least important ones.

2:4 Semi-Structured Pruning. Unstructured pruning removes individual unimportant elements from the weight matrices, producing sparse matrices. 2:4 semi-structured pruning is a variant of unstructured pruning, with a sparse pattern of 2:4. In this pattern, every four elements in the weight matrices are grouped together, with two of the elements in each group set to zero. This semi-structured sparsity can be efficiently accelerated by hardware. We utilize SparseGPT (Frantar and Alistarh, 2023), a well-known 2:4 semi-structured pruning method, to prune LLMs.

For more details about the pruning methods used in this paper, please refer to the Appendix B.

2.2 Post-Training

After the pruning, we conduct post-training on the pruned LLMs to mitigate the performance decline. For LLMs pruned using depth or width pruning, we train all parameters of the pruned LLMs. For sparse LLMs derived from 2:4 semi-structured pruning, inspired by LoRS (Hu et al., 2025), we combine the updated weight from each training iterate with the sparse mask during the post-training process to ensure the model's sparsity, further post-training details about the 2:4 semi-structured pruning is provided in Appendix B.3.

3 Experiments for Finding Necessary Conditions Satisfied by P^2 Law

In this section, we conduct experiments on six LLMs from the Llama-3 and Qwen-2.5 series, covering various model sizes and using depth pruning, width pruning, and 2:4 semi-structured pruning.

First, we detail the pruning settings and post-training settings in Section 3.1. Next, we describe multiple trends observed in the post-training loss curves in Section 3.2. Finally, in Section 3.3, we identify several necessary conditions that the P^2 Law must satisfy based on the observed trends.

	Depth pruning	Width pruning
Llama-3.2-1B	15%,25%,30%	15%,25%,35%
Llama-3.2-3B	16%,25%,34%	15%,25%,35%
Llama-3.1-8B	16%,24%,33%	15%,25%,35%
Qwen-2.5-0.5B	15%,21%,27%	15%,25%,35%
Qwen-2.5-1.5B	15%,24%,33%	15%,25%,35%
Qwen-2.5-3B	17%,25%,32%	15%,25%,35%

Table 1: Pruning rates used for depth pruning and width pruning on different LLMs.

3.1 Settings

We conduct experiments on six LLMs from the Llama-3 and Qwen-2.5 series, including Llama-3.2-1B, Llama-3.2-3B, Llama-3.1-8B, Qwen-2.5-0.5B, Qwen-2.5-1.5B and Qwen-2.5-3B.

Pruning. The pruning rates used for depth pruning and width pruning are shown in Table 1. The pruning processes have been introduced in Appendix B. We randomly select 1,024 samples from the pre-training dataset SlimPajama for pruning.

Post-Training. For Llama-3.2-3B, Qwen-2.5-3B and Llama-3.1-8B, we randomly select 1B tokens from SlimPajama for post-training. For Llama-3.2-1B, Qwen-2.5-0.5B, Qwen-2.5-1.5B, we randomly select 0.5B tokens from SlimPajama for post-training. During the post-training process, we set the learning rate to 2e-5 and the batch size to 262k tokens. All post-training processes are conducted on 4 Nvidia A800-80G GPUs and 4 Nvidia A6000-48G GPUs. The entire training process takes a total of 500 hours. For more details about batch size and learning rate settings, please refer to the Appendix C.

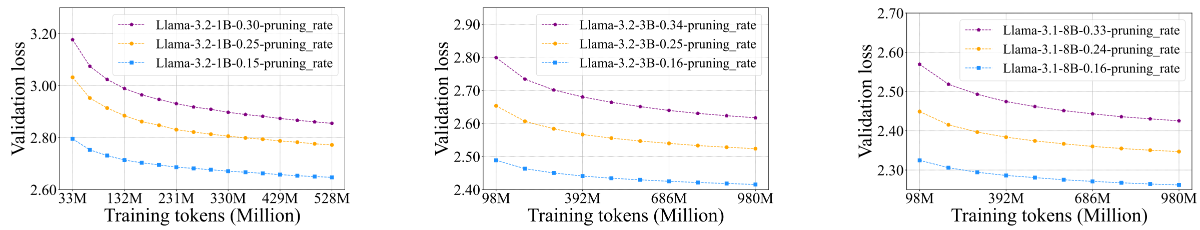
3.2 Trends of the Post-Training Loss Curves

To better explore the trends of the post-training loss curves, we define:

Definition 1 ***Relative post-training loss*** $\Delta\mathcal{L}$. *The relative post-training loss is the difference between the pruned model's post-training loss* $\mathcal{L}$ *and the model's loss* $\mathcal{L}_0$ *before pruning.*

$$\Delta\mathcal{L} = \mathcal{L} - \mathcal{L}_0 \tag{2}$$

Definition 2 ***Normalized relative post-training loss*** $\Delta\mathcal{L}_{norm}$. *The normalized relative post-training loss is defined as the ratio of the relative*



(a) Post-training loss curves of Llama-3.2-1B pruned by depth pruning with different pruning rates.

(b) Post-training loss curves of Llama-3.2-3B pruned by depth pruning with different pruning rates.

(c) Post-training loss curves of Llama-3.1-8B pruned by depth pruning with different pruning rates.

Figure 2: Post-training loss curves of Llama-3 series models pruned by depth pruning with different pruning rates.

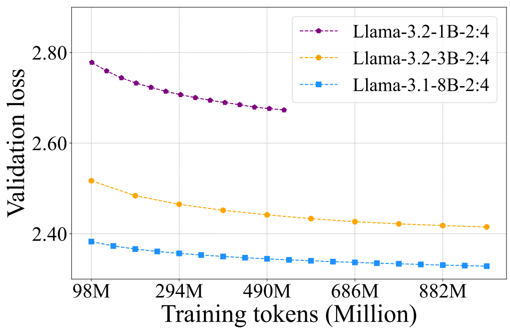


Figure 3: Post-training loss curves of Llama-3 series models pruned by 2:4 semi-structured pruning.

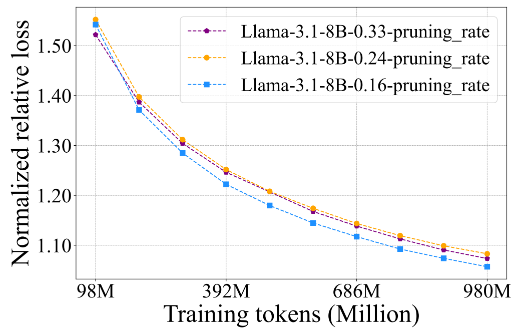


Figure 4: Normalized relative post-training loss curves of Llama-3.1-8B pruned by depth pruning.

post-training loss $\Delta\mathcal{L}$ to a power-law function of the pruning rate ρ.

$$\Delta\mathcal{L}_{norm} = \frac{\Delta\mathcal{L}}{(\frac{1}{\rho})^{\gamma}} \tag{3}$$

where γ is a constant.

In Figures 2 and 3, we present the post-training loss curves for the Llama-3 series models pruned by depth pruning and 2:4 semi-structured pruning. Additional post-training loss curves (exhibiting similar trends) are shown in Figures 8, 9, 10, and 11 in the Appendix D. By analyzing the post-training loss curves, we observe the following trends:

- **Trend 1: Smaller LLMs exhibit faster decreases in post-training loss.** For instance, as shown in Figure 3, with 2:4 semi-structured pruning, the post training loss curve of Llama-3.1-8B is much flatter compared to those of Llama-3.2-3B and Llama-3.2-1B. The same trend is observed under both depth pruning and width pruning, as depicted in Figure 2 and Figure 8. This suggests that smaller LLMs exhibit faster decreases in post-training loss.

- **Trend 2: Relative post-training loss $\Delta\mathcal{L}$ follows a power-law relationship with the pruning rate ρ.** As shown in Figure 4, with depth pruning, the normalized relative post-training loss curves of Llama-3.1-8B at various pruning rates nearly overlap. This can be formally expressed as:

$$\frac{\Delta\mathcal{L}^{(0.33)}}{(\frac{1}{0.33})^{\gamma}} \approx \frac{\Delta\mathcal{L}^{(0.24)}}{(\frac{1}{0.24})^{\gamma}} \approx \frac{\Delta\mathcal{L}^{(0.16)}}{(\frac{1}{0.16})^{\gamma}} \tag{4}$$

where $\Delta\mathcal{L}^{(0.33)}$, $\Delta\mathcal{L}^{(0.24)}$, and $\Delta\mathcal{L}^{(0.16)}$ represent the relative post-training loss of Llama-3.1-8B pruned by depth pruning with pruning rates ρ of 0.33, 0.24, and 0.16, respectively. This demonstrates that the pruning rate and the relative post-training loss are governed by a power-law relationship.

3.3 Necessary Conditions Satisfied by P^2 Law

Based on the aforementioned trends, we identify three fundamental conditions for the P^2 Law:

- **Condition 1.** The post-training loss $\mathcal{L}$ decreases as the number of post-training tokens D increases:

$$\frac{\partial\mathcal{L}}{\partial D} < 0 \tag{5}$$

- **Condition 2.** As derived from Trend 1 in Section 3.2, under similar pruning rates, the post-training loss curves of smaller LLMs decrease faster as the number of post-training tokens D increases:

$$\frac{\partial}{\partial N_0}\left(\frac{\partial \mathcal{L}}{\partial D}\right) = \frac{\partial^2 \mathcal{L}}{\partial N_0 \partial D} > 0 \tag{6}$$

  where N_0 is the model size before pruning.

- **Condition 3.** From Eq.4 in Trend 2, the relative post-training loss $\Delta\mathcal{L}$ follows a power-law relationship with the pruning rate ρ:

$$\Delta\mathcal{L} \propto \left(\frac{1}{\rho}\right)^{\gamma} \tag{7}$$

An ideal P[2] Law should satisfy aforementioned three conditions. Additionally, the P[2] Law should also satisfy the condition that when the pruning rate ρ is 0, the relative post-training loss $\Delta\mathcal{L}$ is 0, which is a necessary condition for Condition 3.

4 P[2] Law

In this section, we aim to parameterize the P[2] Law according to the above three necessary conditions. In Section 4.1, we introduce the metric for assess the quality of different candidate parametrizations. Next, based on the Chinchilla scaling law, we define multiple parameterizations for our P[2] Law and select the most suitable one in Section 4.2. Finally, in Section 4.3, we demonstrate the generalization ability of the P[2] Law from three perspectives: dataset size, model size, and pruning rate.

4.1 Metric for Accessing Law Fitting

Following prior work (Que et al., 2024), we utilize both R^2 (Fisher, 1922) and Huber loss (Huber, 1992) to evaluate different parameterizations of scaling law. The R^2 value, reflecting the proportion of variance explained, trends toward 1 as the fit becomes more robust. Huber loss, a robust loss function, blends the characteristics of mean squared error and mean absolute error, making it less sensitive to outliers. The Huber loss is a positive number, and a lower Huber loss suggests a better fit.

Scaling laws are often used to determine the optimal amount of training data by balancing computational cost and model performance. For instance, as shown in Figure 5, there is one actual loss curve and two predicted loss curves. Traditional metrics like R^2 and Huber loss indicate that predicted curve 1 better matches the actual curve. However, the convergence trend predicted by curve 1 deviates significantly from the actual convergence trend. While curve 1 predicts that the loss has flattened, the actual loss continues to decrease. On the other hand, while predicted curve 2 deviates more from the actual curve in terms of absolute values, its slope is consistently closer to the actual curve. This makes its prediction of the flattening point much more accurate. To address this issue, we propose a new metric called Average Slope Difference (ASD), which measures the difference between the slope of the loss curve predicted by the scaling law and the slope of the actual loss curve. ASD is formally defined as:

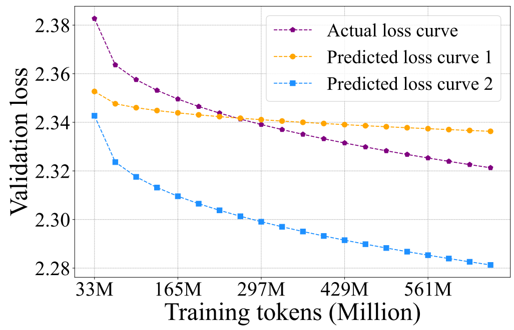


Figure 5: An example showcasing the advantages of ASD. The ASD of predicted loss curve 2 is lower because its slope is closer to that of the actual loss curve.

$$\text{ASD} = \frac{1}{N}\sum_{i=2}^{N} |(y_i - y_{i-1}) - (\hat{y}_i - \hat{y}_{i-1})| \tag{8}$$

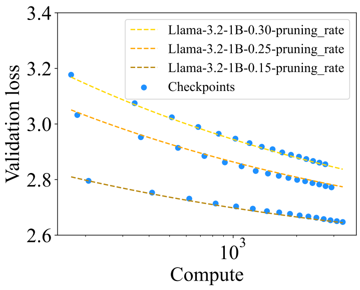
where y_i represents the loss of N points uniformly sampled from the actual loss curve as the number of post-training tokens increases, and $\hat{y}_i$ represents the corresponding loss values on the curve predicted by the scaling law. Since the early parts of the loss curve during post-training do not represent true convergence, we only sample points from the latter half of the training process. A smaller ASD value indicates that the predicted loss curve's slope more closely matches the slope of the actual loss curve.

4.2 Derivation of P[2] Law

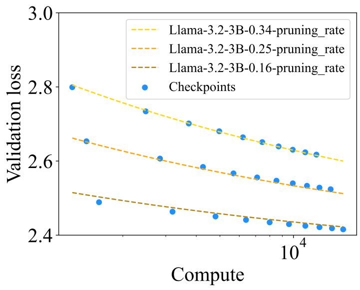
Previous efforts have explored scaling laws for pre-training of LLMs, with Chinchilla scaling (Hoffmann et al., 2022) being a superior work, and we choose it as the foundational parameterization for our P[2] Law. The Chinchilla scaling law describes the relationship between model performance and

LLM	Parameterizations	Depth pruning R^2	Depth pruning Huber loss	Depth pruning ASD	Width pruning R^2	Width pruning Huber loss	Width pruning ASD	2:4 semi-structured pruning R^2	2:4 semi-structured pruning Huber loss	2:4 semi-structured pruning ASD
Llama-3 series	$\mathcal{L}_1$	**0.9717**	**0.000016**	**0.000619**	**-1.2985**	**0.000177**	**0.000592**	**0.8126**	**0.000056**	**0.001466**
	$\mathcal{L}_2$	0.9300	0.000045	0.001150	-2.5578	0.000450	0.001419	0.7797	0.000079	0.002294
	$\mathcal{L}_3$	0.7737	0.000118	0.000827	-4.5905	0.000776	0.001754	-0.2555	0.000493	0.002054
Qwen-2.5 series	$\mathcal{L}_1$	**0.9781**	**0.000011**	**0.000524**	**0.9891**	**0.000010**	**0.000648**	**0.9995**	**0.000000**	**0.000191**
	$\mathcal{L}_2$	0.9423	0.000031	0.000879	0.9803	0.000027	0.000712	0.9867	0.000010	0.000753
	$\mathcal{L}_3$	0.8855	0.000075	0.001270	0.9824	0.000024	0.000733	0.9930	0.000005	0.000491

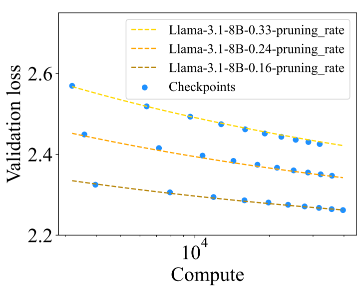
Table 2: Evaluation of three parameterizations for P² Law fitting.



(a) Loss curves derived by P² Law and the actual checkpoints of Llama-3.2-1B pruned by depth pruning.



(b) Loss curves derived by P² Law and the actual checkpoints of Llama-3.2-3B pruned by depth pruning.



(c) Loss curves derived by P² Law and the actual checkpoints of Llama-3.1-8B pruned by depth pruning.

Figure 6: Loss curves derived by P² Law and the actual checkpoints of Llama-3 series models pruned by depth pruning.

key factors such as model size, the number of pre-training tokens, and the computational resources used during the pre-training process. It is formally defined as follows:

$$\mathcal{L}(N, D) = \frac{N_C}{N^\alpha} + \frac{D_C}{D^\beta} + E \tag{9}$$

where N_C, D_C, E, α, and β are constants, N represents the model size, D denotes the number of pre-training tokens and $\mathcal{L}$ represents the model's loss. Compared to the OpenAI scaling law (Kaplan et al., 2020), the Chinchilla scaling law demonstrates superior performance (detailed in Appendix E). Therefore, we adopt the Chinchilla scaling law as the foundational parameterization for our P² Law. Combining the pruning rate ρ and the model's loss $\mathcal{L}_0$ before pruning, we define the following three candidate parameterizations:

$$\mathcal{L}_1(N_0, D, \rho, \mathcal{L}_0) = \mathcal{L}_0 + (\frac{1}{\rho})^\gamma (\frac{1}{N_0})^\delta (\frac{N_C}{N_0^\alpha} + \frac{D_C}{D^\beta} + E)$$

$$\mathcal{L}_2(N_0, D, \rho, \mathcal{L}_0) = \mathcal{L}_0 + (\frac{1}{\rho})^\gamma (\frac{N_C}{N_0^\alpha} + \frac{D_C}{D^\beta} + E)$$

$$\mathcal{L}_3(N_0, D, \rho, \mathcal{L}_0) = \mathcal{L}_0 + (\frac{1}{\rho})^\gamma (\frac{1}{N_0})^\delta (\frac{D_C}{D^\beta} + E)$$

where N_C, D_C, E, α, β, γ and δ are constants, N_0 denotes the model size before pruning, D denotes the number of post-training tokens and $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ denote the pruned model's post-training loss. Additionally, since there is no pruning rate in the 2:4 semi-structured pruning, P² Law for the 2:4 semi-structured pruning does not need to satisfy Condition 3. As a result, both the pruning rate and the loss before pruning are omitted and we adjust the parameterizations to:

$$\mathcal{L}_1(N_0, D) = (\frac{1}{N_0})^\delta (\frac{N_C}{N_0^\alpha} + \frac{D_C}{D^\beta} + E) \tag{10}$$

$$\mathcal{L}_2(N_0, D) = (\frac{N_C}{N_0^\alpha} + \frac{D_C}{D^\beta} + E) \tag{11}$$

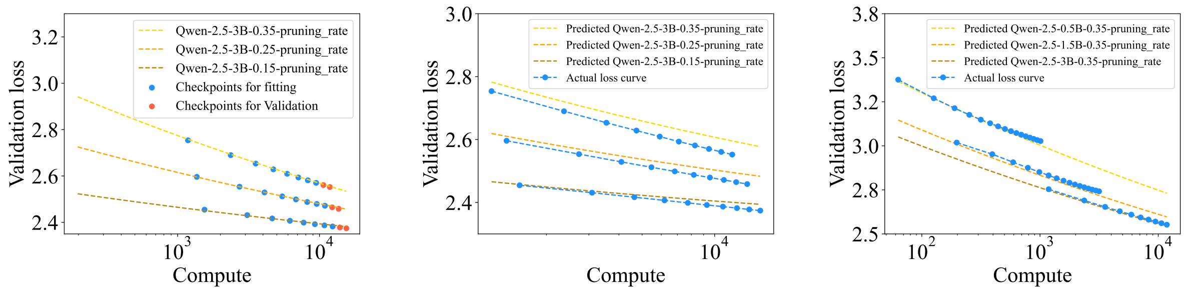
$$\mathcal{L}_3(N_0, D) = (\frac{1}{N_0})^\delta (\frac{D_C}{D^\beta} + E) \tag{12}$$

We utilize all checkpoints to fit the three candidate parameterizations through Levenberg-Marquardt method (Moré, 2006), and the specific parameter values (i.e., the values of N_C, D_C, E, α, β, γ, and δ) for the fitted $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ are provided in Table 5 in Appendix F. As shown in Table 2, $\mathcal{L}_1$ significantly outperforms $\mathcal{L}_2$ and $\mathcal{L}_3$ in terms of the R^2, Huber loss, and ASD metrics. Additionally, as shown in Table 6 in Appendix F, after our calculation and verification, $\mathcal{L}_2$ and some fitted $\mathcal{L}_3$ fails to satisfy Condition 2. In contrast, all of the fitted $\mathcal{L}_1$ satisfy all three conditions. **Based on the experimental results, we select $\mathcal{L}_1$ as the parameterization for our P² Law.**

In Figure 6, we show the loss curves $\mathcal{L}_1$ derived by P² Law alongside the actual checkpoints of the

LLM	Generalization	Depth pruning R^2	Depth pruning Huber loss	Depth pruning ASD	Width pruning R^2	Width pruning Huber loss	Width pruning ASD	2:4 semi-structured pruning R^2	2:4 semi-structured pruning Huber loss	2:4 semi-structured pruning ASD
Llama-3 series	Dataset size	0.9725	0.000016	0.001001	0.9270	0.000019	0.000674	0.8244	0.000063	0.002561
	Model size	-0.5441	0.000745	0.001321	-	-	-	-	-	-
	Pruning rate	0.9676	0.000059	0.000879	0.9707	0.000056	0.001123	-	-	-
Qwen-2.5 series	Dataset size	0.9780	0.000012	0.001026	0.9896	0.000010	0.001116	0.9940	0.000000	0.000299
	Model size	-0.8786	0.000763	0.001573	0.8627	0.000137	0.001772	-	-	-
	Pruning rate	0.9660	0.000043	0.000920	0.9704	0.000095	0.001003	-	-	-

Table 3: Evaluation of generalization results from the perspectives of dataset size, model size, and pruning rate.



(a) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation.

(b) P^2 Law is fitted using checkpoints from smaller LLMs and used to predict the loss curves of larger LLMs.

(c) P^2 Law is fitted using checkpoints from smaller pruning rates and used to predict the loss curves of larger ones.

Figure 7: Generalization of the P^2 Law for Qwen-2.5 series models pruned by width pruning.

Llama-3 series models pruned by depth pruning, where the compute (C) is approximated using the empirical formula $C = 6ND$ (Kaplan et al., 2020), and N denotes the model size after pruning. Additional loss curve derived by P^2 Law are shown in Figure 13, 14, 15 and 16 in Appendix G. As shown in these figures, the loss curve derived by P^2 Law accurately aligns with all actual checkpoints, under all the three pruning methods, except for the width pruning on Llama-3.1-8B. As shown in Figure 2c and 8c, for Llama-3.1-8B, we observe that depth pruning outperforms width pruning at similar pruning rates, which contrasts with the observations in other cases. This suggests that width pruning on Llama-3.1-8B may lead to anomalous performance, making our law unsuitable for this special scenario. We elaborate on this anomalous performance of width pruning on Llama-3.1-8B further in Appendix H.

4.3 Generalization of P^2 Law

In this section, we explore the generalization ability of P^2 Law from three perspectives: dataset size, model size and pruning rate.

4.3.1 Settings

We begin by outlining the settings of generalization experiments as follows:

Dataset Size. The fitting setting follows the same setting as described in Section 4.2, with the only difference being that the first 80% of the checkpoints recorded during each training process are used to fit the P^2 Law, and the remaining 20% for validation.

Model size. We fit the P^2 Law using checkpoints from smaller LLMs and validate it on checkpoints from larger LLMs, while maintaining the pruning rate during both fitting and prediction. Taking the Qwen-2.5 series models as an example, we fit the P^2 Law using all checkpoints from Qwen-2.5-0.5B and Qwen-2.5-1.5B, and subsequently validate it with the actual checkpoints of Qwen-2.5-3B across three pruning rates. Due to the limited number of available actual loss curves for 2:4 semi-structured pruning, we did not conduct experiments for this pruning method.

Pruning Rate. We fit the P^2 Law using checkpoints from lower pruning rates and validate it using checkpoints from higher pruning rates, while keeping the model size constant during both fitting and prediction. Taking width pruning of the Qwen-2.5 series models as an example, we fit the P^2 Law using checkpoints from these models at lower pruning rates (0.15 and 0.25) and then validate it with the actual checkpoints at a higher pruning rate of 0.35. Since there is no pruning rate in the 2:4 semi-structured pruning, we only explore the generalization ability on pruning rates under depth pruning and width pruning.

Due to the anomaly of width pruning on Llama-3.1-8B (see Section 4.2), we exclude this model from generalization experiments.

4.3.2 Experimental Results

Dataset Size Generalization. The evaluation results are shown in Table 3, and the loss curves of Qwen-2.5-3B (pruned by width pruning) derived by P^2 Law are illustrated in Figure 7a. Additional loss curves derived by P^2 Law are provided in Figures 18, 19, 20, and 21 in Appendix I. The results in Table 3 show that the loss curves derived by P^2 Law accurately matches the validation checkpoints, indicating that the P^2 Law generalizes well to larger dataset sizes.

Model Size Generalization. The evaluation results are presented in Table 3, and the loss curves of Qwen-2.5-3B predicted by P^2 Law (pruned by width pruning) are visualized in Figure 7b. Additional loss curves predicted by P^2 Law are shown in Figure 23 in Appendix I. As shown in Table 3, the P^2 Law fitted on smaller LLMs performs poorly in R^2 and Huber loss when applied to larger models, indicating challenges in generalizing to larger, unseen models. However, the low ASD suggests it still captures the slope of the actual loss curve. This trend is also seen in Figure 23, where despite a gap between predicted and actual loss curves, the predicted and actual loss curves align in their downward trend after training stabilizes. This suggests P^2 Law fitted from smaller LLMs can still predict the optimal computation cost point for larger LLMs, confirming its generalization feasibility.

Pruning Rate Generalization. We present the generalization evaluations in Table 3 and illustrate the loss curves of Qwen-2.5 series models predicted by P^2 Law (pruned by width pruning) in Figure 7c. Additional loss curves predicted by P^2 Law are provided in Figure 24 and 25 in Appendix I. As shown in the Figure 7c and Table 3, the values of different metrics indicate that the actual loss curves closely align with the predicted loss curves, suggesting that the P^2 Law generalizes well to higher pruning rates.

5 Related Work

5.1 Model Pruning

Model pruning can be categorized into unstructured pruning and structured pruning.

Unstructured Pruning. Unstructured pruning methods (Frantar and Alistarh, 2023; Zhang et al., 2024; Sun et al., 2024) compress LLMs by removing individual unimportant elements from the weight matrices, producing sparse ones. However, it is often hardware-inefficient and only speeds up LLMs when a specific sparsity pattern, such as 2:4 sparsity (Mishra et al., 2021), is applied. The approach which employ the 2:4 sparsity is known as semi-structured pruning.

Structured Pruning. Structured pruning methods for LLMs can be divided into two categories: depth pruning (Chen et al., 2024; Song et al., 2024; Gromov et al., 2024; Men et al., 2024), which aims to reduce the number of layers in the LLMs, and width pruning (Ashkboos et al., 2024; Hu et al., 2024; Liu et al., 2024; Ma et al., 2023), which aims to reduce the embedding channels, the number of attention heads, or the intermediate size of the FFN.

5.2 Scaling Law

The OpenAI scaling law (Kaplan et al., 2020) and the Chinchilla scaling law (Hoffmann et al., 2022) are the most popular scaling laws in the pre-training of LLMs, both of which establishe a power-law relationship between model performance, model size, the number of pre-training tokens, and the computational resources used during pre-training.

We are the first to investigate the scaling law for the post-training after model pruning, and we propose the P^2 Law as a scaling law for this process. Compared to the OpenAI scaling law, the Chinchilla scaling law demonstrates superior performance (detailed in Appendix E). Therefore, we adopt the Chinchilla scaling law as the foundational parameterization for our P^2 Law.

6 Conclusion

In this paper, we conduct post-training experiments on models from the Llama-3 and Qwen-2.5 series, covering various sizes and employing both typical structured and semi-structured pruning methods. Through extensive experiments, we identify the P^2 Law — the first scaling law for post-training after model pruning. Further experiments validate the effectiveness of the P^2 Law and demonstrate its generalization to larger dataset sizes, larger model sizes, and higher pruning rates, offering valuable insights for resource allocation in the post-training of pruned LLMs.

Limitation

Due to constraints in GPU resources, the experiments conducted in this paper are restricted to LLMs with fewer than 8B parameters. Given the substantial increase in experimental costs for larger-scale models—for instance, training a 70B LLM with 1B tokens on 4 A800 GPUs would require approximately 1,000 hours—we intend to expand our experiments to larger models as soon as sufficient computational resources become available. This will enable us to further validate the applicability of the P^2 Law across a broader range of model parameter scales.

Acknowledgments

This work is supported by the National Key Research & Develop Plan (2023YFF0725100) and the National Natural Science Foundation of China (62322214, U23A20299, U24B20144, 62172424, 62276270).

References

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns. *Preprint*, arXiv:2401.15024.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. 2024. Streamlining redundant layers to compress large language models. *Preprint*, arXiv:2403.19135.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ronald A Fisher. 1922. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. *Preprint*, arXiv:2301.00774.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. 2024. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Preprint*, arXiv:1510.00149.

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, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Yuxuan Hu, Jing Zhang, Xiaodong Chen, Zhe Zhao, Cuiping Li, and Hong Chen. 2025. Lors: Efficient low-rank adaptation for sparse large language model. *Preprint*, arXiv:2501.08582.

Yuxuan Hu, Jing Zhang, Zhe Zhao, Chen Zhao, Xiaodong Chen, Cuiping Li, and Hong Chen. 2024. sp^3: Enhancing structured pruning via pca projection. *Preprint*, arXiv:2308.16475.

Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

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

Yijiang Liu, Huanrui Yang, Youxin Chen, Rongyu Zhang, Miao Wang, Yuan Du, and Li Du. 2024. Pat: Pruning-aware tuning for large language models. *Preprint*, arXiv:2408.14721.

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

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. 2018. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*.

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.

Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. *Preprint*, arXiv:2104.08378.

Jorge J Moré. 2006. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*, pages 105–116. Springer.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. *Preprint*, arXiv:2407.14679.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, et al. 2024. D-cpt law: Domain-specific continual pre-training scaling law for large language models. *arXiv preprint arXiv:2406.01375*.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. 2024. Sleb: Streamlining llms through redundancy verification and elimination of transformer blocks. *Preprint*, arXiv:2402.09025.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. *Preprint*, arXiv:2306.11695.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv*.

Yifei Yang, Zouying Cao, and Hai Zhao. 2024. Laco: Large language model pruning via layer collapse. *arXiv preprint arXiv:2402.11187*.

Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. 2024. Plug-and-play: An efficient post-training pruning method for large language models. In *The Twelfth International Conference on Learning Representations*.

A License

Our research is grounded in the SlimPajama training dataset, which is distributed under the Apache 2.0 license. This license allows for the free use, modification, reproduction, and distribution of the software, both for personal and commercial purposes. Consistent with open science practices, we will make our training data publicly available upon acceptance of this work. The data will be released under the CC BY-SA 4.0 license, which enables reuse and redistribution, provided that derivative works adhere to the same licensing terms

B Details of Pruning Methods

B.1 Depth Pruning

Following the existing depth pruning methods (Men et al., 2024; Chen et al., 2024; Yang et al., 2024), we estimate the layer importance using cosine similarity and prune layers with lower importance. Specifically, we randomly select N samples from the pre-training data. We then record the hidden states generated by the LLMs for these samples and compute the cosine similarity between the input and output hidden states of each layer. Assuming that the input hidden states of layer i are represented by $\boldsymbol{x}^{(i)}$, the importance score (IS) of layer i is computed as:

$$\text{IS}^{\text{layer},i} = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{L}\sum_{k=1}^{L}\frac{\boldsymbol{x}_{j,k}^{(i)} \cdot \boldsymbol{x}_{j,k}^{(i+1)}}{\|\boldsymbol{x}_{j,k}^{(i)}\| \cdot \|\boldsymbol{x}_{j,k}^{(i+1)}\|}\right) \tag{13}$$

where $\boldsymbol{x}_j^{(i)}$, $\boldsymbol{x}_j^{(i+1)} \in \mathbb{R}^{d\times L}$ denotes the input and output hidden states of the j-th sample respectively, L denotes the sequence length and d denotes the hidden size. Given the number of pruned layers n determined by the target sparsity, we remove the n layers corresponding to the top-n highest cosine similarities for pruning.

B.2 Width Pruning

Following the approaches of Wanda (Sun et al., 2024) and MINITRON (Muralidharan et al., 2024), we utilize activation-based metrics for width pruning. Specifically, we randomly select N samples from the pre-training data and assess the importance of embedding channels by analyzing the activations generated by the LayerNorm layers. We then prune the least important channels based on this analysis. The formula for calculating the importance score (IS) of embedding channels (emb) is as follows:

$$\text{IS}^{\text{emb},i} = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{L}\sum_{k=1}^{L}\left|LN(\boldsymbol{x}_{j,k,i}^{LN})\right|\right) \tag{14}$$

where $\boldsymbol{x}_{j,k,i}^{LN}$ denotes the input of the i-th channel of the k-th token in the j-th sample at the LayerNorm layer, L denotes the sequence length, and LN denotes the Layer Normalization operaten. Given a specific sparsity, we calculate the number of embedding channels that need to be pruned, and then remove the channels with the lowest importance.

B.3 2:4 Semi-Structured Pruning

Unstructured pruning removes individual unimportant elements from the weight matrices, producing sparse matrices. When the sparsity structure follows a specific pattern, such as 2:4 sparsity (Mishra et al., 2021), the model can be efficiently accelerated. This approach is known as semi-structured pruning. Let W represent the weight matrix of a linear layer of an LLM, x represent the input of the linear layer. The object of semi-structured pruning is to learn a sparsity mask M and an updated weight ΔW so that the dense matrix W is transformed into a sparse matrix $\tilde{W}$:

$$\begin{aligned} &min \ \ \|Wx - \tilde{W}x\| \\ &s.t. \ \ \ \tilde{W} = M \cdot (W + \Delta W) \end{aligned} \tag{15}$$

where $W \in \mathbb{R}^{d_{out}\times d_{in}}$, $M \in \{0,1\}^{d_{out}\times d_{in}}$, $\Delta W \in \mathbb{R}^{d_{out}\times d_{in}}$ and $x \in \mathbb{R}^{d_{in}}$.

We randomly select 1,024 data samples from the pre-training dataset SlimPajama for pruning. and use SparseGPT (Frantar and Alistarh, 2023) to optimize the aforementioned objectives.

In the post-training process, We train this 2:4 sparse model pruned by SparseGPT. Inspired by LoRS (Hu et al., 2025), during the post-training process, we combine the updated weight $\Delta\tilde{W}^t$ from each training iterate t with the mask M to obtain the weight after update $\tilde{W}^t$, ensuring the model's sparsity:

$$\tilde{W}^t = \tilde{W}^{t-1} + M \cdot \Delta\tilde{W}^t \tag{16}$$

C Batch Size and Learning Rate Settings

Previous research indicates that the relationship between batch size and the number of model parameters is very weak (McCandlish et al., 2018).

Furthermore, OpenAI Scaling Law also utilize the same batch size for models with varying parameter counts. As a result, we apply a consistent and commonly used batch size of 262k tokens across models of different scales. Regarding the learning rate, OpenAI suggests that the optimal learning rate follows a logarithmic relationship with the size of the model parameters (Kaplan et al., 2020). Based on their provided formula, the optimal learning rate for 8B models is calculated to be 2e-3, while for 0.5B models, it is 1.8e-3, indicating a minimal difference. Furthermore, our experiments reveal that the optimal learning rate for post-training of models ranging from 0.5B to 8B is approximately 2e-5. Therefore, we adopt a uniform learning rate across models of different scales.

D Additional Actual Loss Curves

The additional post-training loss curves for models pruned by width pruning or for the Qwen-2.5 series models are provided in Figures 8, 9, 10 and 11.

E Comparison with OpenAI Scaling Law

Kaplan (Kaplan et al., 2020) propose OpenAI scaling law as follows:

$$\mathcal{L}(N, D) = (\frac{N_C}{N^\alpha} + \frac{D_C}{D})^\beta \tag{17}$$

where N_C, D_c, α and β are constants, N denotes the model size and D denotes the number of pre-training tokens. We have also defined the following parameterizations based on the OpenAI scaling law:

$$\mathcal{L}_4(N_0, D, \rho, \mathcal{L}_0) = \mathcal{L}_0 + (\frac{1}{\rho})^\gamma (\frac{1}{N_0})^\delta (\frac{N_C}{N_0^\alpha} + \frac{D_C}{D})^\beta \tag{18}$$

$$\mathcal{L}_5(N_0, D, \rho, \mathcal{L}_0) = \mathcal{L}_0 + (\frac{1}{\rho})^\gamma (\frac{N_C}{N_0^\alpha} + \frac{D_C}{D})^\beta \tag{19}$$

where N_C, D_C, α, β, γ, δ denotes constants, N_0 denotes the model size before pruning, D denotes the number of post-training tokens, ρ denotes pruning rate, $\mathcal{L}_0$ denotes the model's loss before pruning and $\mathcal{L}_4, \mathcal{L}_5$ denote pruned model's post-training loss.

We utilize all the checkpoints to fit the two parameterizations described above, and the evaluation results are presented in Table 4. The results show that the performance of these two parameterizations is weaker than that of $\mathcal{L}_1$. Therefore, we adopt the Chinchilla scaling law as the foundational parameterization for our P^2 Law.

F Parameter Values of Fitted Parameterizations

We present the parameter values of the fitted $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ in the Table 5. In addition, we calculate whether $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ satisfy Condition 2, and the results are shown in the Table 6.

G Additional Loss Curves Derived by P^2 Law

The additional loss curves derived by P^2 Law are shown in the Figure 13, 14, 15 and 16.

H Patterns of the Llama-3 Series Models in Terms of Width

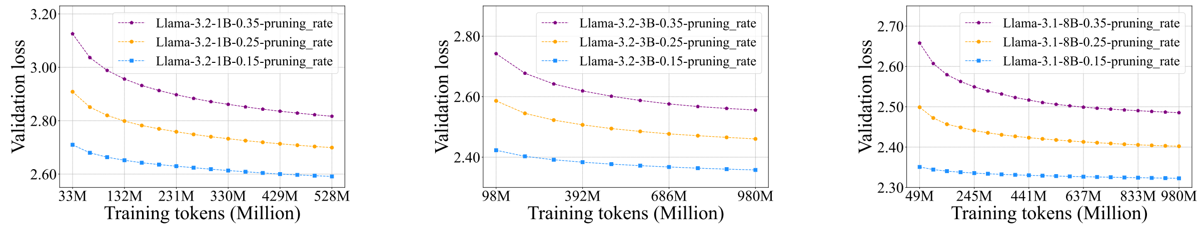
As discussed in Section 4.2, we observe an anomalous phenomenon in Llama-3.1-8B under width pruning. To investigate this further, we analyze the behavior of the Llama-3 series models with respect to width. Using a random sample of 1024 data points from SlimPajama and applying Eq.14, we plot the importance score distributions of the embedding channels for the Llama-3 series models, as shown in Figure 17. For easier comparison, we normalize the Importance score, which is defined as follows:

$$\mathrm{IS}^{\mathrm{emb},i} = \frac{\mathrm{IS}^{\mathrm{emb},i} - \min(\mathrm{IS}^{\mathrm{emb},1}, ..., \mathrm{IS}^{\mathrm{emb},\mathrm{N_d}})}{\max(\mathrm{IS}^{\mathrm{emb},1}, ..., \mathrm{IS}^{\mathrm{emb},\mathrm{N_d}})}$$

where the N_d denotes the number of embedding channels. Additionally, we remove the extremely high values that represent a very small proportion of the data. The figure shows that the importance scores of Llama-3.1-8B are more densely distributed compared to those of Llama-3.2-1B and Llama-3.2-3B. This denser distribution may hinder the ability to effectively distinguish less important channels in Llama-3.1-8B based on importance scores, which could potentially explain the observed anomalies in Llama-3.1-8B.

I Additional Generalization Loss Curves

We present the additional dataset size generalization predicted loss curves in the Figure 18, 19, 20, 21 and 22, model size generalization predicted loss curves in the Figure 23 and pruning rate generalization predicted loss curves in the Figure 24 and 25.

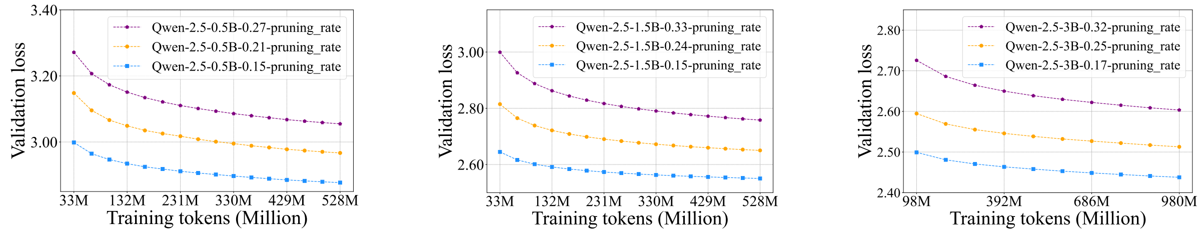


(a) Post-training loss curves of Llama-3.2-1B pruned by width pruning with different pruning rates.

(b) Post-training loss curves of Llama-3.2-3B pruned by width pruning with different pruning rates.

(c) Post-training loss curves of Llama-3.1-8B pruned by width pruning with different pruning rates.

Figure 8: Post-training loss curves of Llama-3 series models pruned by width pruning with different pruning rates.

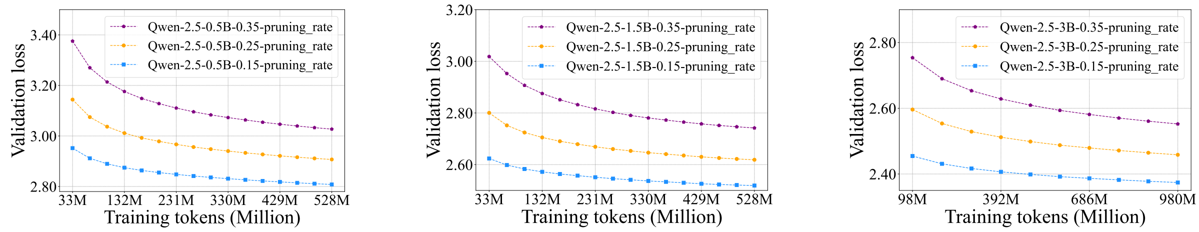


(a) Post-training loss curves of Qwen-2.5-0.5B pruned by depth pruning with different pruning rates.

(b) Post-training loss curves of Qwen-2.5-1.5B pruned by depth pruning with different pruning rates.

(c) Post-training loss curves of Qwen-2.5-3B pruned by depth pruning with different pruning rates.

Figure 9: Post-training loss curves of Qwen-2.5 series models pruned by depth pruning with different pruning rates.



(a) Post-training loss curves of Qwen-2.5-0.5B pruned by width pruning with different pruning rates.

(b) Post-training loss curves of Qwen-2.5-1.5B pruned by width pruning with different pruning rates.

(c) Post-training loss curves of Qwen-2.5-3B pruned by width pruning with different pruning rates.

Figure 10: Post-training loss curves of Qwen-2.5 series models pruned by width pruning with different pruning rates.

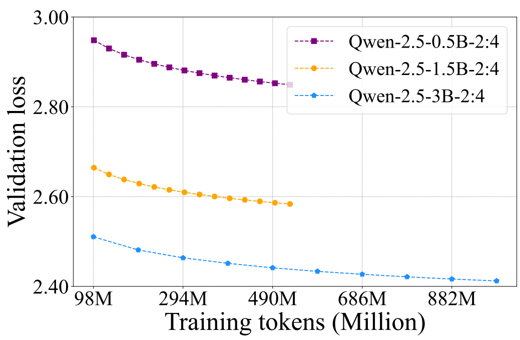


Figure 11: Post-training loss curves of Qwen-2.5 series models pruned by 2:4 semi-structured pruning.

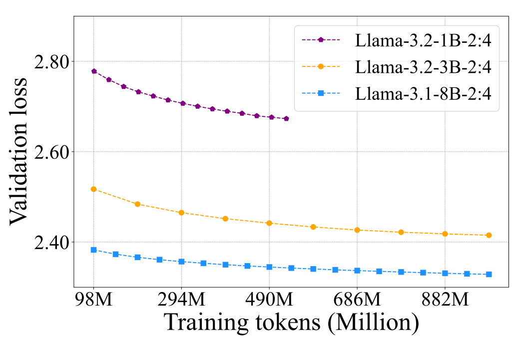


Figure 12: Post-training loss curves of Llama-3 series models pruned by 2:4 semi-structured pruning.

LLM	Parameterizations	Depth pruning R^2	Depth pruning Huber loss	Depth pruning ASD	Width pruning R^2	Width pruning Huber loss	Width pruning ASD	2:4 semi-structured pruning R^2	2:4 semi-structured pruning Huber loss	2:4 semi-structured pruning ASD
Llama-3 series	$\mathcal{L}_1$	**0.9717**	**0.000016**	**0.000619**	**-1.2985**	**0.000177**	**0.000592**	**0.8126**	**0.000056**	**0.001466**
	$\mathcal{L}_4$	0.9339	0.000035	0.001482	-1.3660	0.000203	0.000814	0.7157	0.000112	0.002117
	$\mathcal{L}_5$	0.6535	0.000198	0.001822	-1.7948	0.000345	0.000729	-0.3809	0.000638	0.003687
Qwen-2.5 series	$\mathcal{L}_1$	**0.9781**	**0.000011**	**0.000524**	**0.9891**	**0.000010**	**0.000648**	**0.9995**	**0.000000**	**0.000191**
	$\mathcal{L}_4$	0.7730	0.000192	0.004085	0.9838	0.000015	0.001126	0.9960	0.000002	0.000550
	$\mathcal{L}_5$	0.8283	0.000134	0.002648	0.9694	0.000040	0.001007	0.8360	0.000118	0.003925

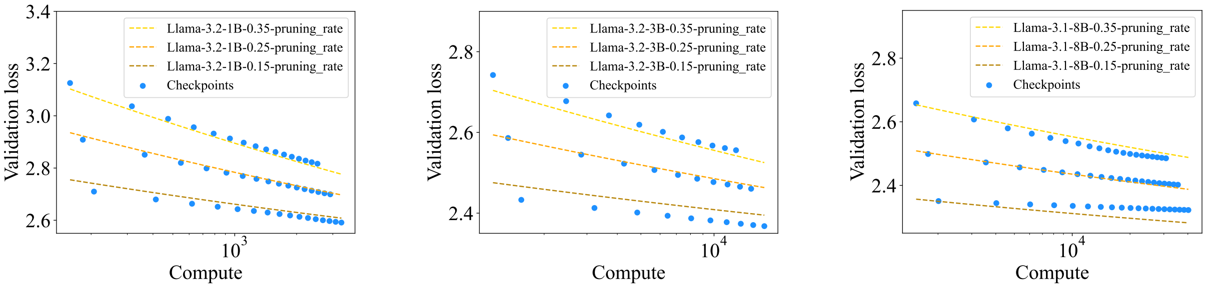
Table 4: Comparison of law fitting results between OpenAI scaling law and Chinchilla scaling law.

LLM	Parameterizations	Depth pruning N_c	D_c	E	α	β	γ	δ	Width pruning N_c	D_c	E	α	β	γ	δ	2:4 semi-structured pruning N_c	D_c	E	α	β	δ
Llama-3 series	$\mathcal{L}_1$	0.02	5.94	0.14	-1.57	0.23	-1.08	0.29	0.05	5.86	-2.52	-1.68	0.08	-0.97	0.38	38.26	0.87	2.49	26.53	0.37	0.05
	$\mathcal{L}_2$	0.64	7.99	0.73	2.45	0.47	-1.08	-	0.00	3.53	0.20	-21.89	0.25	-0.97	-	0.53	0.89	2.19	0.92	0.41	-
	$\mathcal{L}_3$	-	5.93	0.54	-	0.30	-1.06	0.15	-	3.87	0.53	-	0.34	-0.98	-0.05	-	0.80	2.5	-	0.22	0.07
Qwen-2.5 series	$\mathcal{L}_1$	0.01	4.32	0.20	-3.73	0.21	-1.17	0.22	-0.58	7.01	-1.89	0.38	0.10	-1.28	0.16	1.85	0.93	0.32	-0.12	0.10	0.17
	$\mathcal{L}_2$	0.02	4.78	0.62	4.08	0.32	-1.17	-	-0.01	5.84	-0.65	-1.58	0.18	-1.28	-	1.52	0.75	0.92	0.15	0.18	-
	$\mathcal{L}_3$	-	4.77	0.87	-	0.36	-1.15	0.16	-	5.95	-0.91	-	0.16	-1.28	0.02	-	0.76	2.41	-	0.16	0.09

Table 5: Parameter values of fitted parameterizations for P^2 Law fitting.

LLM	Parameterizations	Depth pruning	Width pruning	2:4 semi-structured pruning
Llama-3 series	$\mathcal{L}_1$	✓	✓	✓
	$\mathcal{L}_2$	×	×	×
	$\mathcal{L}_3$	✓	×	✓
Qwen-2.5 series	$\mathcal{L}_1$	✓	✓	✓
	$\mathcal{L}_2$	×	×	×
	$\mathcal{L}_3$	✓	✓	✓

Table 6: Compliance of $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ with Condition 2.

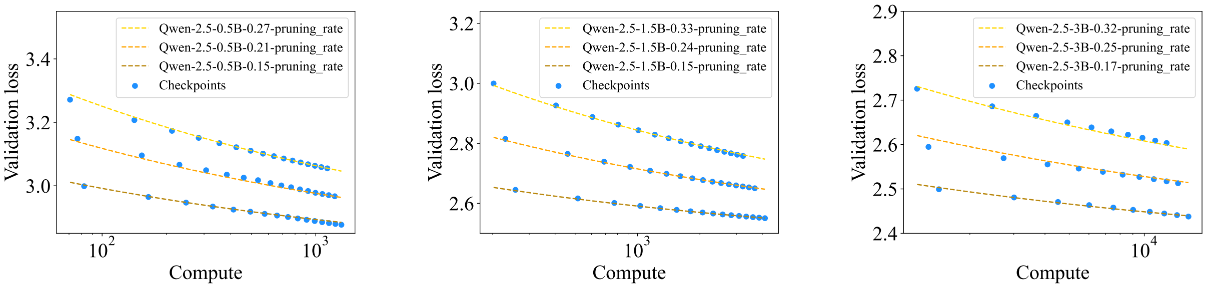


(a) Loss curves derived by P^2 Law and the actual checkpoints of Llama-3.2-1B pruned by width pruning.

(b) Loss curves derived by P^2 Law and the actual checkpoints of Llama-3.2-3B pruned by width pruning.

(c) Loss curves derived by P^2 Law and the actual checkpoints of Llama-3.1-8B pruned by width pruning.

Figure 13: Loss curves derived by P^2 Law and the actual checkpoints of Llama-3 series models pruned by width pruning.

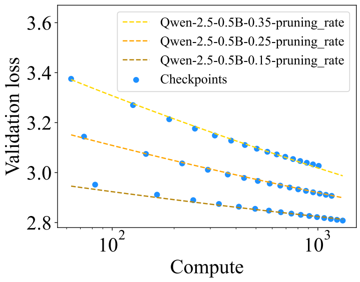


(a) Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5-0.5B pruned by depth pruning.

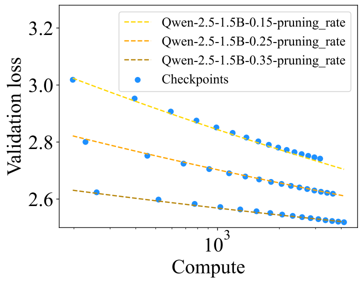
(b) Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5-1.5B pruned by depth pruning.

(c) Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5-3B pruned by depth pruning.

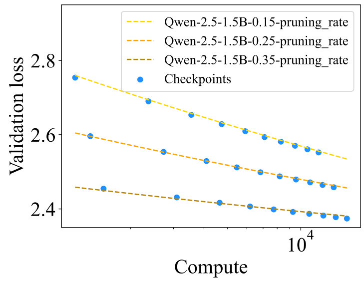
Figure 14: Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5 series models pruned by depth pruning.



(a) Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5-0.5B pruned by width pruning.

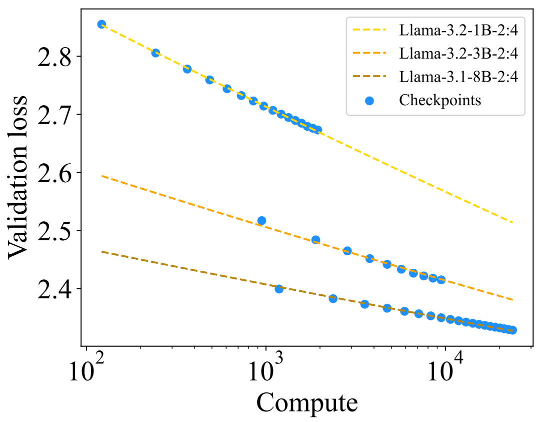


(b) Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5-1.5B pruned by width pruning.

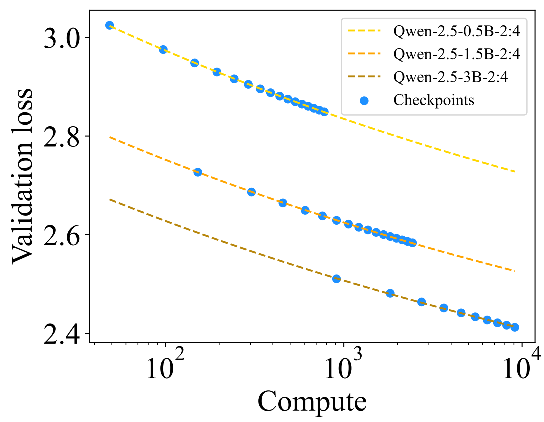


(c) Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5-3B pruned by width pruning.

Figure 15: Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5 series models pruned by width pruning.

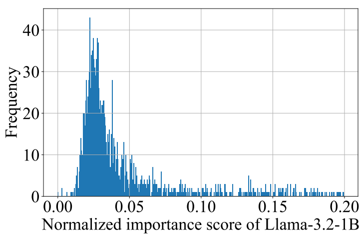


(a) Loss curves derived by P^2 Law and the actual checkpoints of Llama-3 series models pruned by 2:4 semi-structured pruning.

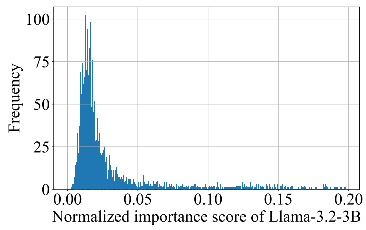


(b) Loss curves derived by P^2 Law and the actual checkpoints of Qwen-2.5 series models pruned by 2:4 semi-structured pruning.

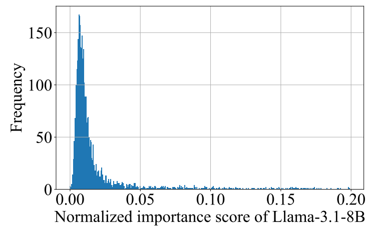
Figure 16: Loss curves derived by P^2 Law and the actual checkpoints of Llama-3 seires and Qwen-2.5 series models pruned by 2:4 semi-structured pruning.



(a) Histogram of the normalized importance scores for the embedding channels of Llama-3.2-1B.

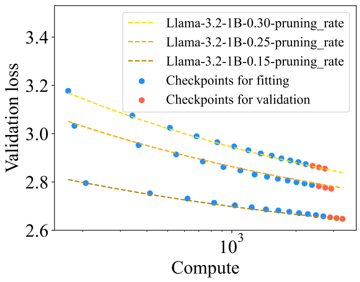


(b) Histogram of the normalized importance scores for the embedding channels of Llama-3.2-3B.

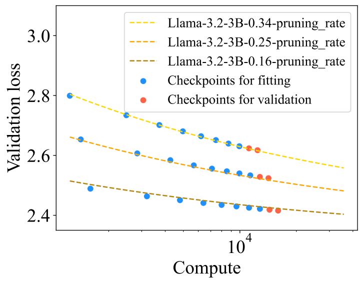


(c) Histogram of the normalized importance scores for the embedding channels of Llama-3.1-8B.

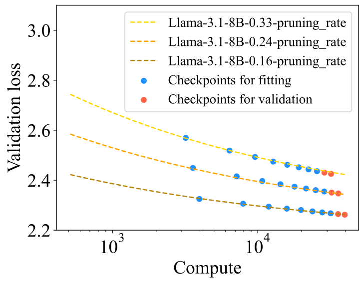
Figure 17: Histogram of the normalized importance scores for the embedding channels of Llama-3 series models.



(a) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Llama-3.2-1B pruned by depth pruning)

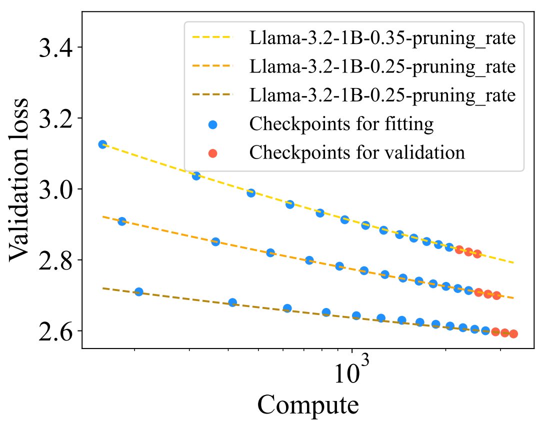


(b) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Llama-3.2-3B pruned by depth pruning)

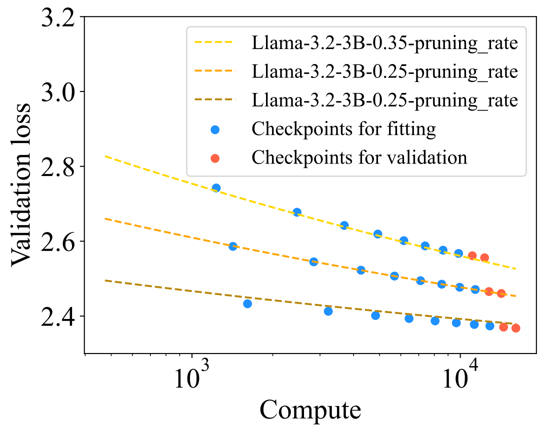


(c) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Llama-3.1-8B pruned by depth pruning)

Figure 18: Generalization of the P^2 Law for Llama-3 series models pruned by depth pruning on dataset size.

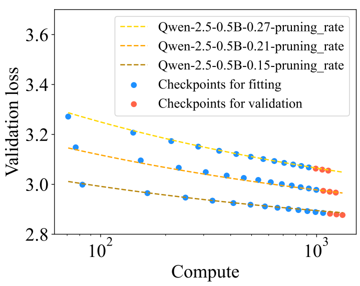


(a) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Llama-3.2-1B pruned by width pruning)

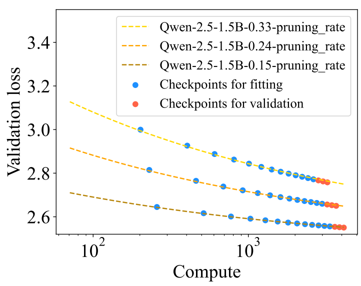


(b) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Llama-3.2-3B pruned by width pruning)

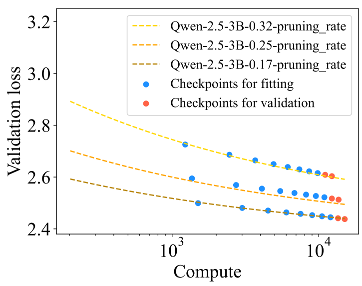
Figure 19: Generalization of the P^2 Law for Llama-3 series models pruned by width pruning on dataset size.



(a) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Qwen-2.5-0.5B pruned by depth pruning)

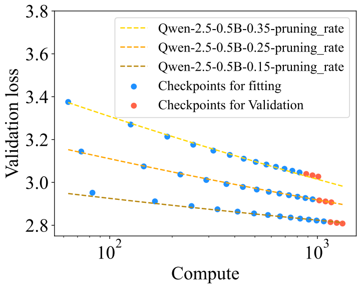


(b) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Qwen-2.5-1.5B pruned by depth pruning)

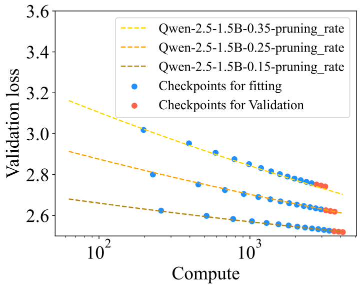


(c) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Qwen-2.5-3B pruned by depth pruning)

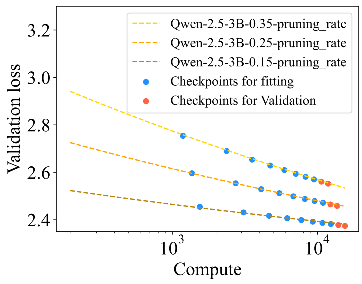
Figure 20: Generalization of the P^2 Law for Qwen-2.5 series models pruned by depth pruning on dataset size.



(a) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Qwen-2.5-0.5B pruned by width pruning)

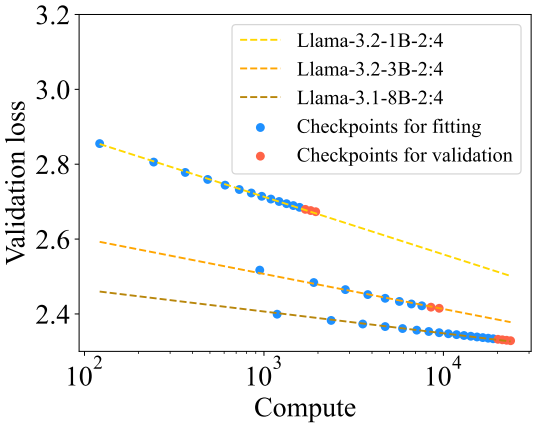


(b) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Qwen-2.5-1.5B pruned by width pruning)

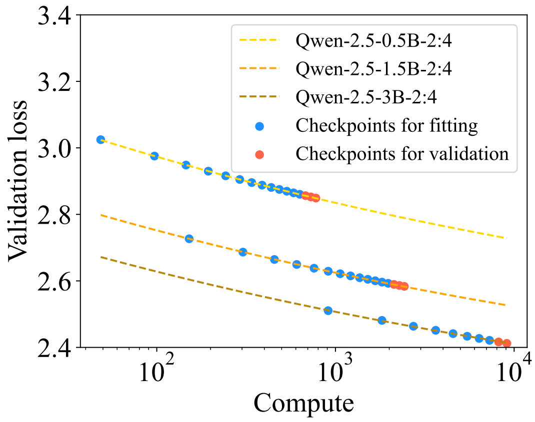


(c) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Qwen-2.5-3B pruned by width pruning)

Figure 21: Generalization of the P^2 Law for Qwen-2.5 series models pruned by width pruning on dataset size.

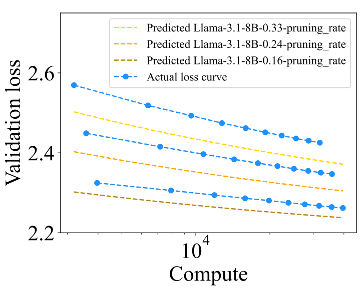


(a) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Llama-3 series models pruned by 2:4 semi-structured pruning)

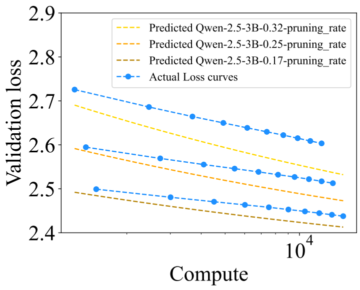


(b) Loss curves fitted with the P^2 Law using the first 80% of checkpoints; the remaining 20% are used for validation. (Qwen-2.5 series models pruned by 2:4 semi-structured pruning)

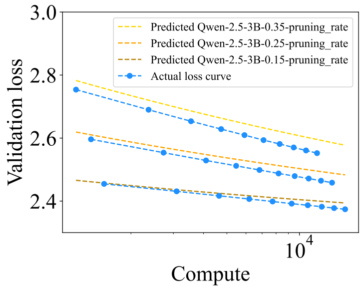
Figure 22: Generalization of the P^2 Law for models pruned by 2:4 semi-structured pruning on dataset size.



(a) P^2 Law is fitted using checkpoints from smaller LLMs and used to predict the loss curves of larger LLMs. (Llama-3 series models pruned by depth pruning)

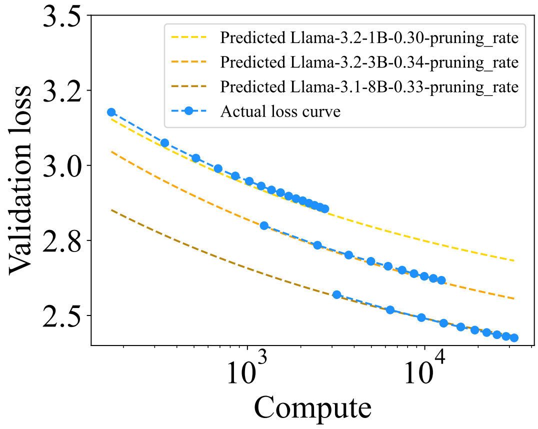


(b) P^2 Law is fitted using checkpoints from smaller LLMs and used to predict the loss curves of larger LLMs. (Qwen-2.5 series models pruned by depth pruning)

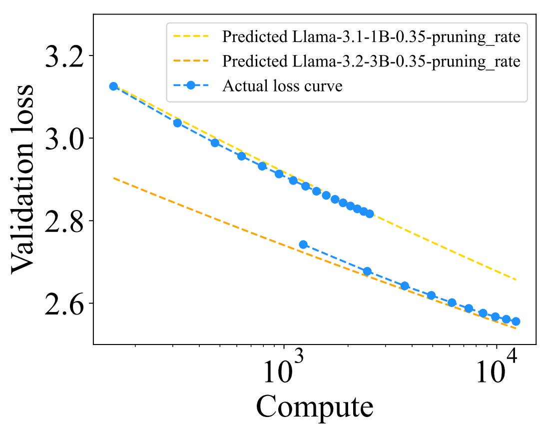


(c) P^2 Law is fitted using checkpoints from smaller LLMs and used to predict the loss curves of larger LLMs. (Qwen-2.5 series models pruned by width pruning)

Figure 23: Generalization of the P^2 Law on model size.

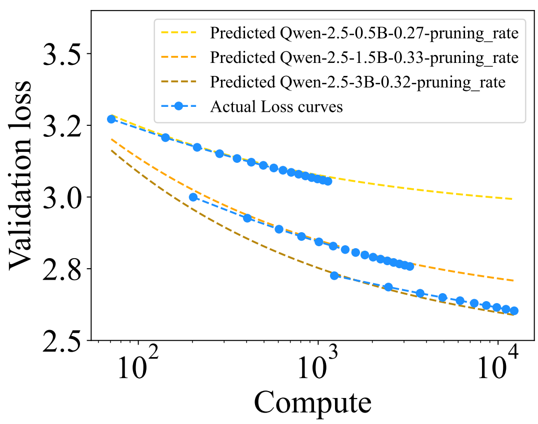


(a) P^2 Law is fitted using checkpoints from smaller pruning rates and used to predict the loss curves of larger ones. (Llama-3 series models pruned by depth pruning)

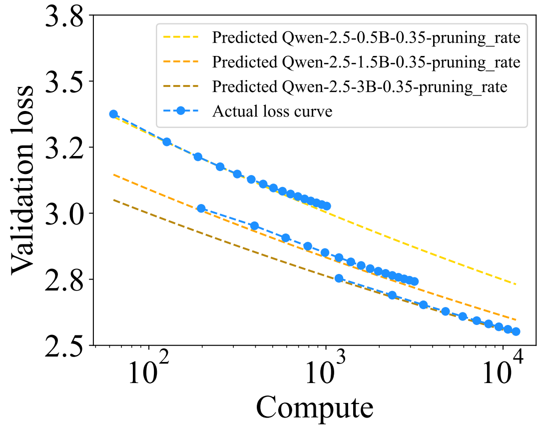


(b) P^2 Law is fitted using checkpoints from smaller pruning rates and used to predict the loss curves of larger ones. (Llama-3 series models pruned by width pruning)

Figure 24: Generalization of the P^2 Law for Llama-3 series models on pruning rate.



(a) P^2 Law is fitted using checkpoints from smaller pruning rates and used to predict the loss curves of larger ones. (Qwen-2.5 series models pruned by depth pruning)



(b) P^2 Law is fitted using checkpoints from smaller pruning rates and used to predict the loss curves of larger ones. (Qwen-2.5 series models pruned by width pruning)

Figure 25: Generalization of the P^2 Law for Qwen-2.5 series models on pruning rate.