A Experimental Setup Details

A.1 Baseline Model

In our ablation studies in Section 3, we used Llama3.1-8B Instruct[1](Dubey et al., 2024) as the baseline model for all experiments. This model comprises a total of 32 decoder layers, pretrained on a diverse array of instruction-following datasets. This model was chosen for its strong generalization performance across a wide range of natural language processing (NLP) tasks, making it an ideal candidate for studying the impact of structured pruning and fine-tuning. The 8B model size strikes a balance between computational efficiency and model quality, providing a robust foundation for the experiments in this study. Hence, served as the starting point for our structured layer pruning ablations and experiments in Sections 3 and 4, respectively.

In Section 4, to further understand the efficacy of our methodology on other LLMs, we also applied it to Mistral-7B Instruct v0.3 [2], an instruct fine-tuned version of Mistral-7B-v0.3. This most recent version of Mistral-7B (Jiang et al., 2023), compared to Mistral-7B-v0.2, includes an extended vocabulary of 32,768 tokens, supports a v3 tokenizer, and enables function calling.

A.2 Structured Layer Pruning

In this study, we focus on structured layer pruning of decoder layers to reduce the computational footprint of the LLM while maintaining its quality. Specifically, we prune in block sizes of {2, 4, 6, 8, 10} layers, corresponding to {30, 28, 26, 24, 22} decoder layers, respectively. Each block size reduction effectively removes a group of layers from the original architecture, creating progressively smaller models. These pruned models allow us to systematically evaluate the trade-offs between computational efficiency (fewer layers) and the accuracy on various downstream tasks. By examining multiple block sizes, we analyze how varying degrees of pruning impact model quality, especially in the context of *self-data distilled fine-tuning*, our proposed methodology.

A.3 Calibration Dataset for Structured Layer Pruning

In structured pruning, selecting a suitable calibration dataset is critical for effectively identifying and removing redundant layers without sacrificing model quality. This study examines the impact of different calibration datasets: C4, RedPajama, and SlimPajama on computing the angular cosine distance block importance metric for Llama3.1-8B Instruct, as shown in Table 7. We found that all three datasets produced similar results across various prune block sizes, with comparable layers identified for removal. Given the consistency of results, we opted to use Redpajama as the calibration dataset in subsequent experiments due to its representative performance and alignment with our goals for efficient model pruning. Recent studies around the time of this submission have explored the nuanced role of calibration data in pruning large language models (Ji et al., 2024). While our analysis focuses on the practical selection of calibration datasets, a deeper investigation into calibration dataset characteristics and their influence on pruning decisions remains an open question for future work.

Table 7. Ablation study on the choice of calibration dataset for computing the angular cosine distance block importance metric on Llama3.1-8B Instruct. For calibration, we use a subset of 128 samples at a maximum sequence length (MSL) of 4096 from each dataset. The datasets C4, Redpajama, and Slimpajama produced similar pruning results across various block sizes, with comparable layers removed. Based on these results, the Redpajama dataset was selected for further evaluations due to its representative performance.

Block Size	Removed Layers	Dataset	Score (avg dist)
2	24-25	C4	0.145
	23-24	Redpajama	0.168
	24-25	Slimpajama	0.153
4	24-27	C4	0.197
	23-26	Redpajama	0.222
	23-26	Slimpajama	0.205
6	22-27	C4	0.241
	22-27	Redpajama	0.270
	23-28	Slimpajama	0.249
8	20-27	C4	0.282
	20-27	Redpajama	0.293
	20-27	Slimpajama	0.289

A.4 Fine-tuning Datasets

The following datasets were used for ablation studies and fine-tuning experiments, representing a range of open-domain conversation, instruction-following, reasoning, and mathematical tasks:

- **Dolly 15k** (Conover et al., 2023) The Dolly dataset is an open-source collection of 15,000 instruction-following records generated by thousands of Databricks employees. It covers a wide range of behavioral categories, as outlined in InstructGPT(Ouyang et al., 2022), including brainstorming, classification, closed question answering (QA), generation, information extraction, open QA, and summarization. Dolly is designed to provide a benchmark for general-purpose

[1] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

instruction-following models, emphasizing diverse task types and behavioral categories.

- **GSM8k** (Cobbe et al., 2021) The GSM8k dataset is a collection of 8,000 high-quality grade-school-level math word problems, developed by OpenAI. Each problem is designed to assess a model's ability to perform multi-step reasoning and problem-solving, making it an essential benchmark for evaluating arithmetic, algebraic, and logical reasoning abilities in large models. Fine-tuning on GSM8k highlights the model's capacity for mathematical reasoning, a key focus of our ablation studies.

- **Alpaca Cleaned**[3] (Taori et al., 2023) The Alpaca Cleaned dataset is a cleaned version of the original Stanford Alpaca dataset, containing 51,760 instruction-following examples. It addresses several issues present in the original release, such as hallucinations, incorrect instructions, and output inconsistencies. This dataset provides high-quality general instruction-following tasks, spanning text generation, summarization, reasoning, and more. The cleaned version offers improved consistency and accuracy, making it ideal for fine-tuning large models in real-world instruction-following tasks.

- **OpenMathInstruct** (Toshniwal et al., 2024) The OpenMathInstruct-1 dataset is specifically designed for fine-tuning language models on mathematical instruction tasks. It contains 1.8 million problem-solution pairs, generated using Mixtral-8x7B (Jiang et al., 2024). The problem sets are drawn from well-established mathematical benchmarks, including the GSM8K and MATH (Hendrycks et al., 2021b) datasets, ensuring a diverse and challenging range of mathematical reasoning tasks. Solutions are generated synthetically by allowing the Mixtral model to leverage a combination of natural language reasoning and executable Python code, which allows for both symbolic computation and procedural solutions. This combination of text and code execution makes the dataset particularly suited for training models to handle complex reasoning, problem-solving, and algebraic tasks.

A.4.1 Data Sampling and Experimental Consistency

To maintain consistency across ablation studies, we fixed the dataset size at 8,000 samples for GSM8k, Alpaca, and OpenMathInstruct, aligning them with the standard GSM8k dataset size. However, the Dolly dataset retained its default size of 15,000 samples to preserve the integrity of this benchmark. To evaluate the impact of dataset size on self-data distillation, we extended the sample sizes for some experiments, using the full 50,000 samples from Alpaca Cleaned and randomly sampling 50,000 training samples from OpenMathInstruct. This allowed us to control for the effects of larger datasets, providing insights into how dataset size influences generalization and model retention following pruning.

A.5 Fine-tuning Pruned Models

For fine-tuning, we employed Low-Rank Adaptation (LoRA) (Hu et al., 2022), as it provides an efficient approach to training while preserving the pretrained model's capacity. Although full fine-tuning is feasible, we focused on LoRA fine-tuning in this study, leaving full parameter fine-tuning for future work. We conducted a comprehensive grid search on an 8k-sample version of the OpenMathInstruct dataset to identify the most effective hyperparameters for LoRA-based fine-tuning. The search was performed across a range of values to ensure optimal performance. We explored different *rank sizes* $\in \{4, 8, 16, 32\}$, aiming to balance model capacity and parameter efficiency. For the *number of epochs*, we tested values ranging $\in \{3, 5, 7, 10\}$, ensuring that the models were fine-tuned enough to converge without overfitting. The *learning rate* was swept across five values $\{2\times10^{-5}, 4\times10^{-5}, 6\times10^{-5}, 8\times10^{-5}, 1\times10^{-4}\}$. Finally, we tested *batch sizes* $\in \{8, 16, 32, 64, 128\}$ to determine the optimal balance between training stability and computational efficiency.

Through this grid search, the optimal configuration was identified as a *rank size* = 8, *epochs* = 5, a *batch size* = 64, and *learning rate* = 1×10^{-4}. These hyperparameters were used consistently across all fine-tuning experiments (i.e., both standard supervised fine-tuning and self-data distilled fine-tuning) in this study to ensure a fair comparison of the models and their quality post-pruning. We conduct our model training using `LLaMA-Factory v0.8.3`[4], a versatile framework designed for large-scale language model training and fine-tuning. This version offers extensive support for efficient parallelism, optimized memory usage, and integration with popular datasets, making it ideal for large model fine-tuning tasks such as those performed in this study.

A.5.1 Computational Resources

Fine-tuning and evaluations were conducted on Nvidia H100 GPUs. For experiments involving larger self-data distillation datasets, we utilized Cerebras CS-3 Inference (Thangarasa et al., 2024b), which achieves output generation speeds exceeding 1800 tokens per second. The CS-3 system was particularly useful for generating large-scale

[3] https://huggingface.co/datasets/yahma/alpaca-cleaned

[4] https://github.com/hiyouga/LLaMA-Factory/releases/tag/v0.8.3

self-distilled datasets. However, for smaller datasets (e.g., up to 15k samples), the H100 GPUs were sufficient for both fine-tuning and generation.

B Extended Results on Fine-tuning Ablations

In this section, we provide extended results from our fine-tuning ablation study to further clarify the impact of dataset choice on self-data distillation efficacy in pruned Llama3.1-8B Instruct models. As detailed in the Section 3, we observed that self-data distillation consistently outperformed SFT across various datasets. Table 8 shows that the largest gains were achieved using the 50k-sample OpenMathInstruct dataset, particularly at medium and large pruning block sizes (e.g., block size 6). At this configuration, self-data distillation was able to recover 95.96% of the baseline model quality, which is a significant improvement compared to other datasets and fine-tuning methods. This result highlights the robustness of the self-data distillation process, especially in recovering quality post-pruning on reasoning-heavy tasks like those in GSM8k, ARC-C, and MMLU.

Moreover, the recovery rates exhibited a clear trend where, larger datasets such as the 50k OpenMathInstruct consistently led to higher quality retention, especially when combined with more aggressive pruning. This suggests that the dataset's ability to approximate the model's original data distribution is critical for maintaining generalization capabilities after pruning. In contrast, smaller datasets like Alpaca or Dolly showed comparatively lower recovery rates, which further confirms the importance of dataset scale in the distillation process. Our results suggest that larger datasets are crucial for mitigating quality degradation in pruned models, with the 50k OpenMathInstruct dataset emerging as the most effective in retaining and enhancing model quality across block sizes, particularly in challenging reasoning tasks.

C Experimental Setup for Understanding Catastrophic Forgetting

To understand the impact of distribution shift on catastrophic forgetting, we conducted experiments using the baseline model (i.e., Llama3.1-8B Instruct) and its pruned variants fine-tuned with both supervised fine-tuning (SFT) and self-data distilled fine-tuning (Self-Data FT). Specifically, we pruned 6 decoder layers, reducing the model from 32 to 26 layers, and evaluated the models on the GSM8k dataset. For these experiments, we generated model responses using the baseline and pruned variants on the GSM8k dataset to capture how the distribution shift affects reasoning tasks post-pruning. Following Yang et al. (2024), to quantify the distribution shift, we employed Sentence-BERT (Reimers & Gurevych, 2019) to derive sentence embeddings from the model-generated responses. Then, similar to the method proposed by Zhang et al. (2023), we calculated the cosine similarity between the sentence embeddings of the pruned models and those generated by the original Llama3.1-8B Instruct model.

A lower cosine similarity score indicates a greater distribution shift, suggesting a higher risk of catastrophic forgetting. Conversely, higher similarity scores indicate better preservation of the original model's knowledge and a lower risk of forgetting. These metrics allowed us to assess the extent to which SFT and Self-Data FT preserved the learned distribution of the base model, with the latter showing superior performance in mitigating forgetting, as detailed in our ablations in Section 3.

D Model Merging Self-Data Distilled Models

We employ the Spherical Linear Interpolation (SLERP) method for merging pruned models, which ensures smooth, geometrically consistent interpolation between two pruned model parameter vectors. SLERP operates within the unit sphere's geometry, contrasting with traditional linear interpolation that may destabilize or yield suboptimal parameter combinations by ignoring the geometric properties of the high-dimensional parameter space. SLERP preserves model integrity during interpolation, leading to more stable and consistent outcomes.

Given two pruned model parameter vectors, $\boldsymbol{\theta}'_0$ and $\boldsymbol{\theta}'_1$, corresponding to pruned models M'_0 (fine-tuned on OpenMathInstruct) and M'_1 (fine-tuned on Alpaca), SLERP generates an interpolated parameter vector $\boldsymbol{\theta}'_t$ for any interpolation factor $t \in [0, 1]$. When $t = 0$, the parameters of the OpenMathInstruct fine-tuned model $\boldsymbol{\theta}'_0$ are retrieved, and when $t = 1$, the parameters of the Alpaca fine-tuned model $\boldsymbol{\theta}'_1$ are retrieved.

Normalization to Unit Sphere The first step in SLERP is to normalize both pruned model parameter vectors to lie on the unit sphere,

$$\hat{\boldsymbol{\theta}}'_0 = \frac{\boldsymbol{\theta}'_0}{\|\boldsymbol{\theta}'_0\|}, \quad \hat{\boldsymbol{\theta}}'_1 = \frac{\boldsymbol{\theta}'_1}{\|\boldsymbol{\theta}'_1\|}.$$

This normalization ensures that both parameter vectors have unit norms, placing them on the surface of the unit sphere in the parameter space. Next, we compute the angle θ_{angle} between the normalized pruned model vectors $\hat{\boldsymbol{\theta}}'_0$ and $\hat{\boldsymbol{\theta}}'_1$. This angle is computed using the dot product, $\cos(\theta_{\text{angle}}) = \hat{\boldsymbol{\theta}}'_0 \cdot \hat{\boldsymbol{\theta}}'_1$, and the actual angle is given by $\theta_{\text{angle}} = \arccos(\cos(\theta_{\text{angle}}))$. This angle represents the angular separation between the two pruned models' parameter

vectors on the unit sphere.

Spherical Interpolation With the angle θ_{angle} determined, SLERP performs spherical interpolation along the great circle connecting $\hat{\boldsymbol{\theta}}_0'$ and $\hat{\boldsymbol{\theta}}_1'$. The interpolated parameter vector $\boldsymbol{\theta}_t'$ is computed as,

$$\boldsymbol{\theta}_t' = \frac{\sin((1-t)\theta_{\text{angle}})}{\sin(\theta_{\text{angle}})} \cdot \hat{\boldsymbol{\theta}}_0' + \frac{\sin(t\theta_{\text{angle}})}{\sin(\theta_{\text{angle}})} \cdot \hat{\boldsymbol{\theta}}_1'.$$

This formula ensures that the interpolation remains on the surface of the unit sphere, respecting the geometric structure of the parameter space. The interpolation factor t controls the contribution from each pruned model, when $t = 0$, $\boldsymbol{\theta}_t' = \hat{\boldsymbol{\theta}}_0'$ (i.e., OpenMathInstruct fine-tuned model), and when $t = 1$, $\boldsymbol{\theta}_t' = \hat{\boldsymbol{\theta}}_1'$ (i.e., Alpaca fine-tuned model). The intermediate values of t produce a smooth, spherical blend of the two pruned models.

D.1 Geometric Consistency and Application

By operating within the unit sphere, SLERP respects the *Riemannian geometry* of the high-dimensional parameter space, ensuring a smooth transition between the two pruned models. Traditional linear interpolation in such spaces can distort the relationships between parameters, leading to sub-optimal combinations and degraded model performance. In contrast, SLERP maintains geometric consistency, ensuring that the interpolation follows a natural path on the unit sphere.

Merging the pruned OpenMathInstruct and Alpaca models using SLERP combines the unique strengths of both models. For instance, OpenMathInstruct's emphasis on mathematical reasoning and logical structure complements Alpaca's broader instruction-following capabilities. By adjusting the interpolation factor t, the merged model can balance these capabilities, resulting in a versatile and robust model for a range of downstream tasks. We use Arcee.ai's `mergekit`[5] for efficiently merging model checkpoints.

[5] https://github.com/arcee-ai/mergekit

Table 8. **Model quality results for pruned Llama3.1-8B Instruct models across various pruning block sizes and fine-tuning strategies.** This table reports the quality of different fine-tuning methods (No Fine-tuning, Standard Fine-tuning (SFT), and Self-Data Distillation) on various datasets, with average accuracy across ARC-C, GSM8k, and MMLU tasks. The "Avg. Recovery" column shows the percentage of model quality recovered relative to the unpruned baseline. The table highlights that the self-data distillation strategy consistently yields superior recovery rates, particularly with the 50k-sample OpenMathInstruct dataset. For instance, at a pruning block size of 6, the self-data distilled OpenMathInstruct model retains 95.96% of the original unpruned Llama3.1-8B Instruct (i.e., 32 layers) model's quality, the highest recovery observed among all datasets and fine-tuning methods.

Prune Block Size	Model Savings	Fine-tuning Method	Dataset	ARC-C (25-shot)	GSM8k (5-shot)	MMLU (5-shot)	Avg. Score	Avg. Recovery
Baseline	-	No FT		58.70	63.15	67.40	63.08	100.00%
2	5.43% (7.59B)	No FT		55.20	67.79	56.18	59.72	94.67%
		SFT	GSM8k	**58.45**	56.25	65.22	59.97	95.07%
		Self-Data Distillation	GSM8k	57.34	64.44	66.60	62.79	99.54%
		SFT	Dolly	55.67	61.64	65.71	61.01	96.71%
		Self-Data Distillation	Dolly	56.48	62.24	66.46	61.73	97.87%
		SFT	Alpaca (50k)	56.61	63.19	65.60	61.80	97.98%
		Self-Data Distillation	Alpaca (50k)	56.91	68.60	**66.50**	63.34	100.41%
		SFT	OpenMathInstruct (50k)	52.91	44.95	60.93	52.93	83.88%
		Self-Data Distillation	OpenMathInstruct (50k)	56.43	**69.97**	66.17	**64.19**	**101.76%**
4	10.86% (7.16B)	No FT		55.20	56.18	**67.79**	59.72	94.67%
		SFT	GSM8k	54.27	43.47	65.40	54.38	86.22%
		Self-Data Distillation	GSM8k	55.20	62.55	66.68	61.48	97.49%
		SFT	Dolly	54.78	59.36	63.65	59.26	93.95%
		Self-Data Distillation	Dolly	51.71	55.96	65.40	57.69	91.44%
		SFT	Alpaca (50k)	56.05	54.40	65.34	58.60	92.89%
		Self-Data Distillation	Alpaca (50k)	**57.27**	66.20	66.24	**63.24**	**100.26%**
		SFT	OpenMathInstruct (50k)	51.62	44.35	61.65	52.54	83.30%
		Self-Data Distillation	OpenMathInstruct (50k)	53.93	**69.44**	65.34	62.24	98.66%
6	16.30% (6.72B)	No FT		49.49	0.00	**67.42**	48.93	70.50%
		SFT	GSM8k	46.67	62.09	64.67	57.81	91.63%
		Self-Data Distillation	GSM8k	51.45	60.05	66.28	59.93	95.02%
		SFT	Dolly	47.18	33.89	65.33	48.13	76.31%
		Self-Data Distillation	Dolly	51.96	50.95	62.56	55.82	88.47%
		SFT	Alpaca (50k)	**54.62**	56.11	64.39	58.37	92.53%
		Self-Data Distillation	Alpaca (50k)	53.80	59.15	66.29	59.75	94.71%
		SFT	OpenMathInstruct (50k)	46.93	43.82	59.98	50.91	80.71%
		Self-Data Distillation	OpenMathInstruct (50k)	50.00	**66.64**	64.96	**60.53**	**95.96%**
8	21.73% (6.29B)	No FT		44.71	0.00	**65.57**	36.76	58.27%
		SFT	GSM8k	44.79	50.86	64.38	53.34	84.56%
		Self-Data Distillation	GSM8k	46.16	50.11	65.53	53.93	85.50%
		SFT	Dolly	39.33	15.39	57.44	37.39	59.27%
		Self-Data Distillation	Dolly	46.50	28.73	62.93	46.05	73.00%
		SFT	Alpaca (50k)	**49.15**	39.59	64.81	51.85	82.19%
		Self-Data Distillation	Alpaca (50k)	48.09	42.77	65.20	52.69	83.51%
		SFT	OpenMathInstruct (50k)	42.15	29.64	60.65	44.81	71.05%
		Self-Data Distillation	OpenMathInstruct (50k)	46.67	**57.70**	64.87	**56.41**	**89.44%**
10	27.16% (5.85B)	No FT		37.46	0.00	64.09	33.85	53.65%
		SFT	GSM8k	39.85	37.45	61.47	46.92	74.36%
		Self-Data Distillation	GSM8k	41.55	37.47	62.33	47.12	74.70%
		SFT	Dolly	38.40	0.76	46.94	28.70	45.51%
		Self-Data Distillation	Dolly	43.60	12.28	62.47	39.45	62.53%
		SFT	Alpaca (50k)	**45.22**	17.51	63.72	42.82	67.88%
		Self-Data Distillation	Alpaca (50k)	44.91	20.05	**64.73**	43.90	69.56%
		SFT	OpenMathInstruct (50k)	39.33	15.39	57.44	37.39	59.25%
		Self-Data Distillation	OpenMathInstruct (50k)	40.88	**44.58**	64.54	**50.33**	**79.79%**