On Pruning State-Space LLMs

Tamer Ghattas **Michael Hassid** **Roy Schwartz**

The Hebrew University of Jerusalem
{tamer.ghattas, michael.hassid, roy.schwartz1}@mail.huji.ac.il

Abstract

Recent work proposed state-space models (SSMs) as an efficient alternative to transformer-based LLMs. Can these models be pruned to further reduce their computation costs? We adapt several pruning methods to the SSM structure, and apply them to four SSM-based LLMs across multiple tasks. We find that such models are quite robust to some pruning methods (e.g., WANDA), while using other methods lead to fast performance degradation.[1]

1 Introduction

Selective-State Space models (SSMs, Gu et al., 2022) have recently gained attention as an appealing alternative to transformers (Gu and Dao, 2024; Dao and Gu, 2024). SSMs leverage both selective memory capabilities and RNN (Elman, 1990) properties, showing comparable results against transformer-based peers. However, SSM-based LLMs are still parameter-heavy, which raises the question of how well they can be compressed.

In this work, we focus on one of the key compression methods—*pruning* (LeCun et al., 1989). Modern LLM pruning methods have been developed and tested mostly for transformer components such as self-attention and feed-forward. Here we study how well SSM-based LLMs can be pruned.

We adapt several structured pruning methods to SSMs, e.g., pruning different SSM heads using different criteria, or merging existing heads (Fig. 1). We apply these methods to four SSM-based LLMs, along with WANDA (Sun et al., 2024), an unstructured pruning method that requires no adaptation. We compare all methods across six different tasks.

Our results show that all models are robust to unstructured pruning with WANDA, even when reducing up to 50% of the SSM parameter count. We also observe that pruning SSM states leads to small degradation in almost all cases. In contrast, pruning the SSM heads leads to a sharp drop in performance in all cases. We also show that the output projection in SSM-based LLMs is much more sensitive to pruning than the input projection. Our results hint that SSM-based LLMs can indeed be made more efficient, but the choice of pruning method has a large effect on the pruning quality.

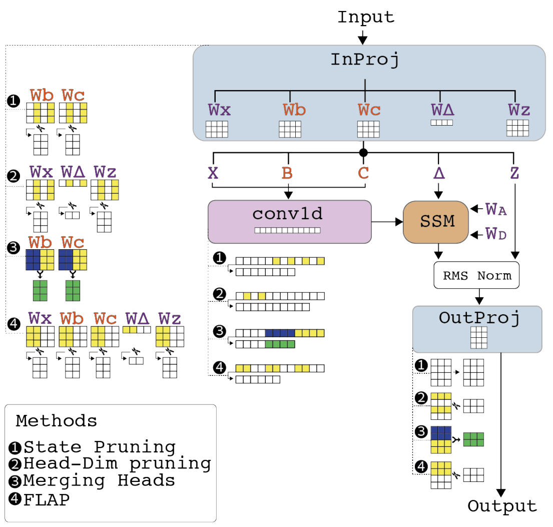


Figure 1: Pruning SSM-based LLMs. **Right**: the Mamba SSM block: the input is linearly projected using five projection matrices (W_Z,W_X,W_B,W_C,W_Δ), to be used in later parts of the block. Every SSM head is represented using two vectors (two rows for InProj and two columns for OutProj). **Left**: our different structure pruning methods. Each yellow cell represents a pruned element in the corresponding head. (1) *State pruning*: head extraction from W_B and W_C tensors then pruning the corresponding conv1d filters; (2) *Head dimension pruning*: head extraction from W_X, W_Z, W_A, W_D and W_Δ, and pruning the corresponding conv1d filters and OutProj rows; (3) *Head Merging*: mean-pooling every two BC-heads and all corresponding components; (4) *SSM-FLAP*: adapting FLAP to SSMs, which prunes whole heads on all InProj sub-components heads and their correspondingly conv1d and OutProj.

[1] www.github.com/schwartz-lab-NLP/SSM-Pruner

2 Background

In this section, we discuss recent SSM architectures and some of the best performing pruning methods commonly used with transformer-based models.

2.1 SSM Architectures

State Space Models (Gu et al., 2022) are a class of seq2seq models that represent inputs via hidden states evolving over time. Unlike attention-based architectures, SSMs leverage structured representations to achieve sub-quadratic complexity.

Recent work has shown that SSM-based LLMs can be trained and reach competitive performance to transformers-based LLMs. Mamba (Gu and Dao, 2024) uses a "selective" SSM variant that adaptively focuses on certain parts of the input at each time step. Mamba-2 (Dao and Gu, 2024) builds upon structured space duality framework, which bridges the gap between recurrent-style processing and attention-like operations. This enables the use of hardware optimizations made for attention. By producing SSM parameters in parallel and simplifying its core layer, Mamba-2 is 2–8× faster than its predecessor while maintaining performance on par with transformers across various benchmarks.

The SSM component in Mamba-2 is composed of several sub-components (Fig. 1). The input is first passed through a linear layer (InProj), composed of the following tensors: W_X,$W_Z \in \mathbb{R}^{D,H,P}$; W_B, $W_C \in \mathbb{R}^{D,H,N}$; $W_\Delta \in \mathbb{R}^{D,H}$. This results in five matrices: X, B, C, Δ, and Z.[2] Then, X, B, and C are passed through a 1-D convolution layer. Its output, along with W_Δ and two learned parameter matrices W_A, $W_D \in \mathbb{R}^{1,H}$, are passed to the SSD algorithm (Dao and Gu, 2024). Its output is then joint with W_Z and normalized using RMS Norm (Zhang and Sennrich, 2019), and projected back to D—the model dimension.

Multi-head patterns in SSM-based LLMs Similarly to group-query attention in transformers (GQA; Ainslie et al., 2023), SSM-based LLMs allow grouping some of the SSM subcomponents. The choice of which components to merge is referred to as the *multi-head pattern* (Dao and Gu, 2024). In contrast to the commonly used GQA pattern in transformers, different SSM-based LLMs use different patterns. For example, Mamba-2 (Dao and Gu, 2024) groups the W_B,W_C matrices, while Hybrid-Llama3-Mamba2-3B (HLM-3B; Wang et al., 2024) groups W_X and W_B.

[2]Where D is the model dimension, H is the number of heads, P is the head dimension, and N is the state dimension.

2.2 Pruning Methods

Pruning is frequently used to compress LLMs. Below we describe common pruning methods.

Unstructured pruning induces sparsity in the model weights. Such methods reduce model size, though it is hard to translate this sparsity to runtime gains. A common implementation of this approach is *magnitude pruning*, which prunes unimportant parameters based on their absolute values (Frankle and Carbin, 2019). A recent highly effective variant of magnitude pruning is WANDA (Sun et al., 2024), which also takes activations into account. Importantly, WANDA does not require fine-tuning, making it highly efficient for compressing LLMs.

Structured pruning removes entire sections of model weights rather than individual elements (Ma et al., 2023; Molchanov et al., 2019; Fan et al., 2019). One such method is FLAP (An et al., 2023), which prunes based on the fluctuation of each input channel. It determines whether the output feature map can be recovered after pruning a column from the weight matrix. FLAP avoids the need for retraining and requires only a single forward pass for both pruning and bias compensation. Finally, group-query attention (GQA), which merges distinct attention KV heads using mean-pooling, can also be thought of as a structured pruning method.

3 Pruning SSM-based LLMs

We aim to study the effect of different pruning methods on SSM-based LLMs. Below we describe how we adapt different pruning methods, developed for transformers, to SSMs. We note that within the Mamba-2 SSM component (Fig. 1), the layer dimension dictates the shapes for the matrices within the input projection (InProj), which comprise approximately 67% of all parameters in the SSM component. The output projection (OutProj) and the 1d convlution layer (conv1d) are roughly 32% and < 1%, respectively.

WANDA is an **unstructured pruning** method, which can be applied to SSM layers with minimal adjustments, as it originally operates on linear layers of any size without further restrictions.

We also adapt four **structural pruning methods** to SSMs.

State pruning Each head in the W_B and W_C matrices of Mamba-2 is composed of a $D \times N$ matrix. Therefore, we can prune the least important tensors according to a desired ratio. In our experiments, we use the second order Taylor approximation-based importance estimator (Molchanov et al., 2019), and then average this score per tensor in each head to estimate its importance. To do so, we perform model pass on 20 wikitext2 samples (Merity et al., 2016) and accumulate gradients to calculate importance. To preserve the dimensionality correctness within the rest of the flow, we prune the OutProj matrix and the conv1d layer weights accordingly.

Head Dimension Pruning Similarly, each head in the W_X tensor is composed of P matrices, and thus can be pruned along with W_Z in the same way.

Merging Heads Inspired by the grouping of KV heads in attention (Ainslie et al., 2023; Jin et al., 2024), we consider pruning SSMs by grouping the BC or XB heads, while maintaining the multi-head pattern, e.g., further merging the already grouped heads (see Sec. 2). To do so, we use mean pooling on consecutive heads. An exception to the pattern preservation policy is Mamba-2, as it has only a single BC head, so we merge its X heads.

SSM-FLAP We adapt FLAP to prune SSM-based LLMs by applying its calculated pruning masks to the submatrices of InProj and pruning the corresponding elements in the rest of the flow. To enable bias compensation, we add a bias term similar to how FLAP operates with attention. Since FLAP prunes a different number of heads per layer, in an MHA based model where the number of BC heads and X heads is the same, we prune both groups, while in a GVA based model, we exclude the BC heads and limit the pruning to round the number of kept X heads to the closest larger multiplier of BC heads, keeping the number of X heads dividable by the number of BC heads.

4 Experiments

Models There aren't many competitive SSM-based LLMs. We consider four main models in different sizes and base architectures: MAMBA-2-2.7B, which is configured with a multi-value attention head pattern; PHI-MAMBA-1.5B (Bick et al., 2024), which is distilled from PHI-1.5 (Li et al., 2023). It preserves the same MLP, embeddings and LM head layers and converts the attention layer to be an SSM component. This model is configured with grouped-value attention; Hybrid-Llama3-Mamba2-3B (HLM-3B; Wang et al., 2024) is a hybrid model of interleaving attention and SSM layers distilled from LLAMA-3.1-70B-INSTRUCT (Grattafiori et al., 2024) but initialized using LLAMA-3.2-3B (Grattafiori et al., 2024) weights. It converts only the attention layer but finetunes all model layers. Unlike the original architecture of Mamba-2, this model uses grouped query attention (GQA) since its initializing weights come from LLama, which is GQA based; Finally, we distill SMOL-MAMBA-1.9B, [3], a Mamba model from SMOL2-1.7B (Allal et al., 2025), using the MOHAWK method (Bick et al., 2024). We note that Mamba-2 is a pure SSM LLM. That is, it only contains SSM blocks. In contrast, the other three models contain interleaving SSM and FFN layers.

Experimental setup For WANDA, state, head, and SSM-FLAP pruning, we prune models by 25% and 50%. For merging heads, we merge 50% and 75% of the heads. In all cases we report the topline—the unpruned models. Pruned models are finetuned on wikitext2. We use LORA (Hu et al., 2021) targeting both the SSM and MLP layers. We use the KD loss with the teacher set as the original model pre-pruning.[4] For each setup, we also report the ratio of pruned parameters of the full SSM component. See App. B for more details.

Benchmarks We use the EleutherAI LM Harness[5] to experiment with lambada (Paperno et al., 2016), hellaswag (Zellers et al., 2019), piqa (Bisk et al., 2020), arc-easy (Clark et al., 2018), arc-challenge (Clark et al., 2018), and winogrande (Sakaguchi et al., 2019).

Results Table 1 shows our pruning results for all methods except head merging (shown in Tab. 2), averaged across tasks.[6] WANDA tends to preserve model quality across models, especially at moderate pruning (25%) in 3/4 models, and does not collapse even at 50%. The exception is Mamba-2-2.7B, which drops more quickly. This is somewhat expected, as in that case there are no FFN layers.[7]

[3]We release the model at `https://huggingface.co/schwartz-lab/Smol2-Mamba-1.9B`.

[4]Preliminary experiments show that it outperforms standard CE loss, see Tab. 4 in App. A.

[5]`www.github.com/EleutherAI/lm-evaluation-harness`

[6]See App. C for full results on all tasks.

[7]We also compare the effect of WANDA pruning on a similar transformer-based model (App. D, Tab. 11), observing a similar drop in performance for both.

Model	Ratio	WANDA	State			Head			SSM-FLAP	
		w/o FT	w/o FT	w/ FT	Comp.	w/o FT	w/ FT	Comp.	w/o FT	Comp.
Phi-Mamba-1.5B (SSM & MLPs)	Dense	0.59	0.59	N/A	0%	0.59	N/A	0%	0.59	0%
	25%	0.59	0.58	0.59	10%	0.40	0.54	25%	0.57	26%
	50%	0.57	0.54	0.58	20%	0.33	0.47	50%	0.49	51%
HLM-3B (SSM & MLPs)	Dense	0.64	0.64	N/A	0%	0.64	N/A	0%	0.64	0%
	25%	0.64	0.30	0.31	20%	0.30	0.32	5%	0.29	5%
	50%	0.63	0.29	0.30	41%	0.29	0.31	10%	0.29	10%
Smol-Mamba-1.9B (SSM & MLPs)	Dense	0.61	0.61	N/A	0%	0.61	N/A	0%	0.61	0%
	25%	0.60	0.59	0.60	13%	0.29	0.30	5%	0.51	26%
	50%	0.56	0.47	0.59	25%	0.28	0.30	10%	0.41	49%
Mamba-2-2.7B (only SSM)	Dense	0.60	0.60	N/A	0%	0.60	N/A	0%	0.60	0%
	25%	0.53	0.53	0.54	0.5%	0.29	0.30	24%	0.30	25%
	50%	0.33	0.47	0.48	1%	0.29	0.29	47%	0.30	49%

Table 1: Results for pruning SSM components in different ratios, with *Dense* being an unpruned baseline. The State, Head, and SSM-FLAP methods report their SSM component compression (Comp.) values, along with the average benchmark accuracy before (w/o FT) and after (w/ FT) fine-tuning. Results for WANDA and FLAP are only w/o FT, as that they do not require fine-tuning to work well in practice. "N/A" denotes that no fine-tuning is performed.

Model	Heads	Comp.	w/o FT	w/ FT
Phi-Mamba-1.5B (SSM & MLPs)	32	0%	0.59	N/A
	16	20%	0.34	0.54
	8	30%	0.31	0.48
HLM-3B (SSM & MLPs)	8	0%	0.64	N/A
	4	10%	0.30	0.31
	2	14%	0.29	0.30
Smol-Mamba-1.9B (SSM & MLPs)	32	0%	0.61	N/A
	16	25%	0.29	0.44
	8	38%	0.29	0.41
Mamba-2-2.7B (only SSM)	80	0%	0.60	N/A
	40	49%	0.29	0.29
	20	70%	0.28	0.28

Table 2: Merging heads pruning results. Heads is the number of heads retained, Comp. is the SSM component compression rate. The topline is the first row per model.

Our structured pruning approaches show larger variance across models. PHI-MAMBA-1.5B maintains reasonable performance across state, head and SSM-FLAP pruning, even at 50% ratios. SMOL-MAMBA-1.9B also performs well across most methods, except head pruning. In contrast, HLM-3B and MAMBA-2-2.7B suffer severe degradations in almost all cases, even at moderate pruning ratios (25%) and post-finetuning.

Analysis We study whether MAMBA-2-2.7B, the most sensitive to pruning, exhibits different behavior to pruning different components. To do so, we apply WANDA exclusively on InProj, OutProj, or both, and report perplexity on wikitext2. Our results (Fig. 2) show that pruning OutProj yields a drastic spike in perplexity, even at moderate pruning ratios (30–40%), while InProj can be pruned more aggressively without catastrophic effects. Importantly, this is despite InProj having more than twice the number of parameters of OutProj.

To estimate the advantage of structural pruning, we measure the decoding throughput (tokens/second) of Smol-Mamba-1.9B before and after applying FLAP in App. D, Tab. 12. Pruning 50% produces a speedup of 5–17%, while pruning 25% produces a speedup of 0–6%.

Aiming to find the breaking point of WANDA pruning, we sweep the pruning ratio 0–90% on Smol-Mamba-1.9B and report results on the same benchmarks in Tab. 3. The steepest drop occurs between 70% and 80%, where model performance drops substantially on several datasets, most notably LAMBADA.

Takeaways SSM-based LLMs seem robust to WANDA pruning. Among structured pruning methods, state pruning seems most effective, leading to small to negligible performance drop in 3/4 models. In contrast, all models crash when applying head pruning, even at moderate rates. The other two methods (SSM-FLAP and head merging) work for some models but not others.

5 Conclusions

We adapted different pruning methods for SSM-based LLMs. Our results show that such LLMs can be pruned successfully with unstructured methods (WANDA), or even structured ones (state-pruning) with little to no performance degradation

Ratio	ARC-C	ARC-E	HellaSwag	PIQA	Winogrande	LAMBADA	Average
0.00	0.430	0.750	0.510	0.770	0.630	0.570	**0.610**
0.25	0.420	0.750	0.500	0.770	0.630	0.560	**0.605**
0.50	0.370	0.720	0.470	0.760	0.590	0.420	**0.555**
0.60	0.379	0.658	0.621	0.748	0.580	0.370	**0.559**
0.70	0.328	0.587	0.555	0.717	0.526	0.195	**0.485**
0.80	0.260	0.446	0.421	0.648	0.500	0.027	**0.384**
0.90	0.239	0.320	0.280	0.547	0.491	0.000	**0.313**

Table 3: Higher sparsity exploration for Smol-Mamba-1.9B with WANDA. Pruning ratio sweep between 0–0.9 in steps of 0.1. Performance drops most significantly between 0.7 and 0.8.

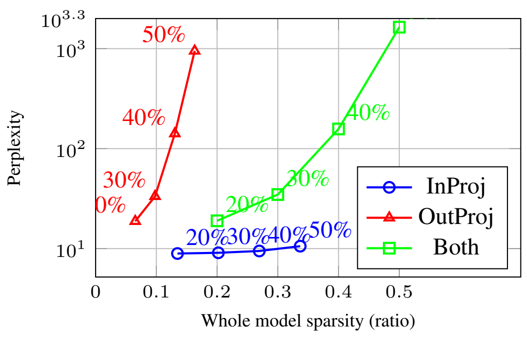


Figure 2: The effect of WANDA pruning ratios on different MAMBA-2-2.7B components. OutProj layer is substantially more sensitive to pruning than InProj.

in some cases. Our results hint at the potential of making SSM-based LLMs even more efficient.

Limitations

Drawing direct conclusions from our experiments is not straightforward. The four LLMs we experiment with differ, among other, in size, head pattern, training data, and structure. These choices are due to the scarcity of SSM-based LLMs, and the computational costs of training or distilling various models to control for specific variables.

In addition, the structure pruning methods we consider also differ in their effect on the models, as indicated by the different compression ratios in Tables 1 and 2. The challenges here stem from the different constraints imposed by the different methods. E.g., in state pruning we prune BC-heads, which cap the compression by the number of parameters W_B and W_C occupy in the Mamba-2 component, especially when the model configuration is GVA which translates to fewer BC-heads. Another example is when pruning Head-Dim we are obliged to prune large portions of OutProj to keep dimensionality correctness in the flow, leading to pruning potentially important parameters from it.

Despite these issues, we believe the signals we observe are valuable. For instance, the robustness of SSM-based LLMs to state pruning is demonstrated by the modest performance drop for 50% pruning ratios (e.g., a 1% drop both with a 20% compression ratio for in PHI-MAMBA-1.5B, and 25% compression ratio for SMOL-MAMBA-1.9B), compared to a huge drop with head pruning for a 25% pruning ratio, sometimes with the same model (e.g., a 30% drop for both HLM-3B and SMOL-MAMBA-1.9B with 5% compression ratio).

We also focused our experiments exclusively on pruning the Mamba-2 components and excluded feed-forward networks, assuming that prior work on transformer pruning had already addressed those extensively. Future work could address pruning all model components to study the interaction between the different components, as well as potential to get more significant computational savings.

Acknowledgments

We thank Jana Omary Ghattas for her help in creating Fig. 1. We also thank Yuval Reif for his support. This work was supported in part by the Israel Science Foundation (grant no. 2045/21).

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf.

2025. Smollm2: When smol goes big – data-centric training of a small language model. *Preprint*, arXiv:2502.02737.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2023. Fluctuation-based adaptive structured pruning for large language models. *Preprint*, arXiv:2312.11983.

Aviv Bick, Tobias Katsch, Nimit Sohoni, Arjun Desai, and Albert Gu. 2025. Llamba: Scaling distilled recurrent models for efficient language processing. *Preprint*, arXiv:2502.14458.

Aviv Bick, Kevin Y. Li, Eric P. Xing, J. Zico Kolter, and Albert Gu. 2024. Transformers to ssms: Distilling quadratic knowledge to subquadratic models. In *Proc. of NeurIPS*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *Preprint*, arXiv:2405.21060.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *Preprint*, arXiv:1909.11556.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proc. of ICLR*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and Arun Rao et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. arXiv:2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently modeling long sequences with structured state spaces. In *Proc. of ICLR*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Qingyun Jin, Xiaohui Song, Feng Zhou, and Zengchang Qin. 2024. Align attention heads before merging them: An effective way for converting MHA to GQA. arXiv:2412.20677.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *Preprint*, arXiv:2309.05463.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. *Preprint*, arXiv:1906.10771.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

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.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M. Rush, and Tri Dao. 2024. The mamba in the llama: Distilling and accelerating hybrid models. In *Proc. of NeurIPS*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Preprint*, arXiv:1910.07467.

A Selecting the Loss Function

Recovering performance in purned models is usually done by finetuning the model on a calibration dataset, however, this comes with the risk of over-fitting the dataset or regaining performance back in the area the dataset focuses on. Therefore, we checked the effect of using KD loss instead of Cross-Entropy (CE) loss in the data set. As can be seen in Tab. 4, it is clear that using KD loss helps the model regain overall performance along most benchmarks, while using CE, the model regains some performance in some tasks (e.g., lambada) but doesn't improve much on other benchmarks.

B Experimental Details

We finetune all models for 10K steps with batch size of 6 and learning rate of $5e-5$, using bf16. We use LoRA with rank=8 and α=16. We run our experiments on a single NVIDIA A100 80GB.

C Full Results

Tables 5 to 10 show our results on all benchmarks for the different models.

D MAMBA Pruning Advantage Experiments

Task accuracy under WANDA sparsity (Table 11). Table 11 contrasts the accuracy of the transformer-based Smol17 (TF) and its Mamba distilled version Smol-Mamba-1.9B when pruned with WANDA at 0 %, 25 %, and 50 % sparsity. At 25 % sparsity, both architectures preserve their dense performance: the average score declines by only 0.3 pp for Smol17 (TF) and 0.4 pp for Smol-Mamba-1.9B, a relative drop below 1 %. Pruning 50 % of the parameters yields a moderate degradation of 6.5 % relative to the transformer and 8 % for the Mamba variant. These results indicate that WANDA can nullify up to 25 % of weights from both model families with negligible impact, and as much as 50 % while retaining useful accuracy, with a slightly larger advantage for the transformer based model.

End-to-end throughput with FLAP pruning (Table 12). Table 12 reports batch inference throughput on a single NVIDIA A100-40 GB (host: 128 GB RAM, 128 CPU cores) when only Mamba blocks are pruned using FLAP. With 50 % sparsity (FLAP p0.5), throughput improves by 5–17 % relative to the dense Smol-Mamba-1.9B baseline, the largest gains occurring for longer sequences and larger batch sizes (e.g., +17 % at batch 16, seq 128). At 25 % sparsity (FLAP p0.25), the speed increases shrink to 0–6 %, and the lightly pruned model occasionally falls slightly below baseline for the longest sequences, suggesting that mild sparsity does not offset the overhead of sparse kernels when memory bandwidth dominates. To conclude, these findings demonstrate that aggressive but still accuracy-preserving FLAP pruning of Mamba components offers measurable end-to-end acceleration, particularly when high throughput is desired in a practical deployment.

LLAMBA 1B checks (Table 14 Table 13). In a concurrent work (Bick et al., 2025) to ours there was a release of additional family of Mamba distilled models from transformer base models, we ran pruning with WANDA and FLAP and ran the same benchmarks and put the results in Table 13 and Table 14.

	arc_challenge	arc_easy	hellaswag	lambada_openai	piqa	winogrande	Average
Baseline	0.418	0.739	0.461	0.500	0.755	0.716	0.598
w/o FT.	0.378	0.713	0.433	0.349	0.747	0.651	0.545
CE loss	0.385	0.721	0.413	0.405	0.748	0.682	0.559
KD loss	0.410	0.726	0.454	0.452	0.753	0.693	0.581

Table 4: Comparing loss choice. Phi-Mamba-1.5B with pruned SSM states according to Taylor importance estimation, by 50% and fine-tuning using LoRA with different losses. The baseline model is the (full) fine-tuned model. KD loss consistently outperforms CE loss.

Model	Ratio	FT	arc_challenge	arc_easy	hellaswag	lambada	piqa	winogrande
HLM-3B	0.25	w/o	0.18	0.37	0.27	0.06	0.54	0.51
HLM-3B	0.25	w/	0.20	0.34	0.27	0.06	0.56	0.51
HLM-3B	0.50	w/o	0.22	0.26	0.26	0.00	0.55	0.50
HLM-3B	0.50	w/	0.17	0.35	0.27	0.06	0.56	0.51
Phi-Mamba-1.5B	0.50	w/o	0.18	0.40	0.27	0.00	0.60	0.52
Phi-Mamba-1.5B	0.50	w/	0.29	0.65	0.38	0.22	0.71	0.54
Phi-Mamba-1.5B	0.25	w/o	0.25	0.60	0.30	0.06	0.69	0.50
Phi-Mamba-1.5B	0.25	w/	0.36	0.70	0.42	0.35	0.74	0.63
Mamba-2-2.7B	0.25	w/o	0.21	0.27	0.26	0.00	0.53	0.50
Mamba-2-2.7B	0.25	w/	0.22	0.25	0.26	0.00	0.52	0.45
Mamba-2-2.7B	0.50	w/o	0.22	0.26	0.25	0.00	0.52	0.50
Mamba-2-2.7B	0.50	w/	0.23	0.26	0.26	0.00	0.53	0.50
Smol-Mamba-1.9B	0.25	w/o	0.22	0.25	0.26	0.00	0.52	0.49
Smol-Mamba-1.9B	0.25	w/	0.18	0.33	0.26	0.00	0.55	0.52
Smol-Mamba-1.9B	0.50	w/o	0.20	0.26	0.26	0.00	0.50	0.49
Smol-Mamba-1.9B	0.50	w/	0.19	0.32	0.26	0.00	0.54	0.51

Table 5: Head dimension pruning benchmarks results. See Fig. 3 for radar plot visualization.

Model	Ratio	arc_challenge	arc_easy	hellaswag	lambada	piqa	winogrande
Phi-Mamba-1.5B	0.25	0.41	0.74	0.46	0.50	0.76	0.72
Phi-Mamba-1.5B	0.50	0.39	0.72	0.45	0.40	0.75	0.69
Mamba-2-2.7B	0.25	0.31	0.63	0.42	0.49	0.72	0.58
Mamba-2-2.7B	0.50	0.20	0.35	0.28	0.02	0.57	0.54
HLM-3B	0.25	0.51	0.80	0.55	0.55	0.77	0.68
HLM-3B	0.50	0.51	0.80	0.54	0.54	0.76	0.66
Smol-Mamba-1.9B	0.25	0.42	0.75	0.50	0.56	0.77	0.63
Smol-Mamba-1.9B	0.50	0.37	0.72	0.47	0.42	0.76	0.59

Table 6: WANDA pruning benchmarks results. See Fig. 4 for radar plot visualization.

model	Heads	FT	arc_challenge	arc_easy	hellaswag	lambada	piqa	winogrande
Phi-Mamba-1.5B	16	w/o	0.39	0.72	0.47	0.42	0.75	0.59
Phi-Mamba-1.5B	16	w/	0.45	0.73	0.49	0.43	0.76	0.62
Phi-Mamba-1.5B	8	w/o	0.38	0.71	0.46	0.41	0.74	0.57
Phi-Mamba-1.5B	8	w/	0.41	0.72	0.47	0.42	0.75	0.58
Mamba-2-2.7B	40	w/o	0.21	0.27	0.26	0.00	0.52	0.50
Mamba-2-2.7B	40	w/	0.21	0.27	0.25	0.00	0.52	0.50
Mamba-2-2.7B	20	w/o	0.21	0.26	0.26	0.00	0.51	0.48
Mamba-2-2.7B	20	w/	0.21	0.26	0.26	0.00	0.52	0.48
Smol-Mamba-1.9B	16	w/o	0.20	0.27	0.26	0.00	0.53	0.49
Smol-Mamba-1.9B	16	w/	0.27	0.59	0.41	0.14	0.70	0.53
Smol-Mamba-1.9B	8	w/o	0.22	0.26	0.26	0.00	0.52	0.48
Smol-Mamba-1.9B	8	w/	0.32	0.66	0.40	0.25	0.72	0.54
HLM-3B	4	w/o	0.20	0.25	0.24	0.00	0.50	0.47
HLM-3B	4	w/	0.22	0.27	0.26	0.00	0.53	0.50
HLM-3B	2	w/o	0.21	0.24	0.25	0.00	0.51	0.46
HLM-3B	2	w/	0.23	0.28	0.27	0.00	0.54	0.49

Table 7: Head merging benchmarks results. See Fig. 5 for radar plot visualization.

model	arc_challenge	arc_easy	hellaswag	lambada	piqa	winogrande
Smol-Mamba-1.9B	0.43	0.75	0.51	0.57	0.77	0.63
Mamba-2-2.7B	0.33	0.70	0.50	0.70	0.76	0.64
Phi-Mamba-1.5B	0.41	0.74	0.46	0.50	0.76	0.72
HLM-3B	0.51	0.80	0.55	0.55	0.77	0.67

Table 8: Models baseline results for the benchmarks.

Model	Ratio	arc_challenge	arc_easy	hellaswag	lambada	piqa	winogrande
Phi-Mamba-1.5B	0.25	0.39	0.72	0.44	0.44	0.75	0.66
Phi-Mamba-1.5B	0.50	0.34	0.69	0.40	0.17	0.74	0.58
HLM-3B	0.25	0.22	0.27	0.26	0.00	0.52	0.49
HLM-3B	0.50	0.24	0.27	0.26	0.00	0.51	0.45
Mamba-2-2.7B	0.25	0.23	0.28	0.25	0.00	0.51	0.48
Mamba-2-2.7B	0.50	0.25	0.28	0.25	0.02	0.50	0.44
Smol-Mamba-1.9B	0.25	0.34	0.69	0.44	0.27	0.74	0.55
Smol-Mamba-1.9B	0.50	0.26	0.57	0.37	0.06	0.69	0.52

Table 9: FLAP benchmark results. See Fig. 10 for radar plot visualization.

Model	Ratio	FT	arc_challenge	arc_easy	hellaswag	lambada	piqa	winogrande
HLM-3B	0.25	w/o	0.25	0.25	0.24	0.02	0.49	0.49
HLM-3B	0.25	w/	0.26	0.27	0.25	0.01	0.51	0.50
HLM-3B	0.50	w/o	0.23	0.24	0.26	0.02	0.50	0.49
HLM-3B	0.50	w/	0.17	0.35	0.27	0.06	0.56	0.51
Mamba-2-2.7B	0.25	w/o	0.30	0.59	0.37	0.32	0.7	0.59
Mamba-2-2.7B	0.25	w/	0.31	0.59	0.38	0.32	0.69	0.59
Mamba-2-2.7B	0.50	w/o	0.30	0.57	0.37	0.32	0.69	0.59
Mamba-2-2.7B	0.50	w/	0.30	0.57	0.38	0.32	0.69	0.59
Phi-Mamba-1.5B	0.25	w/o	0.40	0.73	0.45	0.36	0.75	0.67
Phi-Mamba-1.5B	0.25	w/	0.40	0.74	0.46	0.46	0.75	0.71
Phi-Mamba-1.5B	0.50	w/o	0.38	0.71	0.43	0.34	0.75	0.65
Phi-Mamba-1.5B	0.50	w/	0.41	0.72	0.45	0.45	0.75	0.69
Smol-Mamba-1.9B	0.25	w/o	0.40	0.73	0.48	0.53	0.76	0.61
Smol-Mamba-1.9B	0.25	w/	0.42	0.74	0.50	0.55	0.77	0.62
Smol-Mamba-1.9B	0.50	w/o	0.31	0.63	0.40	0.27	0.70	0.52
Smol-Mamba-1.9B	0.50	w/	0.40	0.74	0.49	0.54	0.76	0.60

Table 10: State pruning benchmarks full results. See Fig. 6 for radar plot visualization.

Model	Ratio	arc_challenge	arc_easy	hellaswag	lambada	piqa	winogrande	Average
Smol-Mamba-1.9B	0.50	0.37	0.72	0.47	0.42	0.76	0.59	0.56
Smol-Mamba-1.9B	0.25	0.42	0.75	0.50	0.56	0.77	0.63	0.61
Smol-Mamba-1.9B	0	0.43	0.75	0.51	0.57	0.77	0.63	0.61

Table 11: WANDA pruning on SMOL models. With 25% sparsity both transformer and Mamba variants show negligible loss; at 50% the transformer drops 4% while the Mamba variant (Smol-Mamba-1.9B) drops $\sim$4%. See Fig. 7 for radar plot visualization.

Batch Size	Seq Len.	FLAP p0.5 (tokens/s)	FLAP p0.25 (tokens/s)	Smol-Mamba-1.9B (tokens/s)
1	128	3328.13 (x0.99)	3341.27 (x1.00)	3346.05 (x1.00)
1	512	9110.18 (x1.15)	8284.75 (x1.04)	7925.90 (x1.00)
1	1024	10315.85 (x1.05)	9992.27 (x1.01)	9866.42 (x1.00)
4	128	9012.99 (x1.17)	8145.97 (x1.06)	7668.75 (x1.00)
4	512	12415.47 (x1.13)	11319.67 (x1.03)	10997.28 (x1.00)
4	1024	13007.35 (x1.12)	11682.80 (x1.00)	11650.94 (x1.00)
16	128	12395.10 (x1.16)	11284.14 (x1.06)	10647.40 (x1.00)
16	512	13635.86 (x1.14)	12373.93 (x1.04)	11931.12 (x1.00)
16	1024	14216.09 (x1.10)	12733.21 (x0.98)	12964.43 (x1.00)

Table 12: Batch-inference throughput (tokens / s) on a single NVIDIA A100-40GB (128 GB RAM, 128 CPU cores). FLAP models are pruned only in the Mamba component; figures in parentheses show speed-up over the baseline. Pruning 50% yields 5–17% gains, while 25% pruning gives 0–6%.

Model	Ratio	arc_challenge	arc_easy	hellaswag	lambada	piqa	winogrande
LLAMBA-1B	0.50	0.32	0.68	0.43	0.43	0.73	0.58
LLAMBA-1B	0.25	0.33	0.69	0.45	0.48	0.74	0.60
LLAMBA-1B	0	0.33	0.70	0.45	0.49	0.74	0.61

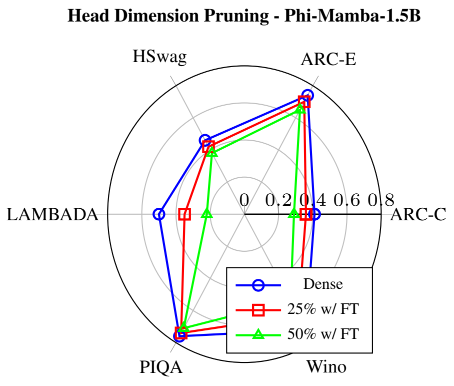
Table 13: WANDA pruning benchmarks results for LLAMBA 1B. See Fig. 8 for radar plot visualization.

Model	**Ratio**	**arc_challenge**	**arc_easy**	**hellaswag**	**lambada**	**piqa**	**winogrande**
LLAMBA-1B	0.25	0.27	0.64	0.37	0.27	0.70	0.52
LLAMBA-1B	0.50	0.18	0.37	0.28	0.01	0.60	0.51

Table 14: FLAP benchmark results for LLAMBA 1B. See Fig. 9 for radar plot visualization.

E Radar Plot Visualizations

This section contains radar plot visualizations corresponding to the results tables in the appendix.



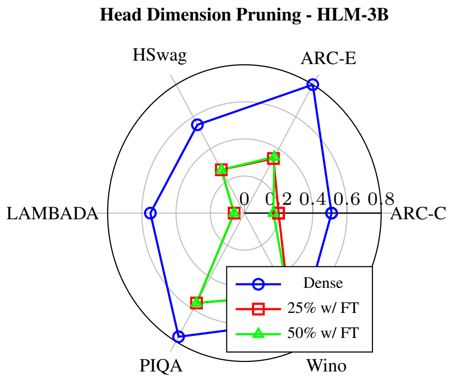
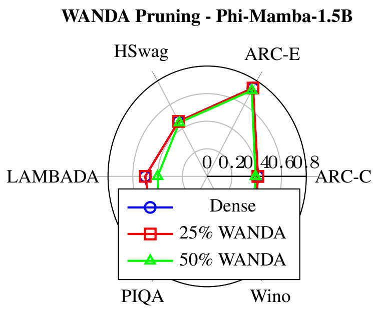
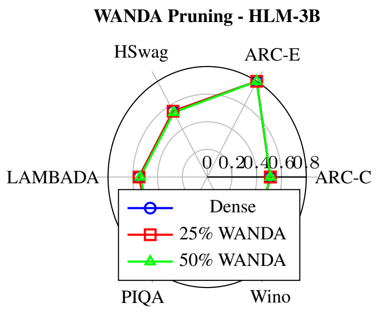
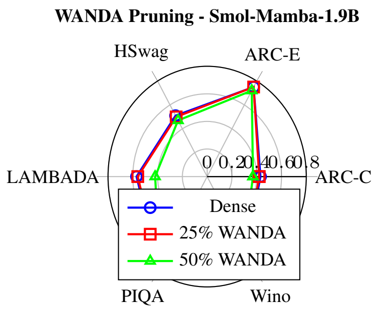


Figure 3: Radar plots showing the effect of head dimension pruning ratios on Phi-Mamba-1.5B and HLM-3B across all benchmarks.







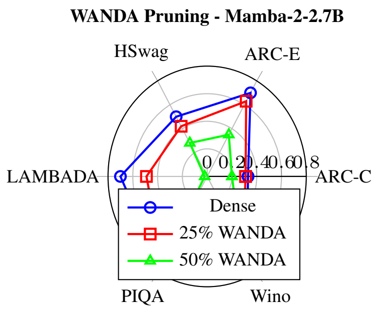


Figure 4: Radar plots showing the effect of WANDA pruning ratios on all four models across all benchmarks.

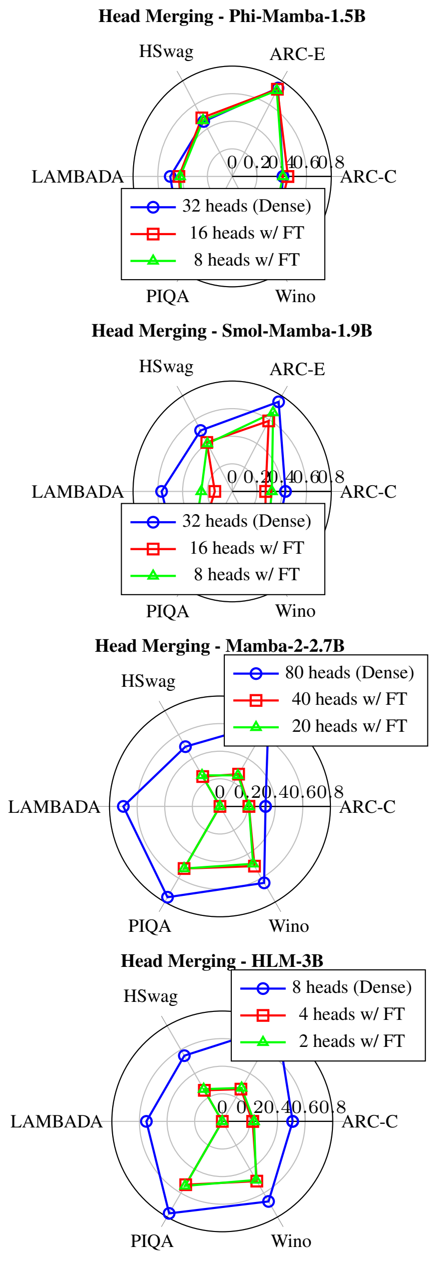


Figure 5: Radar plots showing the effect of head merging on all four models across all benchmarks.

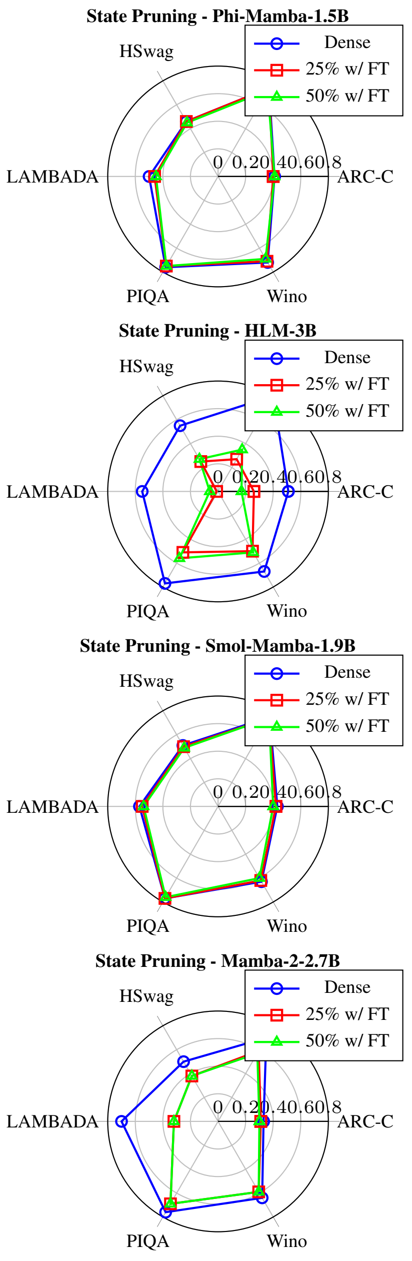


Figure 6: Radar plots showing the effect of state pruning ratios on all four models across all benchmarks.

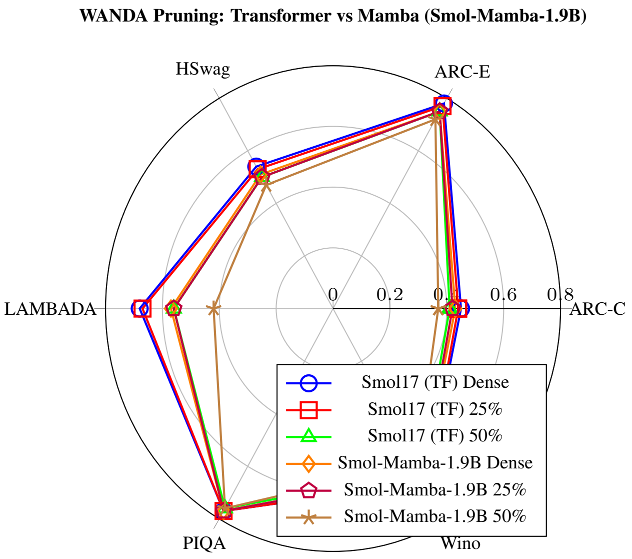


Figure 7: Radar plot comparing WANDA pruning effects on transformer (Smol17) vs Mamba (Smol-Mamba-1.9B) variants across all benchmarks and pruning ratios.

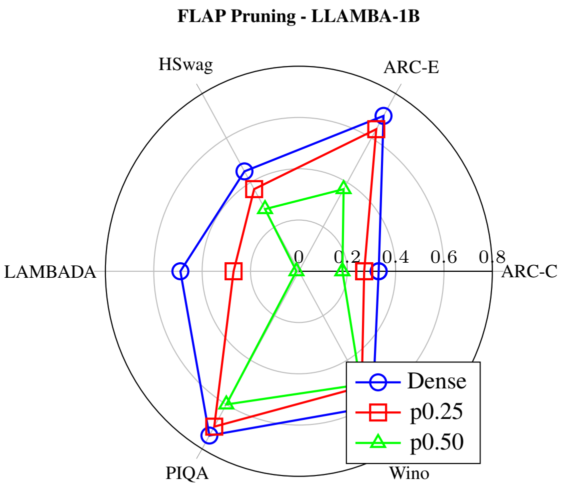


Figure 9: Radar plot showing the effect of FLAP pruning ratios on LLAMBA-1B across all benchmarks.

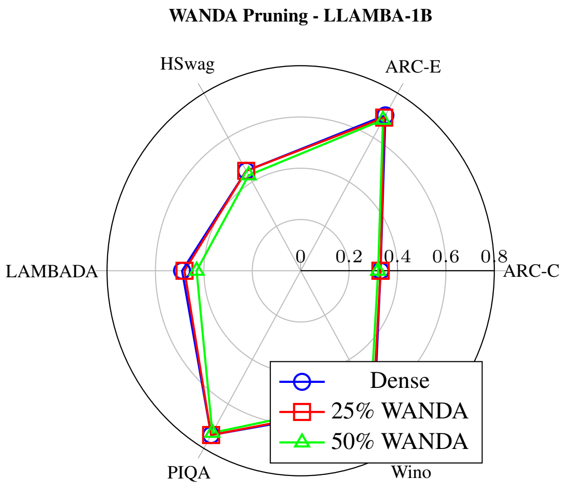


Figure 8: Radar plot showing the effect of WANDA pruning ratios on LLAMBA-1B across all benchmarks.

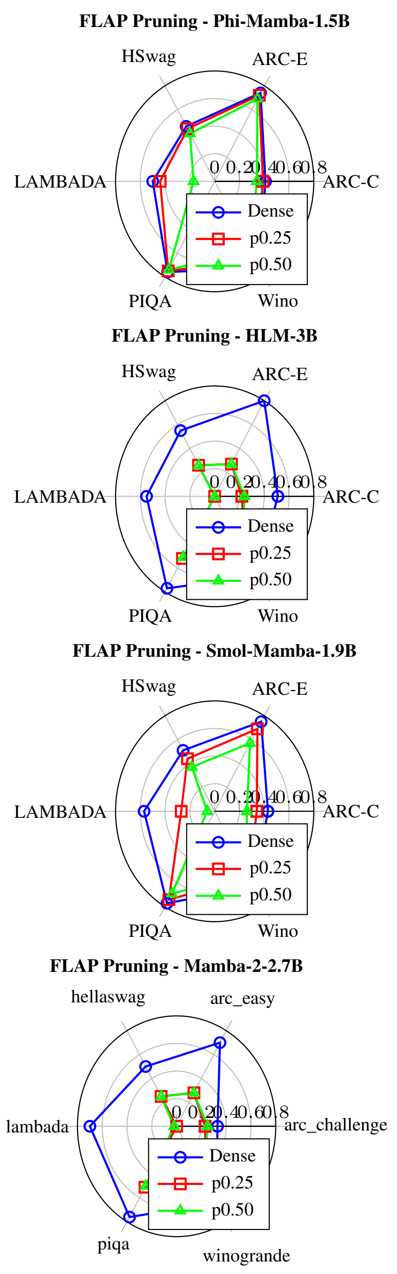


Figure 10: Radar plots showing the effect of FLAP pruning ratios on all four models across all benchmarks.