Mamba-Shedder: Post-Transformer Compression for Efficient Selective Structured State Space Models

Anonymous ACL submission

Abstract

Large pre-trained models have achieved outstanding results in sequence modeling in the past few years. The Transformer block and its attention mechanism have been the main drivers of the success of these models. Recently, alternative architectures, such as Structured State Space Models (SSMs), have been proposed to address the inefficiencies of Transformers. This paper explores the compression of the recently proposed Mamba architecture (and their hybrids). We study the sensitivity of these models to the removal of selected components at different granularities to reduce the model size and computational overhead, thus improving their efficiency while maintaining accuracy. The proposed solutions, collectively referred to as **Mamba-Shedder**, achieve a speed-up of up to 1.39x during inference, demonstrating that model efficiency can be improved by eliminating several redundancies with minimal impact on overall model performance. We make Mamba-Shedder's pruned models available to reviewers for reproducing our results at: https://anonymous.4open.science/r/mamba-shedder-examples-848A

1 Introduction

The past few years have seen an outstanding increase in the number of Transformer-based models (Vaswani et al., 2017) to tackle tasks from Natural Language Processing (NLP) and other domains (Parmar et al., 2018; Dosovitskiy et al., 2021; Arnab et al., 2021; Gong et al., 2021). Their effectiveness at modeling sequences has motivated the adoption of Transformers and their attention mechanism. However, these models also present critical efficiency challenges. For example, the cost of training these models scales quadratically in the sequence length. In the generation stage, Transformers, in their original form, require large caches to store the previously seen tokens. Several variants have been proposed to address these efficiency challenges within the framework of Transformers, but researchers have also explored alternative post-Transformer architectures to address these limitations. *Structured state space models (SSMs)*, e.g., S4 (Gu et al., 2022), followed by *Selective state space* models, e.g., Mamba (Gu and Dao, 2023; Dao and Gu, 2024) have been proposed as efficient alternatives that achieve the desired benefits at training time with linear scaling in sequence length, and during generation, SSMs maintain constant state size.

Model compression methods, e.g., pruning and quantization, have been broadly explored and applied to Transformer-based models. However, more must be done to explore compression in their structured state-space counterparts. This paper explores the pruning of these alternative architectures, presenting experiment results that provide insights on potential opportunities to increase their efficiency without sacrificing accuracy. The rest of the paper discusses the following contributions:

- A pruning solution, **Mamba-Shedder**, which targets structures in selective structured state space models, improving their computational and memory efficiency.
- Comprehensive experiments to determine the tolerance of SSM-based models to the removal of their structures.
- Insights on how the differences in the SSM building blocks and their interaction with Transformer blocks in hybrid models affect the trade-off between efficiency and accuracy.

The following content is organized as follows: Section 2 provides the reader with details of the alternative architectures utilized in our study and popular strategies for element removal in large models. Section 3 describes our methods to study network pruning in Mamba and hybrid (Transformer

and Mamba components) architectures. Section 4 presents the results of our experiments and ablation studies, and we offer concluding remarks in Section 5. Due to page limits, we include a Related Work section in the Appendix.

2 Preliminaries

2.1 State Space Models

State space models (SSMs) have a long history of modeling sequences and dynamic systems. Recently, *structured* SSMs, e.g., S4 (Gu et al., 2022), have been proposed as an alternative to Transformers because of their efficient capabilities for mapping input to output signals. When dealing with discrete sequences as in Natural Language Processing (NLP), the parameters $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ of these models are discretized to transform an input sequence, x_t, and hidden state, h_t, to obtain the output sequence, y_t. It can be formalized as:

$$h_t = \boldsymbol{A}h_{t-1} + \boldsymbol{B}x_t, y_t = \boldsymbol{C}^\top h_t. \quad (1)$$

Mamba: Selective State Space Models S4 and other structured SSMs are linear time-invariant (LTI), i.e., their parameters are fixed, limiting their effectiveness for sequence modeling. For instance, structured state space models fail in many content- and context-based reasoning tasks. These limitations have motivated the development of time-varying alternatives, e.g., Mamba (Gu and Dao, 2023), which incorporates selection mechanisms and are more robust to solving the tasks that previous SSM generations failed. Specifically, Mamba's SSM module, S6, allows the parameters that govern the SSM to depend on the input, thereby modifying the formulation from time-invariant to time-varying. Formally, Equation 1 evolves to:

$$h_t = \boldsymbol{A}_t h_{t-1} + \boldsymbol{B}_t x_t, y_t = \boldsymbol{C}_t^\top h_t. \quad (2)$$

A second improvement proposed in Mamba compared to previous SSMs is a hardware-aware algorithm that speeds up execution while reducing memory IOs.

Furthermore, Mamba-2 (Dao and Gu, 2024) improves the original Mamba architecture by proposing *state space duality (SSD)*, which improves its efficiency on hardware accelerators compared to S6. This improvement is achieved by changing the *state matrix*, $\boldsymbol{A}$, which directly controls the latent state, h. $\boldsymbol{A}$ is modified from being structured as a diagonal matrix to a formulation that utilizes a scalar-times-identity structure.

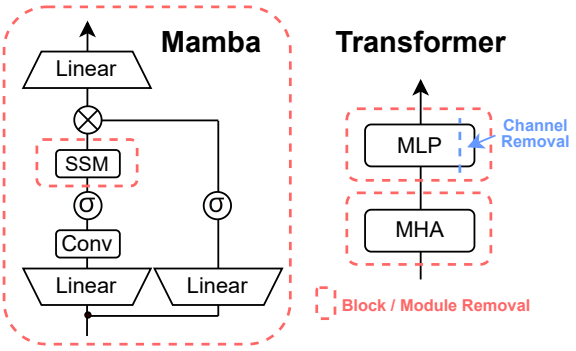

Figure 1: Mamba and Transformer blocks. We highlight in red dashed lines the structures that Mamba-Shedder analyzes for potential removal, e.g., the entire Mamba block or the SSM module. In hybrid models, Mamba-Shedder also targets the main components of the Transformer block: multihead attention (MHA) and multilayer perceptron (MLP) (entire layers and groups of channels). The illustration omits other components of the Transformer block irrelevant to the discussion of Mamba-Shedder, e.g., layer normalization.

Additionally, Mamba-2 introduces the concept of heads in SSMs inspired by how multi-head attention (MHA) works and implementing a grouped-value attention (GVA) head structure. Overall, the Mamba-2 architecture, with its SSD core component, allows for improved parallelism of the block's projections.

Mamba block Mamba models comprise several blocks stacked after each other. Figure 1 on the left illustrates a single Mamba block. Each block has the selective SSM mechanism (S6 for Mamba-1 and SSD for Mamba-2) at its core, placed within a larger structure that combines a gated multilayer perceptron (MLP), a convolution, and SILU activation functions (Elfwing et al., 2018).

For more details about selective structured state space models, we refer the reader to Gu and Dao (2023) and Dao and Gu (2024).

2.2 Hybrid Models

Lately, new models have been proposed that achieve the best of both worlds (Transformers and Selective SSMs) by proposing architectures with both classes of blocks. Zamba (Glorioso et al., 2024) is one example of such a hybrid model. It combines the strengths of Mamba's backbone and the efficiency of selective SSMs with a shared Transformer block that incorporates Transform-

ers' powerful in-context learning capabilities. The *shared attention* mechanisms, in which two attention blocks are reused and interleaved in an ABAB pattern throughout the network, is a characteristic innovation of Zamba. This model also applies LoRA (Hu et al., 2022) adapters to the shared MLP blocks, achieving specialization when interacting with the affected layers. The innovative combination of Transformers and structured SSMs allows Zamba to achieve high performance while being more memory efficient and faster at inference with reduced computational overhead than traditional Transformer-based models.

Section 4 describes the evaluation of Mamba-Shedder on Mamba and Zamba models and includes more details and specifications of the models we analyzed.

2.3 Model Pruning

A popular model compression technique, *pruning* (LeCun et al., 1989), has been effectively used to reduce the size of deep learning models and improve their efficiency. Network pruning operates at two levels: (1) *Unstructured pruning* identifies the importance of individual weights that can be masked to minimize their impact on overall model behavior. Several efficient unstructured pruning methods have been proposed in the age of large models such as Wanda (Sun et al., 2023). At a different level, (2) *structured pruning* focuses on removing more significant structural components of the model, such as whole Transformer blocks (Men et al., 2024), or reducing the granularity to target subcomponents of these layers (Zhong et al., 2024). Other dimensions for pruning include groups of channels in the Transformer's MLPs or heads from the MHA layer. In this paper, the focus is solely on structured pruning applied to Mamba-based models.

Next, we discuss Mamba-Shedder, a solution for efficient removal of structures in Mamba and hybrid models.

3 Methodology

Due to the large sizes of current state-of-the-art sequence models, Mamba-Shedder requires an efficient strategy to identify structures that can be removed without significantly affecting the model's accuracy. We approach this problem using a training-free approach, in which the least essential elements are considered for removal. Similar strategies have been explored in Transformer-based large language models (Ashkboos et al., 2024; Men et al., 2024; Zhong et al., 2024). However, to the best of our knowledge, no study explores the removal of structures in Selective Structured State Space models. Mamba-Shedder conducts structure removal of Mamba models and their hybrid variants at different granularities. As illustrated in the left of Figure 1, in the case of models with only Mamba blocks, we explore the iterative removal of:

- Entire Mamba blocks (§2.1)
- SSM subcomponents, either S6 or SSD modules depending on the version of Mamba (Figure 1)

The proponents of the Mamba architecture do not provide a rationale for the number of Mamba blocks required to build robust models, opening an opportunity for Mamba-Shedder to investigate whether some components might be redundant and hence removed from the model with a minor impact in accuracy.

Algorithm 1 Block / Module Pruning

Input: Set of blocks/modules $\mathcal{M}$ from a model m, Calibration dataset $\mathcal{C}$, Metric ϕ, Target pruning steps t.
Output: Pruned model m^*

for $k \leftarrow 1$ **to** t **do**
  for all $M_i \in \mathcal{M}$ **do**
    $S_i \leftarrow \text{Importance}(M_i, m, \mathcal{C}, \phi)$
  end for
  $M_{\text{min}} \leftarrow \arg\min_{M_i \in \mathcal{M}} S_i$
  $\mathcal{M} \leftarrow \mathcal{M} \setminus \{M_{min}\}$ ▷ Block/Module Pruning
end for
return m^* with the remaining blocks/modules in $\mathcal{M}$

Algorithm 2 MLP Channel Pruning

Input: Set of MLP blocks $\mathcal{M}_{\text{MLP}}$ from a model m, Calibration dataset $\mathcal{C}$, Metric ϕ, Target pruning steps t, MLP channel group size g.
Output: Pruned model m^*

for $k \leftarrow 1$ **to** t **do**
  for all $M_i \in \mathcal{M}_{\text{MLP}}$ **do**
    $S_i \leftarrow \text{Importance}(M_i[:, :\text{-}g], m, \mathcal{C}, \phi)$
  end for
  $M_{\text{min}} = \arg\min_{M_i \in \mathcal{M}_{\text{MLP}}} S_i$
  $M_{\text{min}} = M_{\text{min}}[:, :\text{-}g]$ ▷ Channel Pruning
end for
return m^* with the altered MLP blocks in $\mathcal{M}$

In addition to these components, in the case of hybrid models that also contain Transformer blocks (right of Figure 1), we also explore the removal of entire Transformer blocks or their subblocks: multilayer perceptrons (MLP) modules and multi-head attention (MHA) modules. In hybrid models,

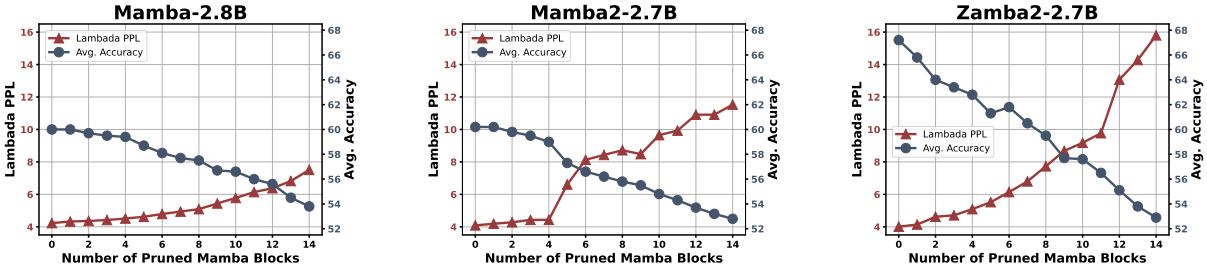

Figure 2: Pruning Mamba blocks. The model composed of Mamba 1 blocks (left) can tolerate the removal of entire blocks without significantly increasing its perplexity or decreasing accuracy compared to Mamba-2 and Zamba-2. In all three models, removing each Mamba block reduces 0.04B parameters from the model. The average accuracy is for the seven tasks used for evaluation. These are *training-free* results, and drops in accuracy might be reduced by a subsequent fine-tuning stage (§4.5).

Model	Method	Num. of Pruned Mamba Blocks	Ratio	PPL (↓)	Lambada	HellaS	PIQA	ARC-e	ARC-c	WinoG	OBQA	Avg. Score
Mamba-2.8B	Dense	0 / 64	0%	4.23	69.2	66.1	75.2	69.7	36.3	63.5	39.6	59.9
	Mamba Block Pruning	7 / 64	10.43%	**4.94**$_{+0.71}$	65.8	63.7	73.8	68.0	33.5	62.5	36.8	**57.7**$_{-2.2}$
		14 / 64	20.86%	**7.51**$_{+3.28}$	58.9	57.6	71.0	62.7	32.0	61.1	33.2	**53.8**$_{-6.1}$
Mamba2-2.7B	Dense	0 / 64	0%	4.10	69.7	66.6	76.4	69.6	36.4	64.0	38.8	60.2
	Mamba Block Pruning	7 / 64	10.42%	**8.43**$_{+4.33}$	53.0	63.8	73.9	66.6	36.4	64.5	35.0	**56.2**$_{-4.0}$
		14 / 64	20.83%	**11.53**$_{+7.43}$	47.0	59.4	71.1	60.6	35.6	60.8	35.0	**52.8**$_{-7.4}$
Zamba2-2.7B	Dense	0 / 54	0%	4.01	69.7	77.0	79.8	77.5	48.5	72.1	45.8	67.2
	Mamba Block Pruning	7 / 54	10.38%	**6.80**$_{+2.79}$	58.9	69.7	77.0	69.8	39.6	67.0	41.8	**60.5**$_{-6.7}$
		14 / 54	20.77%	**15.8**$_{+11.79}$	44.3	62.8	72.7	54.3	34.5	64.3	37.2	**52.9**$_{-14.3}$

Table 1: The results of Mamba-Shedder with *training-free* Mamba block pruning. "PPL" refers to the perplexity in Lambada, while Lambada, HellaS, PIQA, ARC-e, ARC-c, WinoG, and OBQA represent their respective accuracies. "Avg. Score" indicates the average accuracy for these seven tasks. Underlined numbers indicate the smallest average accuracy gap with the dense model under the same level of pruning. A subsequent fine-tuning stage (§4.5) might close this gap.

Mamba-Shedder also explores the removal of structures at a finer granularity by targeting groups of channels in the MLP's linear layers, i.e., based on a channel group size, g, Mamba-Shedder explores the removal of ng channels, where n is the number of groups that could be removed based on their impact of the overall model performance.

Algorithm 1 details the procedure to remove entire structures, e.g., Mamba or Transformer blocks, MLPs, MHA, or SSM modules. Given a set $\mathcal{M}$ of structures selected for potential removal, a proxy data set C and a metric ϕ are used to measure the importance of an individual structure and the impact of removing it from the model (Zhong et al., 2024). In addition to entire structures, Mamba-Shedder follows the same logic to remove channel groups as detailed in Algorithm 2.

Depending on the pruning objective, Mamba-Shedder might treat these pruning targets in isolation, but Section 4 also presents the results of configurations in which Mamba-Shedder sequentially prunes larger structures (e.g., Mamba blocks) and, at a later stage, smaller components, e.g., SSM modules in the remaining Mamba blocks. Future work will explore larger search spaces with more complex configurations of candidate structures for removal. For example, the importance of Mamba blocks and their SSM modules can be assessed in the same pruning iteration.

4 Experiments

We evaluate Mamba-Shedder and study the removal of structures from SSM-based models utilizing several open-source models and datasets. We analyze their absolute and relative drop in accuracy and quantify the inference speedup obtained by the pruning stage. Next, we discuss the resources utilized for our experiments and details on our setup and results.

4.1 Models

Our experiments employed the following pretrained Mamba and hybrid models of selective state space and transformer blocks: **Mamba-2.8b** (Gu

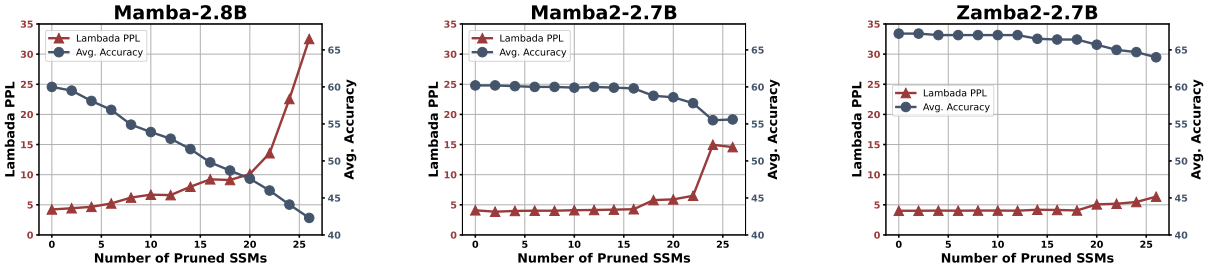

Figure 3: The results of Mamba-Shedder with *training-free* SSM (S6 and SSD modules) pruning , where Mamba-2.8B and Mamba2-2.7B both have 64 SSM modules, while Zamba2-2.7B has 54 SSM (SSD) modules. The *average accuracy* is for the seven tasks evaluated.

Model	Method	Num. of Pruned SSMs	PPL (↓)	Lambada	HellaS	PIQA	ARC-e	ARC-c	WinoG	OBQA	Avg. Score
Mamba-2.8B	Dense	0 / 64	4.23	69.2	66.1	75.2	69.7	36.3	63.5	39.6	59.9
	SSM Pruning	16 / 64	**9.23**$_{+5.00}$	55.2	52.1	68.1	57.8	28.4	55.6	31.6	**49.8**$_{-10.1}$
		20 / 64	**10.10**$_{+5.87}$	57.1	48.2	65.5	50.9	25.9	56.0	29.4	**47.6**$_{-12.3}$
		24 / 64	**22.55**$_{+18.32}$	44.4	43.2	64.4	47.4	25.8	53.6	29.8	**44.1**$_{-15.8}$
Mamba2-2.7B	Dense	0 / 64	4.10	69.7	66.6	76.4	69.6	36.4	64.0	38.8	60.2
	SSM Pruning	16 / 64	**4.26**$_{+0.16}$	66.9	66.1	76.4	68.6	37.2	64.0	39.2	**59.8**$_{-0.4}$
		20 / 64	**5.89**$_{+1.79}$	59.8	66.0	76.1	68.9	36.7	63.6	39.2	**58.6**$_{-1.6}$
		24 / 64	**14.95**$_{+10.85}$	43.4	65.8	74.8	67.1	36.6	62.9	38.0	**55.5**$_{-4.7}$
Zamba2-2.7B	Dense	0 / 54	4.01	69.7	77.0	79.8	77.5	48.5	72.1	45.8	67.2
	SSM Pruning	16 / 54	**4.14**$_{+0.13}$	69.2	75.8	79.2	75.8	46.5	72.2	45.8	**66.4**$_{-0.8}$
		20 / 54	**5.07**$_{+1.06}$	64.2	75.8	79.3	75.5	46.2	73.2	46.0	**65.7**$_{-1.5}$
		24 / 54	**5.46**$_{+1.45}$	62.3	74.7	79.0	75.4	44.3	70.9	46.4	**64.7**$_{-2.5}$

Table 2: The results of Mamba-Shedder with *training-free* SSM pruning. "PPL" refers to perplexity in Lambada, while the remaining tasks represent their respective accuracy. "Avg. Score" indicates the average accuracy across seven tasks. Here, we do not consider the pruning ratio, as the ratio of SSM's parameter weights is small. Its benefit is the reduction of computational overhead. Underlined numbers indicate the smallest gap with Dense under the same level of pruning.

and Dao, 2023), the largest version of the Mamba-1 series, consists of 64 S6 blocks[1]. **Mamba2-2.7b** (Dao and Gu, 2024), the largest version of Mamba-2, consists of 64 SSD blocks [2]. Both Mamba models were trained on 300B tokens on the Pile dataset (Gao et al., 2020). For our choice of a hybrid model, we explored **Zamba2-2.7B** (Glorioso et al., 2024)[3]. It has 54 layers, including 45 single Mamba-2 Blocks and 9 hybrid layers composed of both Mamba-2 Blocks and Transformer Blocks. Zamba-2 was trained on 3T tokens from open web datasets, including Zyda (Tokpanov et al., 2024), and subsequently annealed with 100B additional tokens.

4.2 Datasets

Following the language modeling evaluation of Mamba (Gu and Dao, 2023; Dao and Gu, 2024), we utilize *lm-eval-harness* (Gao et al., 2023) to assess the zero-shot performance, which includes measuring perplexity on Lambada (Paperno et al., 2016), and accuracy on the following downstream tasks: HellaSwag (Zellers et al., 2019), Physical Interaction Question Answering (PIQA) (Bisk et al., 2020), AI2 Reasoning Challenges (Arc-e, Arc-c) (Clark et al., 2018), Large-scale Winograd Schema Challenge (WinoGrande) (Sakaguchi et al., 2021), and the Open Book Question Answering (Mihaylov et al., 2018) dataset.

All the hyperparameters used in our experiments are detailed in the Appendix.

4.3 Results

4.3.1 Pruning Target: Mamba Block

This section explores the impact of pruning Mamba blocks on model performance. Figure 2 and Table 1 present the results of applying Mamba-Shedder to Mamba-2.8B, Mamba2-2.7B, and Zamba2-2.7B models with a focus on removing redundant entire Mamba blocks. The model that utilizes the first version of Mamba blocks (S6) appears to tolerate

[1]https://huggingface.co/state-spaces/mamba-2.8b
[2]https://huggingface.co/state-spaces/mamba2-2.7b
[3]https://huggingface.co/Zyphra/Zamba2-2.7B

Pruning Target	Ratio	Additional Pruned SSMs	PPL (↓)	Lambada	HellaS	PIQA	ARC-e	ARC-c	WinoG	OBQA	Avg. Score
/	0%	0 / 54	4.01	69.7	77.0	79.8	77.5	48.5	72.1	45.8	67.2
Mamba Block & Transformer Block	10.40%	0 / 54	$9.18_{+5.17}$	53.5	67.3	76.3	63.5	37.8	64.3	40.6	$57.6_{-9.6}$
Mamba Block & MLP & MHA	10.33%	0 / 54	$\mathbf{5.01_{+1.00}}$	**65.6**	73.6	78.5	75.3	43.8	69.3	45.2	$64.5_{-2.7}$
Mamba Block & MLP & MHA + MLP Channel	10.27%	0 / 54	$5.45_{+1.44}$	63.4	**74.9**	**80.1**	**79.0**	**49.7**	**70.9**	46.0	$\mathbf{66.3_{-0.9}}$
Mamba Block & MLP & MHA + MLP Channel + SSM	10.27%	**18 / 54**	$5.18_{+1.17}$	63.4	73.9	80.0	**79.0**	48.7	69.5	**46.6**	$65.9_{-1.3}$
Mamba Block & Transformer Block	15.89%	0 / 54	$10.38_{+6.37}$	51.4	65.6	74.0	61.7	37.7	63.5	39.6	$56.2_{-11.0}$
Mamba Block & MLP & MHA	15.54%	0 / 54	$10.64_{+6.63}$	49.3	69.2	76.9	66.1	38.1	66.0	41.8	$58.2_{-9.0}$
Mamba Block & MLP & MHA + MLP Channel	15.48%	0 / 54	$\mathbf{7.39_{+3.38}}$	**57.6**	**70.0**	**78.5**	**74.5**	**43.9**	67.5	**43.8**	$\mathbf{62.3_{-4.9}}$
Mamba Block & MLP & MHA + MLP Channel + SSM	15.48%	**18 / 54**	$7.43_{+3.42}$	56.5	68.9	77.9	73.4	41.8	**67.7**	42.8	$61.3_{-5.9}$

Table 3: The results of Zamba2-2.7B were achieved by pruning its Mamba-2 and Transformers blocks at multiple granularities, including entire Mamba-2 block, MHA block, MLP block, MLP channel, and SSM module. "PPL" refers to perplexity in Lambada, while the remaining tasks represent their respective accuracies. "Avg. Score" indicates the average accuracy across seven tasks. "&" indicates that the pruning targets are considered together in the same pruning step, while "+" signifies the distinction between pruning stages, with pruning occurring sequentially. **Bold** numbers indicate the best performance under the same level of pruning (excluding Dense).

a higher number of removed blocks without significantly affecting its performance. Specifically, the Mamba-2.8B model demonstrates robustness, with its perplexity (PPL) increasing from 4.23 to 7.51 and average accuracy dropping from 59.9 to 53.8 when the pruning ratio reaches 20.86%. In contrast, the Mamba2-2.7B and Zamba2-2.7B models exhibit more significant performance degradation, although they performed better before pruning (Dense). The poorer pruning performance of Zamba2-2.7B may be attributed to the pruning of Mamba blocks disrupting a certain balance within the hybrid layers. Overall, the effects of Mamba block pruning vary across different models, depending on the model architecture and the characteristics of the pre-training stage. In this round, Mamba-1 comes out on top.

4.3.2 Pruning Target: SSM Module

In this section, we delve into assessing the impact of pruning only the SSM modules within Mamba blocks on the performance of various models, as illustrated in Table 2 and Figure 3. When using the same target in Mamba-2.8B, we observe that further pruning SSMs results in a noticeable increase in perplexity, soaring to 22.55 and a decrease in average accuracy to 44.1. This indicates a significant sensitivity to SSM pruning for Mamba-1, where performance degradation is pronounced even at moderate pruning levels. Conversely, Mamba2-2.7B and Zamba2-2.7B exhibit remarkable resilience to SSM pruning. Even with 24 SSMs pruned, the model maintains a relatively stable performance. This robustness suggests that Mamba-2 blocks can tolerate higher SSM module pruning, potentially due to Mamba-2's optimizations or different training strategies with Mamba-1. The Zamba2-2.7B model, with the hybrid architecture, outperforms both Mamba-1 and Mamba-2. Pruning 12 out of its 54 SSMs results in a negligible PPL increase from 4.01 to 4.02, while the average accuracy slightly decreases from 67.2% to 67.0%. The hybrid nature of Zamba2-2.7B may contribute to its ability to maintain performance despite SSM pruning. Overall, these findings underscore the importance of model architecture and training strategies in determining the impact of SSM pruning. They offer valuable insights for optimizing model efficiency without compromising performance. In this round, the model with Mamba-2 blocks comes out on top.

4.3.3 Pruning Target: Finer-grained removal of Mamba and Transformer blocks, and their subcomponents

Table 3 presents the results of pruning various components of the Zamba2-2.7B model, including combinations of Mamba-2 blocks, entire Transformer blocks, and their subcomponents, i.e., MHA blocks, MLP blocks, MLP channels, and SSM modules. We design four search spaces to study the effectiveness of different granularities and their combinations. "&" indicates that the pruning targets are considered together in the same pruning step, while "+" signifies the distinction between pruning stages, with pruning occurring sequentially:

Mamba Block & Transformer Block Pruning This experiment involves pruning the entire Mamba-2 blocks and Transformer blocks.

Table 4: Inference benchmark results for Mamba-2.8B (dense and pruned). Number of batches is 10. The prompt length is 512. Number of new tokens is 16.

Method	Num. of Pruned Mamba Block	Inference Speedup Prefill	Inference Speedup Decode
Dense	0 / 64	1.00×	1.00×
Mamba Block Pruning	7 / 64	**1.12**×	**1.13**×
	14 / 64	**1.31**×	**1.29**×

Table 5: Inference benchmark results for Mamba2-2.7B (dense and pruned), with test-related hyperparameters consistent with Table 4.

Method	Num. of Pruned SSMs	Inference Speedup Prefill	Inference Speedup Decode
Dense	0 / 64	1.00×	1.00×
SSM Pruning	16 / 64	**1.13**×	**1.11**×
	20 / 64	**1.16**×	**1.14**×
	24 / 64	**1.20**×	**1.18**×

Mamba Block & MLP & MHA Pruning This experiment decomposes the transformer block into sub-blocks, pruning Mamba-2 blocks as well as MHA and MLP.

Mamba Block & MLP & MHA + MLP Channel Pruning This experiment prunes the Mamba-2 blocks, MHA, and MLP at the first stage and further prunes the MLP channels at the next stage.

Mamba Block & MLP & MHA + MLP Channel Pruning + SSM Add additional SSM pruning following the previous solution.

The results indicate that pruning Mamba blocks and Transformer blocks alone leads to significant performance degradation. However, **more granular pruning strategies show a more favorable trade-off between pruning ratio and performance**. Specifically, pruning Mamba blocks, MLP, MHA (single stage), and MLP channels subsequently performs the best. Inspired by the SSM pruning of Mamba-2 in Section 4.3.2, we further add SSM pruning to the third strategy, and the results show that removing around 18 SSMs can maintain accuracy performance while reducing computational overhead. An interesting finding is that pruning SSMs can even lower PPL; for instance, at a 10% pruning ratio, PPL decreases from 5.45 to 5.18, suggesting that some SSM modules are redundant after the second pruning stage. Overall, these findings indicate that multi-granularity pruning methods, particularly those including MLP channels and SSM modules, can effectively reduce the complexity of hybrid Mamba models while maintaining a higher level of performance.

None of the above results have undergone fine-tuning to improve the performance of the pruned models. As in many other works, the drop in the accuracy performance of pruned models can be recovered by fine-tuning, which will be incorporated in Section 4.5.

4.4 Inference Acceleration

Through the above analysis, we have gained a good understanding and insight into the impact of Mamba-Shedder's structured pruning on model accuracy and perplexity performance. In addition, through structured pruning, Mamba-Shedder achieves an additional speedup to these already highly efficient models. Next, we discuss the impact of inference acceleration. All the following tests were conducted on a single Tesla V100 32GB GPU.

Mamba-1 When removing entire Mamba blocks, as shown in Table 4, Mamba-Shedder speeds up the decoding stage up to 1.29x when removing 14 blocks, and 1.13x when removing only 7 blocks, which highlights the potential of Mamba-Shedder to optimize computational efficiency in Mamba models. The user's decision on how aggressively to prune will impact the average accuracy or the perplexity as observed in Table 1.

Mamba-2 As detailed in Table 5, removing 24 SSM modules (44% of the total number of modules) results in up to a 1.20x speedup in the prefill stage and a 1.18x speedup in the decoding stage during of inference. A more conservative pruning ratio achieves 1.11x speedup when removing 16 SSM modules. Based on previous observations, the impact on performance metrics is minimal (0.4% for accuracy and 0.16 for PPL). These results underscore the effectiveness of SSM pruning in enhancing computational efficiency while barely affecting model performance, making it a viable strategy for optimizing Mamba models.

Zamba-2 As detailed in Table 6, we observe significant acceleration on inference after multiple granularities pruning of Zamba-2. Specifically, pruning Mamba blocks, MLP, and MHA blocks along with MLP channels results in a 1.34x speedup in the decoding stage. When SSM pruning is included, the speedup increases to 1.39x, indicating that a comprehensive pruning strategy that includes multiple components can significantly enhance inference speed while maximizing the preservation of model performance.

Table 6: Inference benchmark results for Zamba2-2.7B (dense and pruned), with test-related hyperparameters consistent with Table 4.

Method / Pruning Target	Ratio	Additional Pruned SSMs	Inference Speedup	
			Prefill	Decode
Dense	0%	0 / 54	1.00×	1.00×
Mamba Block & MLP & MHA + MLP Channel	15.48%	0 / 54	1.16×	1.34×
Mamba Block & MLP & MHA + MLP Channel + SSM	15.48%	18 / 64	**1.25**×	**1.39**×

Method	Ratio	Additional Pruned SSMs	PPL (↓)	Avg. Score
Dense	0%	0 / 54	4.01	67.2
Mamba-Shedder	15.48%	0 / 54	7.39	62.3
Mamba-Shedder **w/ tune**	15.48%	0 / 54	**5.84**$_{-1.55}$	**64.7**$_{+2.4}$
Mamba-Shedder	15.48%	18 / 54	7.43	61.3
Mamba-Shedder **w/ tune**	15.48%	18 / 54	**5.88**$_{-1.55}$	**64.4**$_{+3.1}$

Table 7: Results of the compressed Zamba2-2.7B model with recovery tuning (post-training).

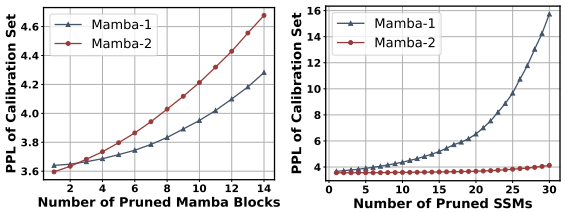

Figure 4: Close examination of the impact of removing Mamba blocks or SSMs from the two versions of Mamba models reveals distinct differences in their tolerance levels. Mamba-1 exhibits a higher tolerance for removing its blocks, while Mamba-2 exhibits greater tolerance for removing the SSM subcomponent.

4.5 Recovery Tuning of the Pruned Model

Following most of the work (Ma et al., 2023; Zhong et al., 2024), we performed post-training on the Mamba-Shedder compressed model using the cleaned version of Alpaca. The results, summarized in Table 7, demonstrate substantial performance gains after just two epochs of recovery tuning (see Appendix for more hyperparameters). The Mamba-Shedder model obtained by removing Mamba Blocks & MLPs & MHAs + MLP Channels + SSM pruning (Table 3), initially exhibits a perplexity of 7.43 and an average score of 61.3 when 18 out of 54 SSMs are pruned. However, after recovery tuning, it achieves a significantly reduced PPL of 5.88 and an improved average score of 64.4. This indicates that the recovery fine-tuning phase effectively enhances the performance of the pruned model, bringing it closer to the original dense model's performance while maintaining computational efficiency. In summary, recovery tuning is crucial to optimize pruned models, making them more viable for practical applications.

4.6 Insights on the Compression Sensitivity of the Variants of Mamba

A research question during our investigation considered, *will the improvements in Mamba-2 make it more sensitive to removing its inner structures?*

The proponents of Mamba modified the original architecture to restrict the expressivity in Mamba-2 and increase the training efficiency. As illustrated on the left side of Figure 4, our experiments suggest that these changes make Mamba-2 models less robust to removing entire blocks than the previous version of the Mamba block. As soon as we remove blocks with the least importance, Mamba-1 exhibits a more robust behavior. However, Mamba-2 demonstrates a significantly higher tolerance to the removal of SSMs, maintaining a stable perplexity even as more SSMs are pruned. This indicates that while Mamba-2's architectural improvements have made it more sensitive to the removal of Mamba blocks, they have also enhanced its robustness to SSM pruning.

5 Conclusion

Selective structure state space models are becoming an efficient alternative to Transformer-based models. In this paper, we propose Mamba-Shedder and investigate structured pruning strategies to remove elements from Mamba and hybrid models and reduce model size, accelerating inference. The results demonstrate that selective structured state space architectures have several redundancies that can be removed without significantly affecting the model's performance. Mamba-Shedder's code will be released open-source for the research community. In the meantime, we make Mamba-Shedder's pruned models available to reviewers for reproducing our results at: `https://anonymous.4open.science/r/mamba-shedder-examples-848A`.

Limitations

Despite their outstanding results, large sequence models are still under investigation to better understand their capabilities and limitations. Mamba-Shedder is, to the best of our knowledge, the first work to investigate the removal of structures in Mamba-based models, including hybrids with Transformer blocks. Our goal is to motivate the research community to better understand this class of models to identify opportunities for future improvements in the model architecture or on custom compression techniques. The results indicate that these models contain redundant elements that might be removed to improve their efficiency. However, future work must explore and attempt to better understand the trade-offs between efficiency and accuracy when removing these models' components. Even more research questions can be entertained when considering Transformer blocks and hybrid models, as in the case of Zamba. For instance, there is much to understand about the right mix of the SSM- and Transformer-based elements.

Ethics Statement

Due to the well-known flaws in large sequence models, e.g., hallucinations, many guard rails must be in place when considering deploying them in production. Our research focuses on improving the efficiency of these models in existing downstream tasks and datasets. However, further experimentation and analysis are needed when considering deploying these compressed models in environments where their output might affect people's well-being.

References

A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid. 2021. Vivit: A video vision transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6816–6826, Los Alamitos, CA, USA. IEEE Computer Society.

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. In *The Twelfth International Conference on Learning Representations*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

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*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.

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

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184, Hong Kong, China. Association for Computational Linguistics.

Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V. Le. 2020. Funnel-transformer: filtering out sequential redundancy for efficient language processing. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Association for Computational Linguistics (ACL)*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11. Special issue on deep reinforcement learning.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: Accurate post-training compression for generative pretrained transformers. *arXiv preprint arXiv:2210.17323*.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. 2023. Hungry Hungry Hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024. Zamba: A compact 7b ssm hybrid model. *Preprint*, arXiv:2405.16712.

Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, pages 571–575.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. 2024. Combining recurrent, convolutional, and continuous-time models with linear state-space layers. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. 2021. Sparsity in deep learning: pruning and growth for efficient inference and training in neural networks. *J. Mach. Learn. Res.*, 22(1).

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander Rush. 2021. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10619–10629, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

J. Pablo Munoz, Jinjie Yuan, and Nilesh Jain. 2024. Shears: Unstructured sparsity with neural low-rank adapter search. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 395–405, Mexico City, Mexico. Association for Computational Linguistics.

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.

Niki J. Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *International Conference on Machine Learning (ICML)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

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

Yury Tokpanov, Beren Millidge, Paolo Glorioso, Jonathan Pilault, Adam Ibrahim, James Whittington, and Quentin Anthony. 2024. Zyda: A 1.3 t dataset for open language modeling. *arXiv preprint arXiv:2406.01981*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Peng Xu, Wenqi Shao, Mengzhao Chen, Shitao Tang, Kaipeng Zhang, Peng Gao, Fengwei An, Yu Qiao, and Ping Luo. 2024. Besa: Pruning large language models with blockwise parameter-efficient sparsity allocation. *Preprint*, arXiv:2402.16880.

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*.

Li Zhang, Jiachen Lu, Sixia Zheng, Xinxuan Zhao, Xiatian Zhu, Yanwei Fu, Xiang Tao, and Jianfeng Feng. 2023. Vision transformers: From semantic segmentation to dense prediction. *arXiv*.

Lin Zheng, Chong Wang, and Lingpeng Kong. 2022. Linear complexity randomized self-attention mechanism. In *International Conference on Machine Learning*, pages 27011–27041. PMLR.

Longguang Zhong, Fanqi Wan, Ruijun Chen, Xiaojun Quan, and Liangzhi Li. 2024. Blockpruner: Fine-grained pruning for large language models. *Preprint*, arXiv:2406.10594.

Supplementary Material

A Related Work

Transformers (Vaswani et al., 2017) and its variants are the main building block of successful deep learning architectures, e.g., Llama (Touvron et al., 2023) and GPT (Brown et al., 2020), that have revolutionized Natural Language Processing (NLP) (Devlin et al., 2019; Gao et al., 2023), Computer Vision (CV) (Parmar et al., 2018; Radford et al., 2021; Zhang et al., 2023), and many other domains. Due to the Transformer's popularity, researchers have proposed variants to improve their computational and memory efficiency further and tackle issues like their quadratic complexity in sequence length during training (Correia et al., 2019; Beltagy et al., 2020; Dai et al., 2020; Choromanski et al., 2021; Katharopoulos et al., 2020; Zheng et al., 2022).

A parallel research effort investigates alternatives to Transformers in the form of *structured state space models* (SSMs) that can power the next generation of sequence models. The initial proposals of structured SSMs were linear time-invariant, e.g., LSSL (Gu et al., 2024), S4 (Gu et al., 2022), H3 (Fu et al., 2023). Recent improvements to the state space model formulation have resulted in the proposal of time-varying selective SSMs, e.g., Mamba (Gu and Dao, 2023; Dao and Gu, 2024).

To our knowledge, Mamba-Shedder is the first study on pruning selective structured state space models (Mamba) and their hybrids. On the other hand, many works have proposed pruning techniques for Transformer-based models (Hoefler et al., 2021). Several of these works focus on *unstructured* pruning (Sun et al., 2023; Xu et al., 2024; Frantar et al., 2022), which can achieve higher sparsity levels. However, it requires highly optimized runtimes to realize the benefits of sparsity. Sophisticated solutions have been proposed to fine-tune sparse models and recover any accuracy drop from the pruning stage (Munoz et al., 2024). Recently, *training-free* approaches have been proposed for *structured* pruning of Transformers. These approaches cannot achieve high sparsity levels as the *unstructured* pruning approaches. However, they are very convenient because their compressed models do not require specialized runtimes and exhibit beneficial inference acceleration. In this line of research, LLMPruner (Ma et al., 2023), ShortGPT (Men et al., 2024), BlockPruner (Lagunas et al., 2021), SliceGPT (Ashkboos et al., 2024) have demonstrated efficient methods for Transformer pruning. BlockPruner improved over many previous approaches by proposing a global metric that can be used to determine the importance of a selected network structure. Mamba-Shedder builds on this and the rest of the extensive literature on *structured* block pruning to explore opportunities for removing redundancies in models with Mamba blocks.

Hyper-parameter	**Value**
Pruning Stage:	
MLP Channel Group Size	1024
Steps of MLP Channel Pruning	20
Calibration Dataset	tatsu-lab/alpaca
Importance Metric	Perplexity (PPL)
Number of Calibration Samples	256
Recovery Stage:	
Dataset	yahma/alpaca-cleaned
Epoch	2
Batch Size	32
Learning Rate	1e-4
LoRA Rank	16
LoRA Alpha	32
LoRA Dropout	0.05
LoRA Target Modules	in_proj and out_proj

Table 8: Hyper-parameters used in the experiments.

B Hyperparameters

Table 8 offers a detailed summary of the hyperparameters employed in our experiments, promoting both reproducibility and clarity.