ECE1512 Project B Report
Part 1 - Mamba

Rémi Grzeczkowicz
MScAC Student
University of Toronto - Department of Computer Science
Student Number: 1010905399
remigrz@cs.toronto.edu

I Summary of the Mamba Model

Mamba was introduced in [8] with the goal of enhancing the performance of the Transformer model, particularly in language modeling tasks. The core idea behind Mamba is to replace the self-attention mechanism of the Transformer with a novel mechanism known as the *Mamba mechanism*. This new approach can match the performance of the Transformer while significantly reducing both training and inference times. The architecture of Mamba is illustrated in Figure 1.

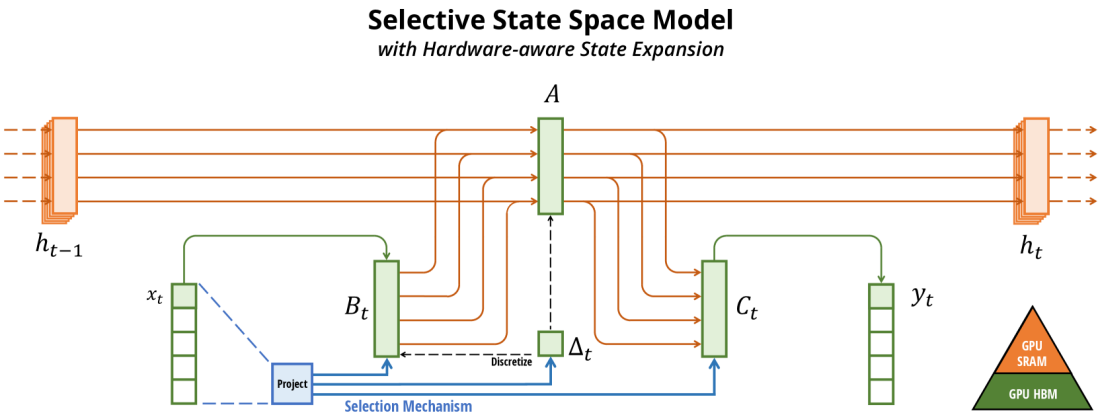

Fig. 1: Mamba architecture

A. Mamba key ideas

As previously mentioned, Mamba introduces a novel class of selective state-space models (SSMs) that address the inefficiencies of Transformers and prior subquadratic architectures for long-sequence tasks across various domains (language, audio, genomics). Its key ideas are as follows:

- **Linear time complexity**: Mamba achieves linear time complexity with respect to sequence length during training, a significant improvement over the quadratic time complexity of the Transformer model. This mechanism also ensures constant time complexity during inference.
- **Selective mechanism**: The model incorporates input-dependent SSM parameters, enabling content-aware information propagation and dynamic filtering of irrelevant data. This selective mechanism mitigates the issues of vanishing and exploding gradients typically encountered by RNNs. Additionally, it enables the model to focus on relevant information, whereas RNNs are forced to propagate all information (or fail to do so if parameters are misconfigured).
- **Hardware-aware algorithm**: The algorithm is designed to minimize the number of transfer operations between CPU and GPU memory, which is often a bottleneck in various models, including Mamba.
- **Empirical success**: The Mamba model has been evaluated on tasks such as language modeling, DNA sequence modeling, and audio processing. It has either matched or outperformed previous state-of-the-art models in terms of performance, while offering significant speed improvements.
- **Open-source implementation**: Although not directly related to the model itself, the authors have released an open-source implementation of Mamba as a Python package available for installation via pip, with all the code provided on GitHub. This enables the scientific community to reproduce the results and integrate the model into their own projects.

B. Technical contributions

Mamba, through its technical implementation, offers several contributions to problems that were present in previous models:

- **Input-Dependent SSMs**:
  - Problem: Traditional Structured State Space Models (SSMs) are Linear Time Invariant (LTI), which makes them computationally efficient but unable to adapt to varying content, particularly in discrete or information-dense data like text.
  - Solution: Mamba introduces a selection mechanism by making key parameters Δ (state update rate), B (input projection), and C (output projection) input-dependent. This allows the model to focus on relevant inputs while dynamically ignoring irrelevant ones. For example, in tasks like selective copying, the model selectively propagates relevant tokens, overcoming the LTI limitation.
  - Impact: This mechanism enables Mamba to combine the expressiveness of Transformers with the efficiency of recurrent architectures.
- **Efficient Scan Implementation**:
  - Problem: The introduction of input-dependent parameters removes the time-invariance property, making traditional convolution-based computation methods inapplicable and increasing memory usage.
  - Solution: Mamba employs a hardware-aware scan algorithm optimized for GPUs:
    - **Kernel Fusion**: Reduces memory transfers between GPU levels by performing all operations within faster memory (e.g., SRAM). During backpropagation, intermediate states are recomputed instead of being stored, significantly reducing memory consumption.
    - **Parallelized Scan**: Mamba implements a parallelized scan operation that avoids the sequential nature of traditional recurrent computations.
  - Impact: The model achieves true linear scaling, outperforming standard LTI convolutional implementations and even advanced Transformer optimizations (e.g., FlashAttention) for sequence lengths beyond 2k.
- **Selection Mechanism**:
  - **Dynamic Parameterization**: Δ controls how much to update or retain the state, functioning as a generalized RNN gate. Large Δ values reset the state to focus on new inputs, while small Δ values retain previous information.
  - **Architectural Simplification**: B and C modulate the influence of input and state, respectively, providing fine-grained control over information flow. Mamba integrates these selective mechanisms into a single homogeneous block, eliminating the need for interleaved MLP or convolutional layers.
  - **Generalization**: Mamba uses SiLU (Swish) activation for smooth gating and consistent performance. This mechanism is broadly applicable, enabling Mamba to adapt across diverse modalities such as language, genomics, and audio.

C. Areas of improvement

Even though Mamba shows promise, there are still some areas for improvement that could be addressed in future work:

- **Continuous-Discrete Spectrum**: Structured State Space Models (SSMs) were originally designed as discretizations of continuous systems, giving them a strong inductive bias for continuous-time data modalities such as audio or video signals. Mamba's selection mechanism, by introducing content dependency, improves performance on discrete data but could potentially hinder performance on continuous data, where LTI SSMs excel. Ablation studies on audio waveforms suggest that the selection mechanism may impair performance in such cases. Future research could explore ways to adapt the selection mechanism to better handle continuous data.
- **Downstream Applications**: While Mamba does not aim to encompass as many use cases as the ecosystem of Transformer-based foundation models, particularly LLMs, the latter provides a variety of features and interaction modes, such as fine-tuning, in-context learning, instruction tuning, quantization, and more. It remains to be seen whether Mamba possesses similar properties and affordances, and whether it can fully replace Transformers in those contexts.
- **Scaling**: Empirical evaluation of Mamba has been limited to small-scale models, below the threshold of most open-source LLMs like GPT-Neo and other recurrent models like RWKV, which have been evaluated at scales of 7 billion parameters and beyond. Whether Mamba remains competitive at such larger scales is yet to be determined. Scaling Mamba may also involve engineering challenges and model adjustments that have not been addressed in the current work.

II Accuracy improvement for translation

Language modeling can be applied to various tasks, including translation. [6] demonstrated that incorporating dropout into a model based on Transformers improves its BLEU score—a standard metric for evaluating translation quality—particularly in low-resource settings. This is because dropout acts as a regularization technique, helping the model generalize better and reduce overfitting. Building on this insight, we could explore integrating dropout into the Mamba model to evaluate its impact on performance. Given that Mamba has been utilized as a relatively small model in [8], this addition might enhance its generalization capabilities.

A. New Architecture of Mamba

The proposed architecture is shown in Figure 2. Two layers of dropout are added to the Mamba model to prevent overfitting and improve generalization.

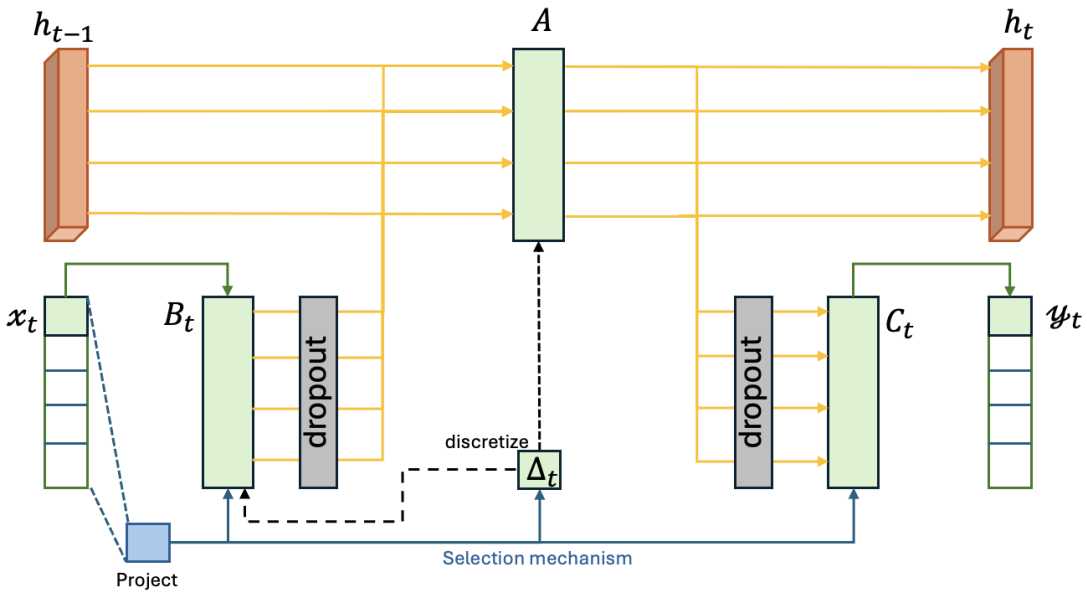

Fig. 2: Mamba architecture with dropout

B. Experimental Setup

To evaluate the impact of dropout on the Mamba model, we will run experiments on translation tasks and compare the results between the transformer encoder-decoder model, the transformer with dropout from [6], the original Mamba model, and the Mamba model with dropout. We will use the BLEU score as the evaluation metric.

For the dataset, we will use the Europarl dataset [10], a parallel corpus of 21 languages. We will specifically use the English-French dataset for our experiments. Additionally, we will evaluate the model's generalization abilities by comparing translation performance between languages with similar origins (e.g., Germanic languages) and languages with different origins (e.g., Germanic and Slavic languages).

Furthermore, following the recommendation from [11], we will ensure that the models are of similar size and complexity to allow for a fair comparison. Specifically, the models will have around 77M parameters to match the number of parameters in the transformer model [11].

Finally, since dropout layers are generally more beneficial for smaller models [6], we will also evaluate the impact of dropout on the Mamba model at different sizes to determine if this effect persists across different model scales.

III Extension of the Mamba Model to Image Classification

Transformers, through models like Vision Transformer (ViT) [5], have shown great success in image classification tasks. [9] also described a broader use of transformers. In the original work on Mamba, the model was tested on language modeling, DNA sequence modeling, and audio modeling tasks. We propose to extend the Mamba model to image classification tasks to evaluate its performance in this domain.

A. Experimental Setup

To evaluate performance on image classification, we propose running experiments on CIFAR-10 and CIFAR-100 [4]. Later, for technical reasons, we chose to use MedMamba [14] as the classifier. Since MedMamba was specifically developed to classify medical images, we will also test its performance on the MHIST dataset [13]. This dataset contains medical images and is more representative of the type of images that MedMamba was designed to classify. However, the authors of [14] did not evaluate the model on this dataset, making it an interesting case for assessing MedMamba's ability to generalize to other medical images, and more broadly to general images.

We will train the model on the CIFAR-10 and CIFAR-100 datasets and evaluate it on the test set for 500 epochs. We will use accuracy as the evaluation metric and compare the results with state-of-the-art models for image classification tasks. Additionally, we will evaluate the performance of the model on the MHIST dataset.

The code for the experiments is available on GitHub at https://github.com/jacedoir/ECE1512_2024F_ProjectRepo_Grzeczkowicz in the folder *Project B/MedMamba*. The code is not modified from the original version provided by the authors of [14].

B. Results

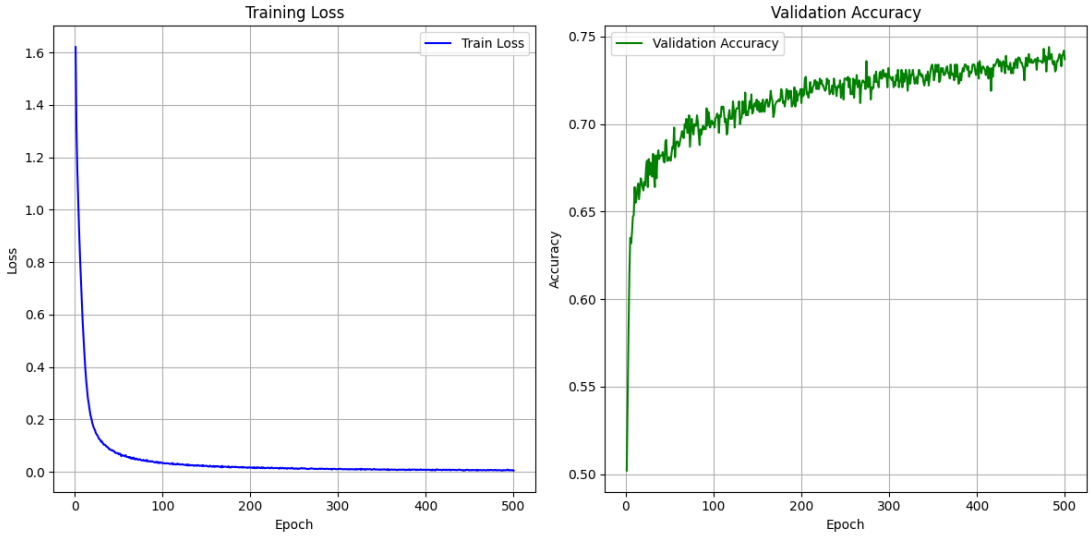

Fig. 3: Results on CIFAR-10 dataset

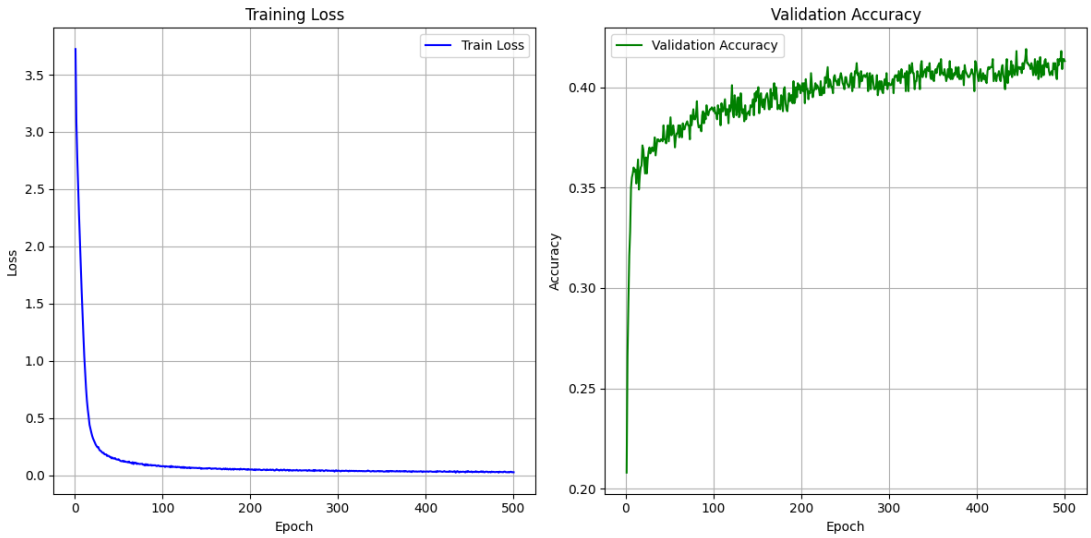

Fig. 4: Results on CIFAR-100 dataset

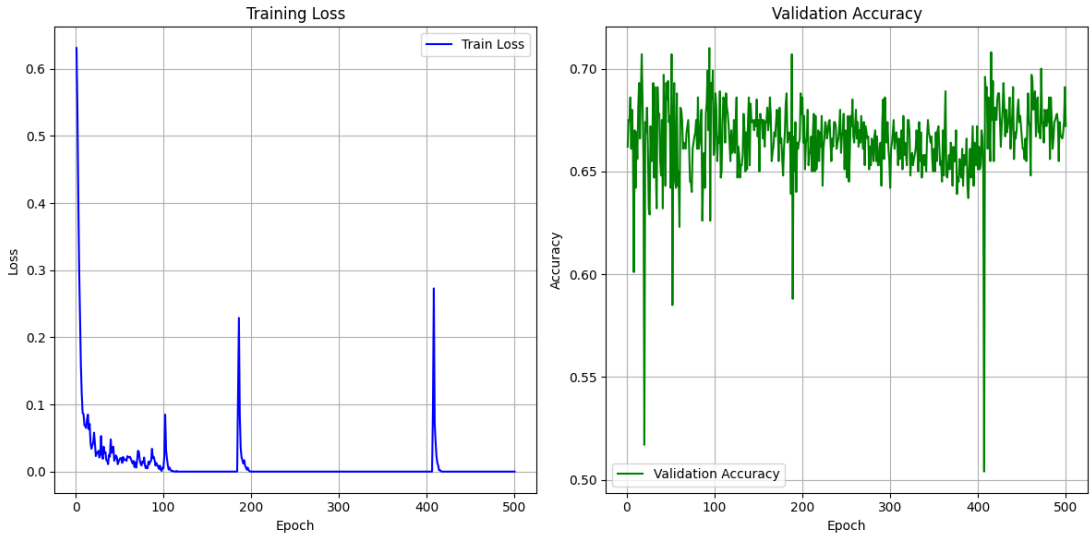

Fig. 5: Results on MHIST dataset

The results of the experiments are not satisfactory. The model is unable to classify images correctly and underperforms compared to state-of-the-art models. Specifically, for CIFAR-10, the best accuracy achieved is 0.744, reached at epoch 481 (see Figure 3). According to [1], the best model, ViT-H/14 (Vision Transformer) [5], achieves an accuracy of 0.995 after 14 epochs. For CIFAR-100, the best accuracy is 0.419, reached at epoch 455 (see Figure 4). According to [2], the best model is EffNet-L2 + SAM [7], a model based on CNN developed by [12] and enhanced by [5] using Sharpness-Aware Minimization, which achieves an accuracy of 0.9608. Finally, for the MHIST dataset, the best accuracy is 0.71, reached at epoch 93 (see Figure 5). According to [3], the best model is MoCo-V2 [7], which achieves an accuracy of 0.8803.

The difference in performance between the state-of-the-art models and MedMamba is smaller on the MHIST dataset, suggesting that the model may be more suited to medical images. However, it still underperforms, which could be due to the fact that Mamba may be interpreted as an RNN, and thus may require the data to be sequential, as is the case in domains like DNA, text, and audio. This is not true for image classification tasks, where data is typically not sequential. This discrepancy could explain why the model is underperforming on image classification tasks.

IV Conclusion

In this study, we explored the performance of the Mamba model in various domains, including language modeling, DNA sequence modeling, and image classification. While Mamba demonstrated promising results in sequential tasks, such as language and audio modeling, its performance on image classification tasks, including CIFAR-10, CIFAR-100, and MHIST, was suboptimal when compared to state-of-the-art models. The underperformance in image classification may be attributed to Mamba's inherent design, which is more suited for sequential data, as seen in its success with DNA, text, and audio data. The addition of dropout did not significantly improve the generalization capabilities of Mamba on these tasks.

Future work could focus on adapting the Mamba model to better handle image data, potentially by modifying its architecture to incorporate image-specific features or by investigating hybrid approaches that combine the strengths of both sequential and convolutional models. Additionally, further exploration into the scalability of Mamba at larger model sizes could provide insights into its performance on more complex datasets and tasks. Despite the challenges, the results presented here offer valuable insights into the potential applications of Mamba and highlight areas for improvement to enhance its applicability across diverse domains.

References

[1] Papers with Code - CIFAR-10 Benchmark (Image Classification) — paperswithcode.com. https://paperswithcode.com/sota/image-classification-on-cifar-10. [Accessed 28-11-2024].

[2] Papers with Code - CIFAR-100 Benchmark (Image Classification) — paperswithcode.com. https://paperswithcode.com/sota/image-classification-on-cifar-100. [Accessed 28-11-2024].

[3] Papers with Code - MHIST Benchmark (Classification) — paperswithcode.com. https://paperswithcode.com/sota/classification-on-mhist. [Accessed 28-11-2024].

[4] Krizhevsky Alex. Learning multiple layers of features from tiny images. *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*, 2009.

[5] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Carlos Escolano, Francesca De Luca Fornaciari, and Maite Melero. Residual dropout: A simple approach to improve transformer's data efficiency. In *The 3rd Annual Meeting of the ELRA-ISCA Special Interest Group on Under-resourced Languages@ LREC-COLING-2024 (SIGUL 2024): Turin, Italy*, pages 294–299. ELRA Language Resources Association and the International Committee on ..., 2024.

[7] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.
[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024.
[9] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.
[10] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86, 2005.
[11] Hugo Pitorro, Pavlo Vasylenko, Marcos Treviso, and André F. T. Martins. How effective are state space models for machine translation?, 2024.
[12] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
[13] Jerry Wei, Arief Suriawinata, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Louis Vaickus, Charles Brown, Michael Baker, Naofumi Tomita, Lorenzo Torresani, et al. A petri dish for histopathology image analysis. In *Artificial Intelligence in Medicine: 19th International Conference on Artificial Intelligence in Medicine, AIME 2021, Virtual Event, June 15–18, 2021, Proceedings*, pages 11–24. Springer, 2021.
[14] Yubiao Yue and Zhenzhang Li. Medmamba: Vision mamba for medical image classification. *arXiv preprint arXiv:2403.03849*, 2024.