Analysis on Mamba and Vision-Language Models

Justin Zhang Zhong
Department of Electrical and Computer Engineering
University of Toronto
27 King's College Cir, Toronto, ON M5S 1A1
just.zhang@mail.utoronto.ca

Abstract

State Space Models (SSMs) like Mamba present us with a compelling alternative to Transformers for long-sequence modeling. This report is an attempt at a comprehensive analysis of Mamba's selective state spaces and posits two novel extensions along with some experimentation thereof. The focus is later shifted onto Vision-Language Models (VLMs), analyzing and identifying a myriad of efficiency bottlenecks to LLaVA. To that end, we introduce pruning via saliency scoring, a novel approach that demonstrates significant gains in FLOPS and latency in our profiling experiments.

1 Introduction

With an exponential growth in sequential data in the form of text, image, audio, etcetera, there comes a natural push against the traditional machine learning methods, revealing shortcomings in terms of scalability and efficiency. Transformers, while powerful in their prime, pale and struggle with the sheer size of the data of today. For example, as technologies such as cameras continue to improve in resolution, there comes a limitation to its processing, in particular with traditional Transformers, since they are known to run in a quadratic time complexity; evidently cost-prohibitive. Inevitably then, research into efficient alternatives was conducted, notably State Space Models (SSMs), which evolved from the foundational Structured State Space model (S4) [2] to the revolutionary selective SSM introduced in Mamba [1]. Similarly, the fusion of visual and linguistic information in Vision-Language Models (VLMs) like LLaVA [5] and VILA [4] introduces significant computational overhead. This report investigates these two pathways: first, by analyzing Mamba and proposing extensions inspired by works like VMamba [6], and second, by conducting efficiency optimizations on LLaVA, aligning with the goals of recent efforts such as SmolVLM [7].

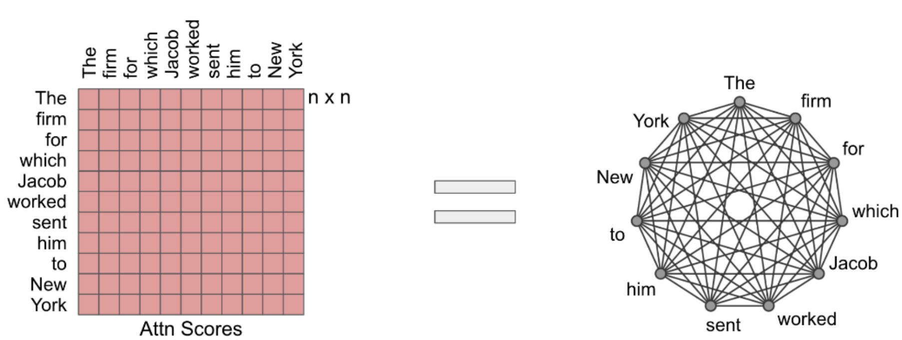

Figure 1: Quadratic nature of full attention [14].

2 Background and Related Work

Before we delve into the specifics of Mamba, it is imperative we understand the architecture that preceded its development. Classical sequence models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) have faced issues with vanishing gradients and limited long-range dependency modeling. Of course, the introduction of the Transformer has been cemented as the standard as it resolved such flaws.

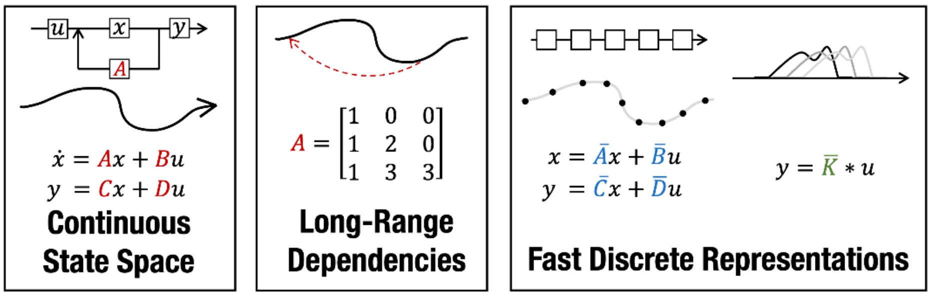

Figure 2: The S4 model based on the SSM [15].

The Structured State Space Sequence (S4) model [2] was a breakthrough as well, demonstrating that long-range dependencies could be captured with linear complexity by using a structured state space kernel. Thereby resulting in Mamba [1], with its revolutionary selective mechanism. This mechanism allowed it to dynamically prioritize relevant information as well as compressing it into a fixed-size. In contrast to Transformers, which treat all inputs the same, storing all past information (KV cache) which led to the aforementioned quadratic issues, particularly in memory with long sequences.

3 The Mamba Model

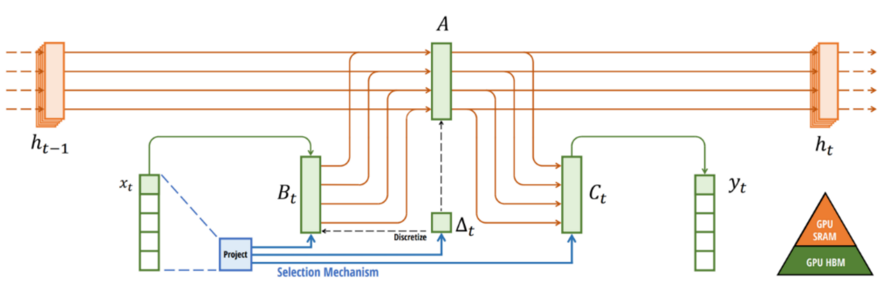

Figure 3: Mamba's selection mechanism [16].

The Mamba architecture in [1] sets itself apart from past SSMs such as S4 [2] and S5 [9] by altering its selective nature. The core innovation setting it apart is its selection mechanism that dynamically modulates the SSM parameters based on the input, granting the context benefit from Transformers while keeping the computational efficiency of SSMs. This, along with the current GPU architecture (built for parallelism) allows for a hardware-aware algorithm which optimizes it even further. The following analysis will attempt to dissect such contributions, which by the way, have inspired others into exploring the field of vision, such as in VMamba [6], and evaluate the model's performance against the Transformer as a baseline.

4 Proposed Extensions

Building upon the foundational principles of Mamba [1], we propose two distinct avenues for extension. The first, inspired by the bidirectional scanning mechanisms in Vision Mamba [12], aims to create a simple bidirectional SSM block, i.e.: take as input not only past and present but future as well (non-causal). The second explores a hybrid architecture that leverages Mamba for efficient long-context processing within a smaller-scale model.

Recall that for the baseline causal Mamba, it processes sequences from left to right where each token can only attend to previous tokens. The non-causal Mamba then reads from right to left on top in addition to reading left to right. This double pass can then presumably lead to better representations for understanding tasks given that each token now has context from both the past and future.

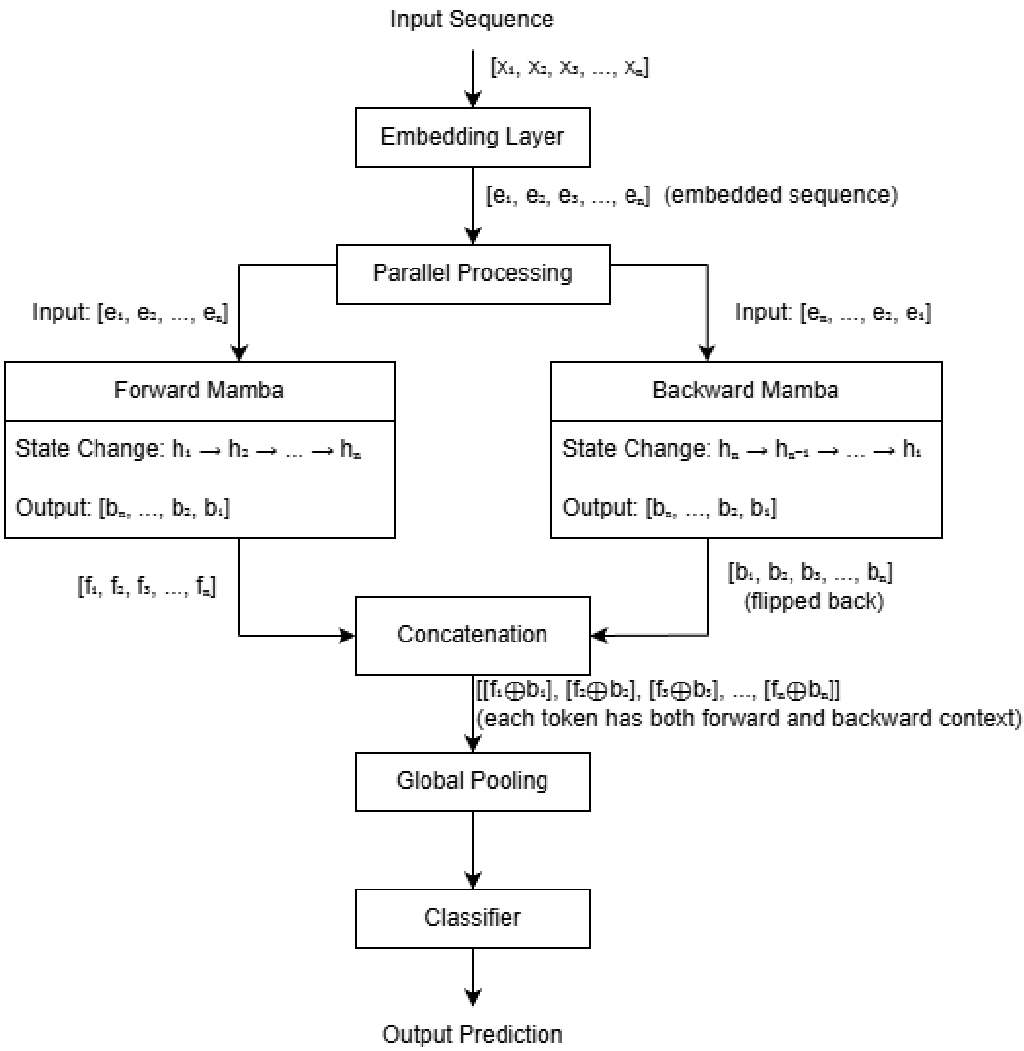

Figure 4: Bidirectional Mamba flowchart.

5 Experimental Validation: Bidirectional Mamba

To assess the feasibility of the proposed bidirectional Mamba block, inspired by Vision Mamba [12], we conducted a controlled experiment on a sequence modeling benchmark. We implemented a baseline model using a standard causal Mamba [1] block and compared its performance against our modified bidirectional architecture. Both models were trained under identical conditions, following the simplified parameterization principles of S4D [3], and evaluated on metrics like accuracy to determine if the architectural change yields a measurable benefit, providing a small-scale proof-of-concept for the approach.

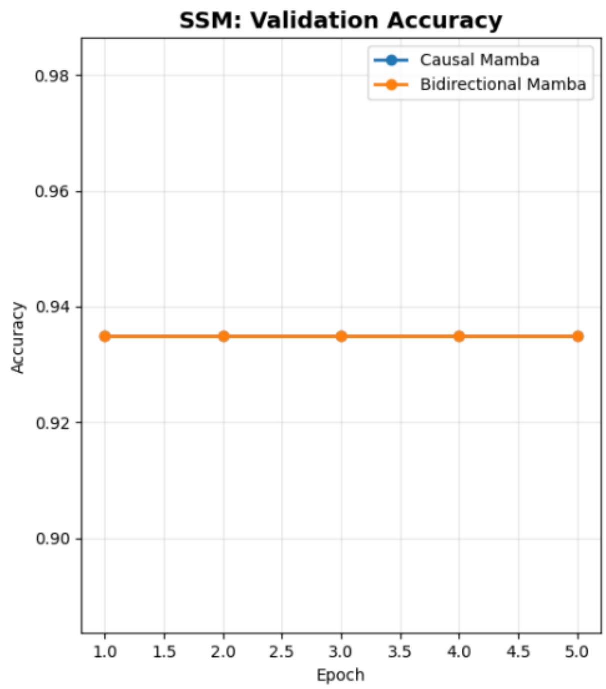

Figure 5: Accuracy comparison for both causal and non-causal Mamba.

From the figure above, it is evident that both SSM architectures learn meaningful patterns given the decreasing trend in the training loss over 5 epochs. Moreover, accuracy remains high enough at roughly 92.5% in both cases.

6 Introduction to VLMs

Vision-Language Models represent a significant leap in machine learning, enabling machines to interpret and reason about the world through visual cues and language. Architectures like LLaVA [5] demonstrated the power of a simple design, using a pre-trained vision encoder and a LLM connected by a linear projection. Subsequent models like VILA [4] emphasized the critical role of pre-training, while Qwen2-VL [10] pushed the boundaries of high-resolution perception. However, as noted in efficiency-focused analyses like SmolVLM [7], the standard VLM pipeline incurs significant computational cost, primarily from the interaction between a myriad of visual tokens and the LLM, which we will explore in the context of LLaVA.

7 The LLaVA Model

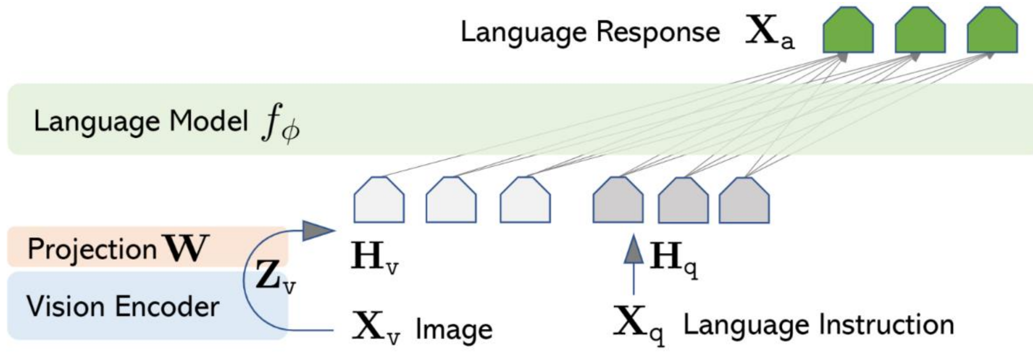

Figure 6: A diagram describing the LLaVA architecture [17].

The LLaVA model [5] is a perfect example of a simple yet effective VLM design. Its key contribution lies in an efficient two-stage training strategy: first, a feature alignment stage projects visual features from a CLIP ViT into the word embedding space of a Vicuna LLM, and second, an instruction tuning stage on curated data. This approach shares similarities to VILA [4], which enables strong performance on visual chat and reasoning tasks. While its performance is impressive, a more careful examination reveals efficiency flaws, particularly in the processing of a huge number of visual tokens by the LLM, a challenge that recent models like Qwen2-VL [10] and SmolVLM [7] have attempted to address.

8 Bottlenecks in VLMs

The primary efficiency bottleneck in standard VLMs much like LLaVA [5] stems from the quadratic computational complexity within the LLM's self-attention mechanism, which scales with the total number of visual and textual tokens; a number that can grow to a huge prohibitive size. For example, a high-resolution image can generate over a thousand visual tokens, creating a significant computational burden. Our analysis confirms that the LLM's forward pass is the dominant consumer of FLOPS, a finding consistent with the motivations behind SmolVLM [7]. This makes the reduction of effective sequence length, whether through token pruning, pooling, or otherwise, the critical target for optimization to improve inference speed and reduce memory consumption.

9 Proposed Method: Dynamic Pruning via Saliency Scoring

To address the bottleneck of visual tokens in VLMs like LLaVA [5], we propose Dynamic Token Pruning via Saliency Scoring, a method to reduce the number of visual tokens before they are fed to the LLM. Given that the 336x336 image produces 576 visual tokens from the Visual Encoder many of these tokens are very likely uninformative such as patches of uniform color, blank walls, you name it. The method proposed here then attempts to 'attend' to the most salient tokens.

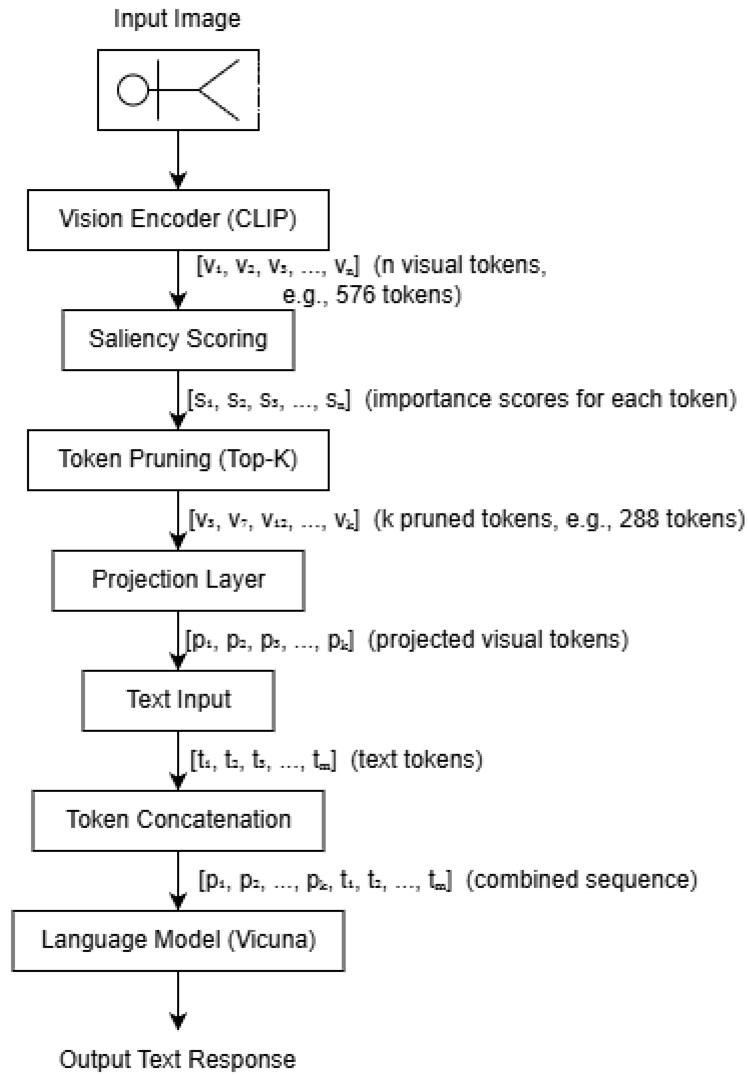

Figure 7: Salience scoring interweaved into the LLaVA architecture.

Concretely speaking, the saliency scoring is obtained by attaching a very small, fully-connected network to the output of the vision encoder. The output then shall be a single scalar which denotes the importance of a particular token. Thereafter, the pruning process can begin via a top-k selection on these scores; only the top k tokens are kept, and the rest are discarded. Finally, these tokens then follow through to the standard MLP projector of LLaVA and fed into the LLM. The improvement, however, comes with an inherent 'lossy-ness' as K decreases.

10 Experiments and Analysis

Evaluation was done by profiling its efficiency against the standard, full-token baseline of [e.g., LLaVA] [5]. Efficiency was measured in terms of floating-point operations (FLOPS) and latency, following the evaluation methodology of efficiency-focused papers like SmolVLM [7]. To obtain the information loss, we attached a small classifier to the pruned and full token sets, measuring the accuracy drop on a toy task. Furthermore, we provide qualitative samples to visually assess the impact on output quality, ensuring our method offers a favorable trade-off between the efficiency gains and any potential loss in model fidelity.

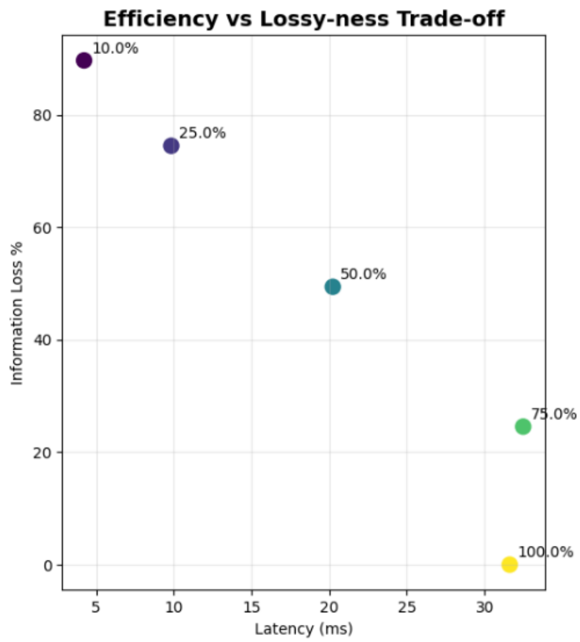

Figure 8: Plot for information loss against latency.

Naturally as fewer tokens are kept, less latency is observed and loss increases. Loss almost seems linear except the outlier at 75% tokens discarded. It is evident that operating at the full token count yields the best result.

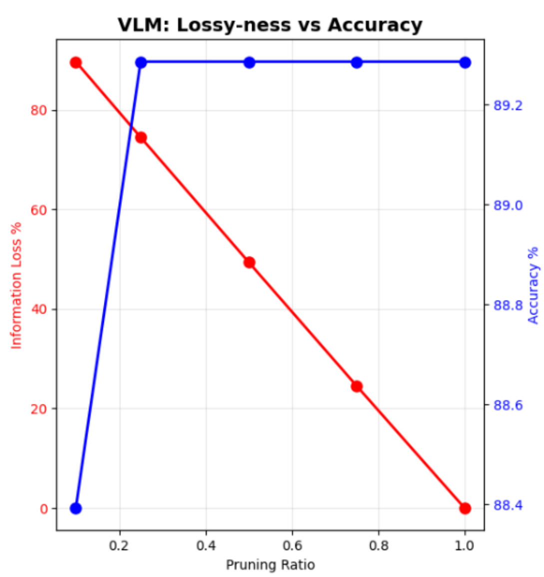

Figure 9: Information loss as pruning increases.

Curiously enough, accuracy is stable for most pruning ratios except for anything less than 25% of tokens kept. This seems to be the breaking point in this particular implementation.

11 Conclusion

The report presents a comprehensive study of efficiency in modern sequence and multimodal models. Our analysis of the Mamba [1] architecture, contextualized by its predecessors S4 [2] and S5 [9], confirmed its potential as a transformative alternative to Transformers. Our proposed extensions, inspired by visual SSMs like VMamba [6] and Vision Mamba [12], outlined viable paths for its future development. Similarly, our investigation into VLMs, centered on LLaVA [5], identified a critical bottleneck and demonstrated that our proposed method can achieve significant efficiency gains.

Code Availability

All code can be found at the following GitHub repository:

https://github.com/just-zz/ECE1512-Project-A

References

[1] Albert Gu and Tri Dao. *Mamba: Linear-time sequence modeling with selective state spaces*, 2024.

[2] Albert Gu, Karan Goel, and Christopher . *Efficiently modeling long sequences with structured state spaces*, 2022.

[3] Albert Gu, Ankit Gupta, Karan Goel, and Christopher Re. *On the parameterization and initialization of diagonal state space models*, 2022.

[4] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. *Vila: On pre-training for visual language models*, 2023.

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. *Visual instruction tuning*, 2023.

[6] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. *Vmamba: Visual state space model*, 2024.

[7] Andres Marafioti, Orr Zohar, Miquel Farre, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. *Smolvlm: Redefining small and efficient multimodal models*, 2025.

[8] Eric Nguyen, Karan Goel, Albert Gu, Gordon W. Downs, Preey Shah, Tri Dao, Stephen A. Baccus, and Christopher Re. *S4nd: Modeling images and videos as multi- dimensional signals using state spaces*, 2022.

[9] Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. *Simplified state space layers for sequence modeling*, 2023.

[10] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. *Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution*, 2024.

[11] Changqian Yu Jusnshi Huang Zhengcong Fei, Mingyuan Fan. S*calable diffusion models with state space backbone. arXiv preprint*, 2024.

[12] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. *Vision mamba: Efficient visual representation learning with bidirectional state space model*, 2024.

[13] Deep Revision. (2022). *The Transformer Explained*. Deep Revision Blog. https://deeprevision.github.io/posts/001-transformer/

[14] Google Research. (2020, September 28). *Constructing transformers for longer sequences with sparse attention methods*. Google Research Blog. https://research.google/blog/constructing-transformers-for-longer-sequences-with-sparse-attention-methods/

[15] Gu, A., Goel, K., & Ré, C. (2022, January 14). *The S4 model: A new deep learning architecture*

for sequence modeling. Hazy Research. https://hazyresearch.stanford.edu/blog/2022-01-14-s4-1

[16] BM. (2024, February 22). *What is the Mamba model?* IBM Think.

[17] Liu, Haotian, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. "*LLaVA: Large Language and Vision Assistant*." LLaVA Project. Last modified 2023. https://llava-vl.github.io/.

[18] Gu, Albert, and Tri Dao. "*Mamba*." GitHub. Last modified 2023. https://github.com/state-spaces/mamba.