Efficiency Meets Translation Quality: Evaluating Mamba Against Attention in Neural Machine Translation

Yihang Chen

Master of Science
School of Informatics
University of Edinburgh
2024

Abstract

Neural Machine Translation (NMT) plays a crucial role in the advancement of natural language processing and the real world. However, competitive NMT systems predominantly rely on Transformer architectures, which face significant computational and memory bottlenecks when handling long sequence tasks. The recently proposed Mamba model is regarded as a hidden-attention model with linear complexity concerning sequence length, offering a promising solution to the computational overhead associated with the Transformer. This project aims to determine whether the Mamba model can effectively replace attention mechanisms in NMT systems, aiming to achieve greater efficiency while maintaining translation quality.

This project implements a Mamba-based model (Mamba Base) and a Mamba with Attention Model (MA), evaluating their performance on the WMT14 English-German Dataset. The results indicate that the Mamba Base model achieves a 5-7× increase in inference speed compared to the Transformer, although it struggles with capturing long sequence dependencies. In contrast, the MA model incorporates cross-attention and achieves translation quality comparable to the Transformer while maintaining the efficiency of the Mamba model. Additionally, this project analyzes specific linguistic phenomena and visualizes attention distributions, revealing that the MA model captures local features better and exhibits clearer and more uniform attention patterns than the Transformer. This research demonstrates the potential of the Mamba model to replace self-attention and MLP in Transformer-based NMT systems with higher efficiency.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yihang Chen*)

Acknowledgements

I would like to express my deepest gratitude to my supervisors, John Pate, Prarit Agarwal, Salman Khan Pathan, and Tania Bakhos, for their invaluable guidance, support, and encouragement throughout this project. Their expertise and insights have been instrumental in shaping the direction and success of my research.

I am also sincerely thankful to Professor Chris Williams, Professor Iain Murray, Eddie's support team, and Cirrus's support team for providing the essential computing resources that enabled the successful completion of this research. Their technical support was crucial in overcoming the computational challenges of this project.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Machine translation (MT) has long been crucial to bridging language divides and facilitating more efficient cross-cultural communication in this globalised society. The advent of Neural Machine Translation (NMT) has marked a tremendous advancement in the MT field, dramatically enhancing the accuracy and fluency of translations compared to traditional statistical machine translation (SMT). NMT leverages deep learning technologies to grasp better the nuances and long dependencies of sequence, which can generate more natural and coherent translations. In recent years, the rapid development of large language models (LLMs) has provided a new pretrain-prompt paradigm for the MT field. By training with large multilingual corpora, LLMs freed it from the limitations of parallel data and also achieved dominant performance. These technologies propose efficient alternatives to human translation, which, despite its accuracy, can be sluggish and costly. By automating the translation process, MT enables instant communication and information access at a low cost. However, challenges persist, particularly as translation quality tends to decline noticeably with sentence length increasing [36]. Moreover, progress in MT research has slowed, with only modest advancements in recent years. This slowdown highlights the urgent need for innovative theories and new approaches to achieve breakthroughs and significantly enhance MT systems.

1.2 Problem Statement

Current mainstream NMT systems and LLMs are based on the Transformer [58] architecture, which has brought impressive performance to these methods. However, the complexity of the self-attention layer in the Transformer is $O(L^2D)$ where L is the sequence length, and D is the dimension of word embedding, the attention matrix consumes extensive memory as the context length increases. This issue poses significant challenges for these Transformer-based models regarding computational resources. Additionally, LLMs not only require substantial computational resources but also perform poorly under specific language conditions due to the lack of teacher-forcing supervision from parallel data. Therefore, Gu et al. [23] introduce a novelty Structured State Space Model (S4), which offers an approach to memory almost indefinitely history with finite memory and loss. However, the state space models were indicated that fall short of detecting alignment between source and target sequence in MT due to their time-invariant and input-invariant [57]. Then, recent research by Ali et al. [1] and Gu et al. [10] has shown that a Selective State Space Model (S6) [21] and a related architecture Mamba can function as a high-efficiency linear-complexity hidden-attention model which overcomes the time-invariant character of traditional SSM, proposing its potential to tackle the limitations encountered by both attention mechanisms and S4 in MT. Generally, the main focus of this project is to determine whether the Mamba model, featuring linear-complexity hidden-attention capabilities, can efficiently replace attention in MT while maintaining competitive translation quality.

1.3 Why NMT instead of LLM?

As the two main MT task solutions, NMT systems and LLMs have achieved dominant performance in recent years. However, the purpose of this project is to explore the potential of the Mamba model as an efficient hidden attention model to replace the traditional attention mechanism in MT tasks. In this context, an NMT system provides a more suitable foundation for investigation than LLMs. The main reason is that given the source sequence S and the target sequence T, NMT systems model the conditional probability $p(T|S)$, which clearly reflects the model's ability to handle challenges in MT tasks like capturing the dependencies between the source and target sequences. On the other hand, LLMs model the joint probability $p(S,T)$. Applying the Bayesian theorem to expand the joint probability results in $p(S,T) = p(T|S)p(S)$, where calculating $p(S)$ introduces unnecessary resource consumption. LLMs achieve strong performance on MT tasks primarily through extensive training data and large model sizes, which might obscure the specific mechanisms and performance differences of models in the project. Therefore, compared to LLMs, NMT systems provide a better architecture for this project, as they more clearly reflect the model's specific ability for MT tasks, such as detecting dependencies between source and target sequences.

1.4 Research Hypothesis and Objectives

The primary question this project seeks to answer is whether the Mamba model, with its high efficiency, hidden-attention property and linear complexity concerning sequence length, can effectively replace traditional attention mechanisms in NMT systems. Specifically, the project aims to determine if the Mamba model can provide more efficient memory usage and faster inference and training speeds while maintaining translation quality. Here are the main objectives to address the main question:

1. Research existing methodologies and datasets of the NMT system to select a dataset and establish a baseline for comparison.
2. Design and implement an NMT system incorporating the Mamba model.
3. Train the implemented NMT system with the selected dataset and evaluate its quality of translation and efficiency.
4. Integrate cross-attention into the Mamba NMT model to compare Mamba's ability with self-attention and conduct ablation experiments.
5. Visualize the implicit attention in the Mamba model and analyze the models' attention distribution.
6. Analyze the model's ability to handle different linguistic phenomena based on the translation results of the implemented model.

1.5 Timeliness, Novelty, and Significance

The Mamba model was introduced in December 2023 and is considered a strong challenger against the Transformer due to its faster inference speed and scalability in handling long sequences. Currently, Mamba has emerged as a trending research direction within the deep learning community, demonstrating impressive performance across various domains such as image processing [35], language modeling [21], and multi-modal tasks [48]. However, no research has yet focused on applying Mamba to MT tasks except this project. It is noteworthy because MT is a core task in natural language processing (NLP) that requires the model to effectively capture features of the target sequence while also addressing dependencies between the source and target sequences. Achieving this necessitates a model with robust capabilities for capturing long-term dependencies, feature representation, and generalization. Additionally, fitting Mamba into an encoder-decoder architecture remains an open problem. This project developed an encoder-decoder NMT system based on the Mamba model, which achieves competitive translation quality compared to the Transformer while boasting faster inference speed.

The insights gained from tackling complex dependencies during MT research could have broader implications for other tasks, encouraging the development of models to address intricate dependencies in various applications better. Additionally, the Mamba model's linear complexity regarding sequence length enables it to tackle long-sequence tasks, such as document-level translation effectively. This capability addresses the computational and memory bottlenecks previously faced by NMT systems that utilize the Transformer architecture. By exploring Mamba's performance and effectiveness in MT, this project aims to contribute valuable insights that could advance the state of the art in both translation quality and efficiency.

1.6 Contributions and Result

The contributions of this project are summarized as follows:

- A study was conducted on existing NMT models and datasets, selecting the encoder-decoder auto-regressive model as the foundational architecture, with Transformer as the baseline, specifically utilizing the WMT14 EN-DE dataset.

- A Mamba-based encoder-decoder model (Mamba Base Model) was implemented and evaluated for translation quality and model efficiency, achieving 5-7× faster inference speed compared to Transformer, although it faced challenges when translating long sentences.

- By employing cross-attention in the Mamba Base Model to create a Mamba with Cross-Attention model (MA model), a similar translation quality to the Transformer was achieved while maintaining the similar efficiency of the Mamba Base.

- The translation results were analyzed on linguistic phenomena, revealing that the MA model better handled unseen words and constituent ambiguity than the Transformer but struggled with complex sentence structure dependencies.

- The model's attention distribution was visualized, demonstrating that the MA model exhibited a clearer and more uniform attention distribution than the Transformer, indicating Mamba's superior ability to handle fine-grained features.
- Prove the potential of Mamba to replace the self-attention layer, along with a discussion of using diffusion models instead of auto-regressive models, potentially allowing for the Mamba-based block to substitute for cross-attention.

1.7 Overview of the Thesis

The remainder of this thesis is organized as follows:

- **Background:** This chapter covers the preliminary knowledge necessary to understand this thesis, including the architectural details of Transformers and certain SSM models. It also discusses previous research and datasets in the field of NMT.
- **Methodology and Implementation:** This chapter introduces and justifies the model used in this project, including the Baseline model Transformer, the Base encoder-decoder Mamba model implemented by this project, and the Mamba encoder-decoder model with attention. Additionally, this chapter also introduces the evaluation metrics and visualization techniques.
- **Training Detail:** This chapter presents the detailed configurations regarding the model training process, including the dataset selection, data preprocessing methods, software and frameworks used for training, hardware platforms, and hyperparameter settings for the model.
- **Result and Analysis:** This chapter presents the evaluation results regarding the models' translation quality and efficiency, followed by a detailed analysis. Additionally, the project analyzes the model's ability to handle various linguistic phenomena through specific examples. The chapter also visualizes the model's attention distribution and showcases the results of the ablation experiments.
- **Conclusion and Further Direction:** This chapter concludes this project's result, findings, and contribution and suggests future research directions.

Chapter 2

Background and Related Works

2.1 Background: Transformer

Transformer is a revolutionary deep learning network originally used for NMT [58]. Then, it quickly became the dominant architecture in computer vision (CV), NLP, and even the entire AI field.

Transformer is an encoder-decoder model, and its strong performance is mainly due to its self-attention mechanism, which enables it to capture long-distance dependencies within the sequence. This mechanism allows the model to assess the importance between all elements in a single step, assigning varying weights to each element, allowing it to capture long-distance dependencies, even when elements are distant within the sequence. Such mechanisms not only make the Transformer more accurate and adaptable in handling long sequence data but also enable efficient parallel processing, outperforming the previous dominant recurrent and convolutional models. Cross-attention is also applied in the Transformer to detect the attention between two different sequences. The equation for attention is as follows:

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V = \alpha V, \quad \alpha = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}) \tag{2.1}$$

Where $\alpha \in \mathbb{R}^{B\times L_Q \times L_{KV}}$ is the attention matrix and $Q \in \mathbb{R}^{B\times L_Q\times D}$, $K \in \mathbb{R}^{B\times L_{KV}\times D}$, and $V \in \mathbb{R}^{B\times L_{KV}\times D}$ are the matrices depending on input data, given the batch size B, the length of the sequence Q depended L_Q, the length of the sequence K, V depended L_{KV}, and the dimension of hidden state D. Q, K, and V are matrices obtained by applying linear transformations to the input sequence, which enables the model to be data-dependent. When Q, K, and V depend on the same sequence, the formula represents self-attention. However, when Q represents one sequence and K and V

represent another sequence, it signifies cross-attention. The scaling factor $d_k \in \mathbb{R}$ is the hidden dimension of the matrix K, which is used to keep gradients stable.
However, since the time and space complexity of the attention matrix α for sequence length L is $O(L^2)$, Transformer-based models face serious computational and memory resource bottlenecks when dealing with longer contexts.

2.2 Background: State Space Models to Mamba

2.2.1 State Space Models (SSM)

State Space Models (SSMs) [30] are a class of models designed for processing continuous-sequential data by leveraging the concept of state spaces from control theory. SSMs are defined by two key equations: the state equation (Left in the Equation 2.2) and the output equation (Right in the Equation 2.2). The state equation explicitly maintains a hidden state that evolves over time, while the output equation generates observations based on this hidden state:

$$h(t) = Ah(t-1) + Bx(t), \quad y(t) = Ch(t) + Dx(t) \tag{2.2}$$

Where $t \in \mathbb{R}$ is the current time step, $x(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$ are the 1-D input signal and 1-D output at time step t and $h(t) \in \mathbb{R}^{N\times 1}$ is the hidden state, given dimension of state space N. $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times 1}$, $C \in \mathbb{R}^{1\times N}$ and $D \in \mathbb{R}$ are learnable, time-invariant parameter matrices. For input sequences with hidden dimension D, the SSM is applied to each dimension independently. This model computes the output of the current time step through the previous hidden state and the current input.
Although SSMs have been demonstrated to handle long-range dependencies effectively with the appropriate choice of the state matrix A [22], this approach suffers from excessive computational and memory demands [24], making it impractical as a general solution for sequence modelling.

2.2.2 Structured State Space Models (S4)

To efficiently process long sequential data with SSMs, Gu et al.[23] developed the Structured SSMs (S4). This advancement refines the SSM framework through the introduction of an initialization strategy, a discretization form, and a structured design.

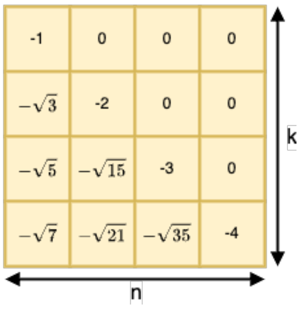
$$h_k = \bar{A}h_{k-1} + \bar{B}x_k, \quad y_k = \bar{C}h_k \tag{2.3}$$

Where h, x, and y are discrete sequences instead of continuous functions of time step k. $\bar{A}$, $\bar{B}$ and $\bar{C}$ means discretized [56] parameter matrix with the same size as basic SSMs, enabling the S4 to handle discrete sequences recurrently. Additionally, since $Dx(t)$ in Equation 2.2 can be regard as a skip connection, the S4 assume $D = 0$ to simplify the model.

The matrix A constructs the current hidden state by leveraging historical information. The S4 model initializes the matrix A using the HiPPO operator [22], enabling it to store almost infinite historical information within a limited memory capacity with minimal compression loss. Besides, the S4 model integrates concepts from both recurrent and convolutional networks to facilitate efficient inference and parallel training. The S4 model reformulates the problem by discretizing the parameter matrix into discrete time steps to process discrete sequences recurrently. At each time step, an update of the hidden state is recurrently involved, which only depends on the current state instead of the entire history. This property endows the S4 model with the ability to infer efficiently. Meanwhile, the S4 model utilizes a convolutional kernel to process input, streamlining intermediate operations and enabling parallel training. The convolution kernel can be precomputed and saved by expanding the expression of y using the state equation to eliminate the hidden state in the output equation. This structured design enables the S4 model to achieve fast training like Convolutional Neural Networks (CNNs) and to perform inference as efficiently as Recurrent Neural Networks (RNNs). The remainder of this section will introduce the details of the HIPPO operator and discretization process:

High-order Polynomial Projection Operator (HiPPO) According to Equation 2.2, matrix A capture information from previous state to build new state. However, the matrix A has a fixed size. With limited memory space to represent long previous states, the previous SSM performed poorly on long sequence modelling tasks. This issue arises primarily due to severe information loss caused by gradient vanishing/exploding during backpropagation. When computing gradients in traditional SSM models, the matrix A is repeatedly multiplied, and unselected values of matrix A can lead to exponential scaling of the gradients over time.

To avoid gradient vanishing/exploding when dealing with long-term dependencies, the S4 model uses the HiPPO matrix to initialize A, which achieves the gradient decay/increase with a polynomial rate rather than exponentially. HiPPO tries to compress previous input signals into a vector of coefficients that can capture recent tokens well and decay old tokens. This architecture allows S4 models to retain almost all historical information through functional approximation with minor information loss. Here is an



-1	0	0	0
$-\sqrt{3}$	-2	0	0
$-\sqrt{5}$	$-\sqrt{15}$	-3	0
$-\sqrt{7}$	$-\sqrt{21}$	$-\sqrt{35}$	-4

Figure 2.1: The Example HiPPO Matrix, when $n = k = 4$.

example HiPPO matrix when $n = k = 4$ in Figure 2.1 and the equation of the HiPPO matrix:

$$A_{nk} = -\begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k \\ n+1 & \text{if } n = k \\ 0 & \text{if } n < k \end{cases} \tag{2.4}$$

Discretization Except for continuous inputs, the sequential modelling task is also confronted with discrete data, such as textual sequences. To address the challenge of processing discrete data, S4 adopts a strategy to discretize the continuous SSM, thereby facilitating the model's approximation of the underlying continuous domain. The discretized parameters matrices $\bar{A}, \bar{B}$ and $\bar{C}$ in discretized SSM (Equation 2.3) which can be computed by a bilinear approximation [56] with step size $\Delta \in \mathbb{R}$:

$$\bar{A} = (I - \Delta/2 \cdot A)^{-1}(I + \Delta/2 \cdot A), \quad \bar{B} = (I - \Delta/2 \cdot A)^{-1}\Delta B, \quad \bar{C} = C \tag{2.5}$$

It is important to note that this discrete SSM can only be used to process 1-dimensional data. Therefore, to handle high-dimensional data, these discrete SSMs need to be stacked, and the outputs of each SSM should be concatenated to obtain the final result. The S4 model uses the HIPPO matrix to preserve long-term dependencies effectively and combines convolutional and recurrent implementations for fast training and inference. However, the state space matrices in the S4 model are unable to adapt to various inputs, which limits the model's ability to execute input-dependent inference.

2.2.3 Selective State Space Models (S6)

To overcome the S4 model's limitation in performing input-dependent inference, Gu et al. [21] introduced the S6 model, which processes inputs selectively. Unlike its predecessor, the S4 model, which utilizes time-invariant parameter matrices (A, B, C,

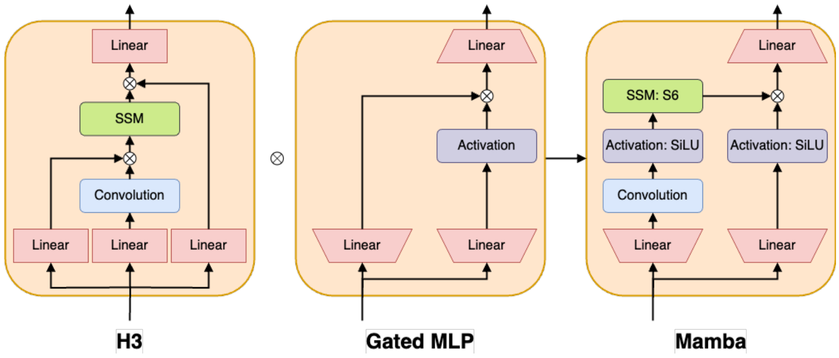


Figure 2.2: The Detailed Structure of Mamba [21], which Combines the H3 Model [14] and Gated MLP [33] to Integrate SSM and MLP into a Single Block.

and Δ), the S6 model obtains $B \in \mathbb{R}^{B\times L\times N}$, $C \in \mathbb{R}^{B\times L\times N}$, and $\Delta \in \mathbb{R}^{B\times L\times D}$ by applying a linear transformation to input sequence $x \in \mathbb{R}^{B\times L\times D}$, making these matrices data-dependent, and enabling the model to adapt its behaviour based on the input. Although A is not data-dependent, the input dependency matrix Δ enabled A to become data-dependent by the discretization process. Here is the new discretization approach in the S6 model, while C is still identical:

$$\bar{A} = \exp(\Delta A), \quad \bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B \tag{2.6}$$

Furthermore, to optimize traditional SSMs for efficient computation on modern GPUs, the S6 model integrates the Flash Attention [9] technology. This technique utilizes the hierarchical memory structure by computing the SSM states in Static Random Access Memory (SRAM), thereby minimizing the bottleneck caused by frequent read-write operations on the slower High Bandwidth Memory (HBM).

2.2.4 Mamba

To better encapsulate and leverage the efficient characteristics of the S6 layer, Gu et al. [21] proposed the Mamba architecture (shown in Figure 2.2), which combines the fundamental blocks of most SSM-based models, H3 model [14], with the gated MLP [33] commonly found in modern neural networks. Such architecture allows the Mamba block to integrate the SSM and MLP layers, enabling the Mamba-based model to be constructed by simply stacking the Mamba blocks homogeneously. Additionally, the Mamba block performs a convolution operation on the input before the SSM layer, which is adept at capturing local features, while the SSM is responsible for processing and capturing long-term dependencies in sequences. For the activation function, Mamba chose Sigmoid Linear Unit (SiLU) [13], which is nonlinear, continuously differentiable, and defined over the entire range from negative to positive infinity:

$$SiLU(x) = x \cdot Sigmoid(x), \quad Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{2.7}$$

SiLU not only addresses the vanishing gradient problem but also solves the issue with the Rectified Linear Unit (ReLU) [19] function, which is not zero-centred and has zero gradients in the negative range. These architectural features enable the Mamba block to effectively leverage the efficiency and long-term dependency-capturing capability of the S6 layer, which is considered a strong challenger to the Transformer. This project will build an NMT system based on the Mamba block to study its performance on MT tasks.

2.3 Related Works

2.3.1 Neural Machine Translation

The mainstream competitive NMT models are based on Encoder-Decoder auto-regressive [20] architecture. Since for a given sequence and target sequence pair $(x_{1:s}, y_{1:t})$, the translation task can be modeled as $P(y_{1:t}|x_{1:s})$. By applying the chain rule, this conditional probability can be expanded as follows:

$$p(y_{1:t}|x_{1:s}) = \prod_{i=1}^{t} p(y_i|x_{1:s}, y_{<i}) \tag{2.8}$$

Where $p(y_i|x_{1:s}, y_{<i})$ is modelled using the NMT model. The encoder receives source sequences as input and then encodes them to extract dependencies where the decoder captures dependencies in both encoder outputs and previous target tokens. Additionally, the translation model is auto-regressive where each target token is generated based on the source sequence and the previous generated tokens are then used to generate the next token. These encoder-decoder auto-regressive NMT systems are primarily implemented based on RNN, CNN, and Transformer models:

RNN-based NMT In 2014, Cho et al. [6] implement the first end-to-end RNN-based MT model. While their approach mainly leveraged SMT architectures, they incorporated phrases learned through NMT to strengthen SMT features. Later that year, Sutskever et al. [53] introduced a completely end-to-end NMT model, employing two LSTMs as the encoder and decoder, and found that reversing the training samples improved translation quality. Furthermore, to address the alignment between source and target sequences more effectively, Bahdanau et al. [3] proposed the attention mechanism, which set the stage for the Transformer and quickly became a cornerstone of NMT even deep learning research. However, RNNs depend on the sequential processing of previous time steps to generate the current output, which hampers their parallelization and leads to reduced computational efficiency, especially when dealing with large-scale datasets.

CNN-based NMT Given the ability of Convolutional Neural Networks (CNNs) to perform parallel computations, they present a promising avenue for research in NMT. Gehring et al. [17] introduced the ConvS2S model, which utilizes CNNs as the encoder and decoder to build an end-to-end NMT system, achieving notable success. Besides, Kaiser et al. [29] showed that stacking multiple convolutional layers can effectively extend the context length for long sequence translation. While CNN-based NMT benefits from parallel processing and exhibits high efficiency, the computational demands grow with the sequence length.

Transformer-based NMT The proposing of the Transformer model [58] represented a shift away from recurrent architectures by leveraging self-attention mechanisms. This approach greatly improved parallel processing capabilities and the management of long-range dependencies. Since its introduction, it has become a fundamental component in NMT systems and continues to dominate today. Following this, mBART [34] introduced the pretrain-finetune paradigm to the NMT field by pretraining a Transformer model on monolingual corpus and fine-tuning it with parallel text, greatly enhancing multilingual translation performance. Although Transformer-based models have revolutionized the NMT field, they still address dependencies between sequence elements in a relatively brute-force manner. Thus, they demand significant computational resources and encounter challenges with longer sequences.

Alternatively, some recent studies have adopted a language model perspective, employing encoder-only [16] and decoder-only [59] architectures that concatenate the source and target sequences as input, achieving results similar to those of encoder-decoder models while facing efficiency issues. The introduction and development of diffusion models [37, 5], which generate entire sequences simultaneously by denoising random noise in target sequences, have also provided an alternative solution to auto-regressive methods in the MT field and offered efficient inference speeds.

In summary, most competitive and mainstream NMT systems in recent years are based on the Transformer architecture, which requires substantial computational and storage resources. Moreover, no new paradigm has emerged to challenge the Transformer for many years. SSM has the capability to hold almost infinite historical information with

Workshop	Text Domain	Language Pair
WMT24	News	EN-DE,HE,ZH,JA,UK,RU,CS; CS-UK
	Biomedical	EN-FR,DE,IT,PT,RU,
	Literary	ZH-EN
IWSLT24	TED Talks	EN-DE,ZH,JA
	Physical Training	EN-JA
	Accent Challenge	EN- JA
WAT23	Scientific Paper	EN-JA
	Business Scene Dialogue	EN-JA
	Patent	JA-EN,ZH,KO
	IT domain and Wikinews	EN-HI,TH,MS,ID,VI

Table 2.1: Text Domain and Language Pair Supported by WMT24, IWSLT24, WAT23

limited storage space through functional approximation, making it a highly promising research direction.

2.3.2 Datasets

Bilingual parallel datasets are the most important data resources in NMT research. Since NMT models deployed between different language pairs require different parallel corpora for training, there is a significant demand for parallel corpora in NMT research. Currently, the publicly available datasets used in mainstream research are primarily provided by three workshops: The Workshop On Machine Translation (WMT) [32], The International Workshop on Spoken Language Translation (IWSLT) [50], and The Workshop on Asian Translation (WAT) [38]. WMT is the world's largest machine translation workshop, mainly targeting European languages, while WAT is focused on Asian languages. IWSLT also provides spoken language translation data for audio tasks. These datasets support a wide range of language pairs and domains and are continuously updated. Table 2.1 indicates the language pairs and corpus domains supported by recent workshops in 2023/24, and many more datasets are supported in previous workshops. In addition to the workshops mentioned above, OPUS [55] also provides a large number of parallel corpora for various language pairs and has released the OPUS-100 [61] dataset, which is an English-centric multilingual parallel corpus that covers over 100 languages.

Chapter 3

Methodology and Implementation

This project adopted an encoder-decoder auto-regressive model as the foundational architecture for the Mamba-based model. This choice is motivated by the aim of this project, which investigates the advantages and limitations of the Mamba block in comparison to the traditional attention layer regarding translation quality and performance. Thus, unlike alternative perspectives such as language modelling or diffusion models, this choice provides a more robust benchmark, baseline, and reference for this research.

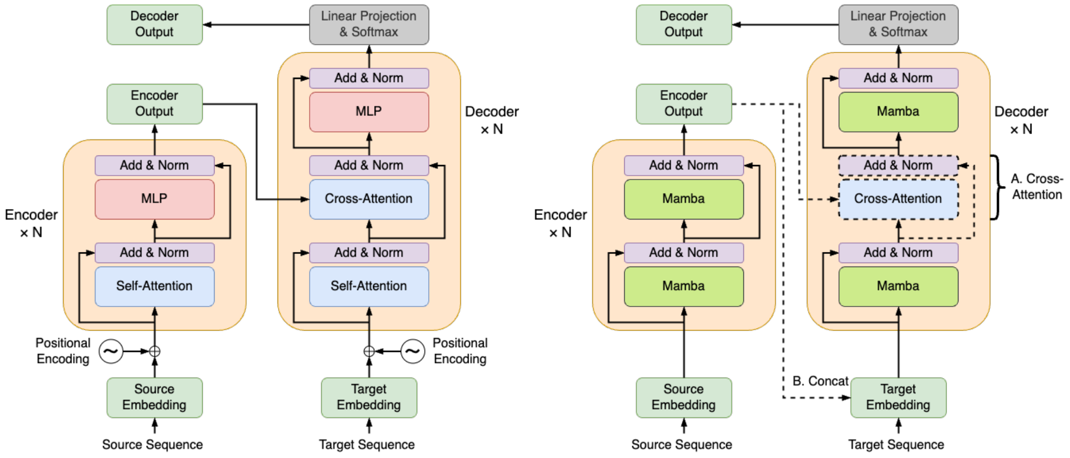
3.1 Model Architecture

3.1.1 Baseline

This project chose the Transformer Base model (called Transformer in the following chapter) [58] as the baseline since it employs both self-attention and cross-attention and remains the best primitive in NMT and even the NLP field. These attention mechanisms enable the Transformer to handle long-term dependencies well but suffer computational and memory bottlenecks when sequence length increases, making it a robust and reasonable benchmark for translation quality and model efficiency comparison with the

	Layer Count (Encoder + Decoder)	Encoder Parameters (Millions)	Decoder Parameters (Millions)	Embedding Parameters (Millions)	Total Parameters (Millions)
Transformer Base	6+ 6	3.15	4.20	22.4	66.5
Mamba Base	12 + 12	1.84	1.84	22.4	66.6

Table 3.1: Parameter Details and Layer Count for Transformer and Mamba Base Models



(a) Transformer Base Model (b) Mamba Base and MA Model

Figure 3.1: Model Architecture for Transformer [58] (a) and Mamba-based Model (b). ”Add & Norm” means residual connection and layer normalization. In Figure (b), the Mamba Base Model concatenates (Option B) the encoder output and target input, while the MA Model employs cross-attention (Option A).

Mamba-based model in this project.

Following the original Transformer Base settings (shown in Table 3.1), both the encoder and decoder stack $N = 6$ identical layers and set the number of heads to 8 for multi-head attention (MHA). The details of the Transformer are shown in Figure 3.1(a). The encoder layer consists of two sub-layers: self-attention and MLP, while the decoder layer is composed of three sub-layers: self-attention, cross-attention, and MLP. A residual connection and layer normalization follows each sub-layer.

3.1.2 Base Architecture

The Mamba Base NMT model (Figure 3.1(b)) also follows the encoder-decoder architecture by stacking the Mamba Block. Similarly, residual connection and layer normalization are applied after each Mamba layer. The parameter and layer details of the Mamba Base model are shown in Table 3.1. To match the model size with the Transformer, both the encoder and decoder stack $N = 12$ identical Mamba layers with an expansion factor $E = 2$, which can expand the model hidden dimension $D = 512$. For state space dimension d_{state}, this project uses a reasonable choice $d_{\text{state}} = 64$ provided by Gu et al. [21], which balances the performance and computing speed.

The Mamba block’s parameters consist of linear projections and SSM parameters (A, B, C, and Δ), with the majority of parameters concentrated in the linear projections, totalling $3ED^2 = 6D^2$. In contrast, one self-attention and one MLP layer in the Transformer have a total of $12D^2$ parameters. Thus, the encoder uses $N = 12$ Mamba blocks, which is twice the Transformer, to match the parameter count.

The decoder concatenates the encoder output with the target sequence as input, which results in the decoder processing approximately twice the length of sequences as the encoder. The reason for using this less efficient method is that Mamba blocks cannot aggregate features from two different sequences like a cross-attention layer. Additionally, Mamba's selective scan is completed through a single scan operation, making it difficult for the model to initialize or embed the encoder's final state as RNN-based models do. Therefore, this approach of utilizing the encoder output might undermine the efficiency of the Mamba Base model, including its memory usage and execution speed.

3.1.2.1 Word Embedding

Similar to most translation models, this project employed learnable word embeddings as the hidden dimensions of the tokens, setting $D = d_{\text{model}} = 512$. These embeddings capture semantic information about the words and facilitate better representations in the model. The model also employed a fully connected layer to project the output of the final Mamba layer from d_{model} dimension to the vocabulary size dimension, generating the logits for the tokens used to predict the next token. Additionally, to improve the model's parameter efficiency, all models in this project share embeddings across the encoder, decoder, and the final fully connected layer weights.

For the Transformer model and subsequent ablation experiments with mamba-based models, position embeddings are used alongside word embeddings to convey the order of tokens in a sequence explicitly. Additionally, to maintain numerical stability [58] between word and position embeddings, models employing position embeddings must multiply the weights of the word embeddings by embedding scale coefficient $\sqrt{d_{\text{model}}}$.

3.1.2.2 Layer Normalization

To accelerate the training speed of the model, this project utilizes RMS Norm (Root Mean Square Normalization) [60] for normalization instead of the basic Layer Norm (Layer Normalization) [2]. RMS Norm is an improved method based on the Layer Norm, which only requires the calculation of the root mean square instead of the mean and standard deviation for the Layer Norm. This simplification has been shown both theoretically and experimentally to save between 7% to 64% of computation resources while achieving similar performance [60]. Here is the equation of the RMS Norm:

$$\mathrm{RMSNorm}(x) = \frac{x}{\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} \cdot \gamma + \beta \tag{3.1}$$

Where D is the embedding dimension, γ and β are the learnable parameters, and the denominator is the RMS of samples.

3.1.3 Mamba with Cross-Attention

In base Mamba NMT models, this project concatenated the encoder output and the target sequence to serve as input to the decoder, resulting in the decoder needing to handle sequences of double the length. This may pose challenges for the model in capturing long-distance dependencies, and subsequent experimental results confirmed this hypothesis. Therefore, this project also implements a Mamba with Attention Model (MA, shown in Figure 3.1(b)), which employs a cross-attention layer, with the number of heads set to 8, between two Mamba layers in the decoder. Similarly, the cross-attention layer also includes a residual connection and RMS normalization. These changes allow the MA model to use the target sequence directly as the decoder input without concatenation, reducing the sequence length and improving the model's efficiency. Besides, the MA model features an architecture that is more similar to Transformer, allowing for a deeper analysis of the strengths and weaknesses of the Mamba model and its characteristics.

3.2 Evaluation Methods

3.2.1 Quality of Translation

This project intends to use Bilingual Evaluation Understudy (BLEU) [43], Character F-score (ChrF) [46] and Crosslingual Optimized Metric for Evaluation of Translation (COMET) [49] as automatic metrics to evaluate the token overlapping, character overlapping and semantic similarity between the hypothesis and reference. Additionally, this project conducts human analysis on specific examples from the test set based on word-level and sentence-level linguistic phenomena. This comprehensive approach aims to achieve a more rigorous and reasonable assessment of translation quality.

BLEU BLEU is the mainstream MT evaluation metric assessing how closely a translation matches reference translations by computing the geometric average of precision scores for n-grams overlapping between the model's translations and references. The BLEU score equation is given, with higher scores indicating better translation quality:

$$\text{BLEU}_N = BP \times \exp(\sum_{n=1}^{N} W_n \log(P_n)), \tag{3.2}$$

$$P_n = \frac{\sum_{c \in \text{hypotheses}} \sum_{\text{n-gram} \in c} \text{Count}_{\text{references}}(\text{n-gram})}{\sum_{c' \in \text{hypotheses}} \sum_{\text{n-gram}' \in c'} \text{Count}_{\text{hypotheses}}(\text{n-gram}')} \tag{3.3}$$

Where W is the weight of n-grams, N is the maximum length of n-grams, and BP is the Brevity Penalty coefficient to discourage shorter translations. This project set $N = 4$ and $W = \frac{1}{N}$, which is the convention for most MT research. For Brevity Penalty (BP):

$$BP = \begin{cases} 1 & \text{if } l_c > l_r \\ exp(1 - \frac{l_r}{l_c}) & \text{if } l_c \leq l_r \end{cases} \tag{3.4}$$

Where l_r is the length of reference, and l_c is the length of the hypothesis. The primary advantage of BLEU is its straightforwardness and computational efficiency, which has led to its wide use in MT tasks. However, BLEU assumes that input sentences have been tokenized, and variations in tokenization methods can produce result biases. To address this issue, this project uses sacreBLEU [47] to calculate the BLEU score, which employs standardized tokenization after removing BPE and detokenization.

ChrF ChrF is a character-level evaluation metric in MT, which is similar to the BLEU score. It computes precision and recall based on character n-grams, making it particularly effective for languages with rich morphology, such as German, in this project. By focusing on character matches rather than words, ChrF captures subtleties that may be overlooked by word-level metrics, offering a more nuanced evaluation of translation accuracy. The ChrF score is derived from the harmonic mean of precision and recall, and it is robust to morphological variations, allowing it to assess translations in languages where word forms change significantly. As a result, ChrF has gained popularity in MT research as a complementary metric to traditional evaluations like BLEU, providing a comprehensive perspective on translation performance.

COMET Although BLEU remains the most mainstream evaluation metric in current MT research, it only considers the formal similarity of the translation without taking into account semantics, same as ChrF. Therefore, this project also introduces COMET to provide a more scientific analysis of translation quality. COMET is a deep learning-based metric that achieved SOTA results correlating with human judgements in WMT19, and WMT20's Metrics shared tasks. The architecture of COMET is shown in Figure

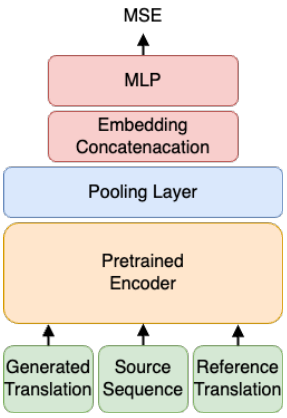


Figure 3.2: The Structure of COMET [49] . It inputs generated translation, source sequence and reference translation and trains by minimizing the Mean Square Error (MSE).

3.2. COMET takes hypothesis, source, and reference sentences as inputs and leverages a pretrain cross-lingual language model as an encoder to extract features from the input texts. Then, it employs a scoring model trained on human evaluation datasets to produce the final score, with higher scores indicating better translation quality.

Linguistic Phenomena Besides these automatic evaluation metrics, this project will also analyse specific translation sentences to evaluate the ability of models to handle different linguistic phenomena. These linguistic phenomena are primarily categorized at the word-level and sentence-level. The word-level includes unseen words, synonyms, proper nouns, and morphology, while the sentence-level encompasses interrogative sentences, passive voice, constituent structure, and some complex sentence structures. Analyzing these linguistic phenomena can lead to a deeper understanding of the model's capabilities and limitations, providing more interpretability.

3.2.2 Efficiency and Resource Usage

In addition to translation quality, model efficiency and resource utilization are crucial model performance indicators. Therefore, under the premise of having nearly equal model parameters, this project uses Words Per Second (WPS), FLOPs, and GPU memory usage to evaluate the efficiency and resource utilization of the models.

WPS WPS represents the number of words processed per second by the model and is an important metric of the processing speed of the model. Higher WPS means that the model can process more tokens within a fixed interval, which is suitable for evaluating the model's single step inference performance and training efficiency. Therefore, this project will record and compare the WPS of different models in training and validating to evaluate the efficiency of models.

Inference Speed Inference speed refers to the efficiency of a model in generating outputs, typically measured by the number of tokens produced per second. In this project, the model's inference speed is evaluated with input sequences of varying lengths, specifically 1, 10, 100, and 1000 tokens, using 10 synthetic sentences for each group. The total time taken for the incremental inference steps is recorded, and the inference speed is calculated by dividing the total inference time by the total number of tokens. This method provides a quantitative assessment of the model's efficiency in generating outputs, allowing for a comparison between different models.

GPU Memory Usage GPU memory usage indicates the model's memory usage during execution. Lower GPU memory usage means the model uses hardware resources more efficiently, helping to run larger models or handle larger batches of input sequences in limited hardware environments. This project will record the average GPU memory usage during training and peak GPU memory usage during incremental inference to evaluate the model's memory utilization efficiency.

These three metrics reflect the model's execution speed and efficiency in using computational and memory resources. By analyzing these results, a more comprehensive understanding of the model's efficiency and resource utilization can be obtained.

3.2.3 Attention Distribution

Aside from evaluating the model's performance, the main objective of this project is to explore the potential of the Mamba model as a hidden-attention model to replace traditional attention mechanisms. Therefore, this project will also use attention heatmaps to investigate Mamba's ability to capture the correlation between the source and target sequences. Since Mamba is a hidden-attention model, it cannot directly obtain the attention matrix as an intermediate variable like the Transformer. Therefore, this project adopts the method proposed by He et al. [26] to calculate the word importance using integrated gradients. To obtain the attention of source tokens on the generated target tokens, this project masks each source token individually and replaces it with a padding token. Subsequently, it computes the relative change in the activation of the decoder's final layer caused by this substitution, employing L2 distance as the measure. Applying this method across all source tokens derives a two-dimensional matrix that reflects the impact of each source token on each corresponding target token.

Chapter 4

Training Details

4.1 Dataset, Preprocess and Batching

This project trained Transformer and all Mamba-based models on WMT 2014 English-German dataset [4], which is a shared task of the annual Workshop on WMT Conference and serves as benchmarks for NMT systems training. This dataset specifically comprises parallel texts in English and German, including 4.51M rows for training, 3K for validation, and 3K for testing. It features diverse content, such as parliamentary records and news articles. Besides, the WMT14 EN-DE datasets have high consistency and quality. They are meticulously curated to encompass a wide range of genres, complexities, and linguistic features, making them well-suited for benchmarking the performance of NMT systems. Additionally, German is a morphologically rich language, which provides a valuable target for evaluating translation models. Its complex inflectional vocabulary challenges models to accurately capture and translate nuanced meanings. Furthermore, previous entries in the WMT Workshops offer a rich set of references and baseline, enabling this project to gauge expected performance standards and evaluate the effects of Mamba's innovations relative to established benchmarks.

For preprocessing, sentences were tokenized by Byte-pair encoding (BPE) [51] with 40000 BPE tokens and shared source-target vocabulary. BPE is a subword-level tokenization technique widely used in NLP, especially in machine translation. BPE works by iteratively merging the most frequent pairs of characters or character sequences in the text to form subword units. This method effectively handles out-of-vocabulary words and rare word issues by breaking them into more common subwords. BPE allows for a controlled vocabulary size, making it memory-efficient and adaptable to multiple languages. It combines the advantages of word-level and character-level tokenization methods, improving translation quality and model robustness while enhancing processing efficiency.

To avoid memory overflow and low computational efficiency caused by varying input sequence lengths during training, this project controls the input amount per batch using maximum tokens instead of batch size. This approach allows for flexible handling of input sequences of different lengths, ensuring that each batch's total number of tokens does not exceed GPU memory limits. Each training batch contained sentence pairs containing approximately 12K tokens (including source and target sequence).

4.2 Hardware and Software Configuration

This project has chosen to train the model simultaneously on the Eddie [40] and Cirrus [41] GPU clusters to expedite the completion of the model training due to the heavy training tasks. Each training job utilizes two NVIDIA A100 80GB PCIe GPUs on the Eddie cluster, while four NVIDIA Tesla V100-SXM2-16GB GPUs are employed for training on the Cirrus cluster. The stopping criteria for training are either 100 epochs or a validation loss that does not improve for 50K steps.

This project is based on Python, implementing the model within the PyTorch [44] framework and using the Fairseq [42] framework for data preprocessing, training, and translation generation. For GPU acceleration, the training script relied on CUDA [39]. It also utilized Apex [7] for mixed precision training and Causal-Conv1d [8] for efficient causal convolutions in autoregressive models optimizing training and inference efficiency. TensorboardX [28] was employed to log and monitor intermediate variables during training and inference to facilitate subsequent analysis, while SacreBLEU [47] and COMET [49] were used to evaluate the quality of translations. Here is the list of the framework, package and language used in this project.

- Python: 3.9.19
- Fairseq: 0.12.2
- PyTorch: 2.3.1
- CUDA (Eddie): 12.1.105
- CUDA (Cirrus): 11.8.89
- APEX: 24.4.1
- Causal_Conv1D: 1.4.0
- TensorboardX: 2.6.2
- SacreBLEU: 2.4.2
- COMET: 2.2.1

4.3 Optimizer and Learning Rate

This project used the Adam [31] optimizer, consistent with the original Transformer, to maintain the same training recipe. Compared to the Stochastic Gradient Descent (SGD) optimizer, Adam dynamically adjusts the learning rate for each parameter by computing the first and second momentums of the gradients. This allows Adam to maintain a higher learning rate in certain directions, leading to faster convergence.

The Adam optimizer has three main hyperparameters: β_1, β_2 and ε. β_1 and β_2 control the exponential decay rates for the moving averages of the first and second momentums of the gradients, respectively, where ε is a small constant added to the denominator to ensure numerical stability. This project set $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 10^{-9}$ and warm up step is 4000 which is also same as original Transformer recipe.

For the initial learning rate, this project trained the model with different learning rates and ultimately chose an initial learning rate of 5×10^{-4} that achieved the lowest validation loss. It is worth noting that the final performance obtained with a learning rate around $\times 10^{-4}$ was quite similar. This may be because the Adam optimizer can adaptively adjust the learning rate based on historical information and uses exponential decay moving averages to smooth the changes in gradients, making it less sensitive to the choice of the initial learning rate.

4.4 Regularization

Due to the significant architectural differences between the Transformer and Mamba models, this project employs different regularization strategies for each model. This project employs three types of regularization methods which are:

- **Residual Dropout** [52]: During training, Dropout randomly sets a portion of outputs to zero, forcing the model to rely on different features in each iteration.
- **Weight Decay** [27]: Adding the L2 norm of the weight parameters to the loss function to encourage weights to stay small and avoid.
- **Label Smoothing** [54]: In the final projection, instead of using one-hot encoded targets, a small portion of the probability mass is distributed to other class labels to prevent the model from becoming overly confident on the training set instances.

For Transformer models, this project followed the official setting in the original paper. Dropout was applied after the output of each block with residual connection and after

Dropout	Weight Decay	Valid Loss
0.1	0.0	4.224
0.0	0.1	4.181
0.0	0.2	4.167
0.1	**0.2**	**4.141**
0.2	0.2	4.156

Table 4.1: Hyperparameter Tuning Results of Dropout and Weight Decay on Validation Loss

compositing the word embeddings with positional embeddings with a dropout probability of 0.1. For label smoothing, this project set the smoothing factor to 0.1, meaning that 10% of the probability mass is distributed to non-target classes. Weight decay is not applied to the Transformer model.

For the Mamba-based model, when trained under the same regime as the Transformer, this project observed that the Mamba model exhibited faster convergence but suffered from severe overfitting. This necessitates tuning the hyperparameters for regularization. Due to computational constraints, performing a grid search for the parameters is impractical. This project opts to tune these two parameters separately. Since the models in the Mamba paper only applied weight decay, this project first selects the optimal weight decay coefficient without applying dropout. Then, under this setting, different dropout values will be adjusted. Since label smoothing directly affects the output distribution rather than the parameter weights, the setting for label smoothing remains consistent with that of the Transformer. Table 4.1 shows the result of different hyperparameter combinations. Based on the experimental results, this project obtained the following regularization configuration:

- **Transformer**
- Dropout: 0.1
- Weight Decay: 0.0
- Label Smoothing: 0.1

- **Mamba**
- Dropout: 0.1
- Weight Decay: 0.2
- Label Smoothing: 0.1

Chapter 5

Result and Analysis

This project implements the Mamba Base Model and Mamba with Attention Model (MA), training both on the WMT14 EN-DE dataset alongside the baseline Transformer. All models maintain a parameter count between 66M and 72M. This section evaluates the performance and efficiency of the models, revealing that the Mamba Base offers faster inference than the Transformer, while the Transformer achieves superior translation quality. The MA model attains translation quality similar to the Transformer without compromising the efficiency of the Mamba Base. Additionally, this project conducted ablation experiments to study the impact of components on the performance of the MA model, which indicates that Mamba can effectively replace both self-attention and the MLP block. Furthermore, this section analyzes specific translation sentences and visualizes the attention distribution. The analysis shows that the MA model handles word-level and sentence-level linguistic phenomena effectively. It even outperforms the Transformer in dealing with unseen words and ambiguous constituents while encountering difficulties with complex long-term dependencies. The attention visualization shows the MA model has a clearer and more uniform attention distribution, which indicates better fine-grained feature capture ability than the Transformer.

5.1 Mamba Base Model

5.1.1 Translation Quality

The Mamba Base model converged after 3 days of training on 4 NVIDIA V100 GPUs, while the Transformer model took 4 days to stop at 100 epochs. The project used the model obtained by averaging the last 10 checkpoints and applied beam search to

	BLEU	ChrF	COMET
Transformer Base	**25.82 ± 0.63**	**56.53 ± 0.44**	**0.82 ± 0.11**
Mamba Base	21.21 ± 0.72	49.80 ± 0.79	0.74 ± 0.15

Table 5.1: Translation Quality with a Standard Deviation Evaluated on WMT14 EN-DE Test Set with BLEU, ChrF and COMET between Transformers and Mamba Base Models.

generate translation with a beam size of 4 and a length penalty of 0.6, consistent with the original settings of the Transformer. The translations are evaluated using three metrics: BLEU, ChrF, and COMET, which respectively assess word-level similarity, character-level similarity, and semantic alignment. The results are presented in Table 5.1, which demonstrates Mamba Base perform worse than Transformer on these metrics. The Mamba Base model achieved a BLEU score of 21.21, much lower than the Transformer's 25.82, indicating its shortcomings in generating word combinations that align with reference translations. For the ChrF, the Mamba Base scored 49.80, lower than the Transformer's 56.53. It suggests that the Mamba Base model possesses a certain capability to handle morphology-rich language, such as German, but there is still a significant gap compared to the Transformer. Additionally, the Mamba Base achieved a COMET score of 0.74, reflecting a semantic gap compared to the Transformer.

These results indicate that the Mamba base model is inferior to the Transformer in overall translation quality, and it still falls short of the Transformer in capturing global, local, and hidden dependencies and patterns. One potential reason is that the implementation of Mamba makes it challenging to fit into the encoder-decoder framework efficiently. To improve computational efficiency, Mamba employs a hardware-aware algorithm that utilizes scanning to compute the SSM layer in parallel. This results in difficulties in obtaining the final SSM state of the encoder, and the implementation of Mamba also does not allow the initialization of the SSM state. Consequently, the model can only concatenate the encoder output and target sequence as decoder input. Consequently, compared to the Transformer, the Mamba decoder needs to handle longer sequences, which introduces more noisy dependencies and increases the difficulty of capturing dependencies between sequences, ultimately leading to worse translation quality.

Sequence Length Scaling To verify the hypothesis that the Mamba Base model has difficulty capturing long-distance dependencies due to the longer sequence length of the decoder input, this project evaluated the models' translation quality at different sequence lengths. The test set (3K rows) of the WMT14 EN-DE dataset was divided equally into

	Short (2-19)	Medium (20-30)	Long (31-80)	Overall
Transformer Base	**25.50 ± 1.53**	**25.31 ± 1.01**	**26.22 ± 0.93**	**25.82 ± 0.63**
Mamba Base	25.00 ± 1.44	24.03 ± 1.05	18.30 ± 1.14	21.21 ± 0.72

Table 5.2: SacreBLEU Score with a Standard Deviation of Mamba Base Models and Transformers on the WMT14 EN-DE Test Set for Different Source Sentence Lengths.

three subsets, short, medium, and long, based on the source sequence length, with each subset containing 1K sentences. Specifically, the short subset consists of sentences with less than 20 words, the medium subset contains sentences with 20 to 30 words, and the long subset includes sentences with more than 30 words (the maximum length is 80). Since the scores of BLEU, ChrF, and COMET are aligned across both the entire test set and all subsets, this project will focus on analyzing the BLEU score in subsequent research, as it is the most widely used metric. The results of the other metrics are presented in Appendix A. The BLEU scores of both the Mamba Base model and the baseline model were tested on each subset, and the results are presented in Table 5.2.

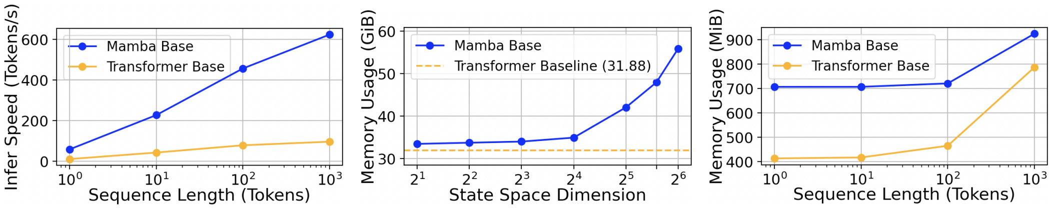
According to the result, it is evident that as the sequence length increases, the gap in translation quality between the Mamba Base and Transformer models also widens. In the short subset, the Mamba model achieved a BLEU score of 25.00, almost on par with the Transformer's score of 25.50. In the medium subset, the gap between the Mamba and Transformer is maintained at around 1.3. However, in the long sequence subset, the Transformer scored 26.22 while the Mamba Base only reached 18.30, resulting in a significant gap of nearly 8. This further demonstrates that the Mamba Base struggles with capturing long-term dependencies, indicating a potential need for a new architecture to fit into the encoder-decoder framework efficiently. In subsequent experiments, the Mamba with Attention model was evaluated, which employs cross-attention in the decoder to capture dependencies between the encoder output and decoder input, achieving similar translation quality and even performing better on handling specific linguistic features compared to the Transformer.

5.1.2 Model Efficiency

The main advantage of the Mamba model over the Transformer model is its ability to achieve constant complexity per step during autoregressive inference, and it applies

	WPS	WPS	GPU Memory Usage/GiB
	Train	Validate	Train
Transformer Base	$\mathbf{7.5 \times 10^4}$	$\mathbf{2.3 \times 10^5}$	**31.88**
Mamba Base	1.7×10^4	7.7×10^4	55.89

Table 5.3: Result of Training Efficiency for Mamba Base Models and Transformers. Including training WPS, validation WPS, and average GPU memory usage



(a) Inference Speed vs. Sequence Length

(b) Avg. Training GPU Memory Usage vs. State Space Dim.

(c) Peak Inference GPU Memory Usage vs. Sequence Length

Figure 5.1: Efficiency Metrics of the Mamba Base and Transformer Model: Inference Speed and Memory Usage with scaling sequence length and state space dimension

hardware-aware algorithms that reduce I/O overhead to O(d_{state}), resulting in speed improvements of $20-40$ times [21]. This project evaluates the model's efficiency during training and inference from both speed and memory perspectives. The model uses WPS and inference speed with varying input lengths as metrics to evaluate training and inference speed, respectively, while average and peak GPU memory usage is employed to assess memory efficiency during training and inference. Training is conducted on 4 NVIDIA V100 16GB GPUs, while inference is performed on an NVIDIA T4 16GB GPU. Experimental results (shown in Table5.3 and Figure 5.1) show that the Mamba model exhibits much faster inference speed during autoregressive tasks but slower training speed and higher memory usage. However, the training process indicates the Mamba Base model converges more quickly than the Transformer, resulting in lower total FLOPs overhead during training.

For the training speed of the models, Table 5.3 shows both the training and evaluation WPS of the Mamba Base Model are lower than those of the Transformer. This may be because of teacher forcing during training, where the model performs only one forward pass for the full sequence rather than generating tokens incrementally. The computational complexity of the linear projections in both models is similar, but the self-attention complexity of the Transformer is $O(BL^2D)$, while the complexity of the

SSM layer in Mamba is $O(BLd_{\text{state}}D)$. Given that the average sequence length in both the training and validation sets is 28, which is smaller than the setting of $d_{\text{state}} = 64$ in this project, and the value of L in the Mamba decoder is twice of the Transformer, these factors contribute to the slower training speed of the Mamba model compared to the Transformer. As the sequence length increases, this performance gap may narrow, which could warrant further investigation in future studies on document-level datasets.

For the inference speed of the models, Figure 5.1(a) indicates that the Mamba Base model achieves 5-7× faster than the Transformer, with an even larger gap as the input sequence length increases. This is primarily because Mamba, based on the SSM model, achieves constant complexity during auto-regressive inference, while the Transformer can only achieve linear complexity, even applying KV cache [45]. Additionally, Mamba utilizes hardware-aware algorithms to reduce I/O overhead while performing parallel computations of the SSM layer through scanning, which significantly accelerates the model's inference speed. Meanwhile, the reason both models experience increased inference speed as the sequence length grows could be that longer sequences result in lower average I/O overhead per token. Additionally, since Mamba handles I/O more efficiently, it shows a more significant speedup compared to the Transformer.

For GPU memory usage, the Mamba base model allocates more memory in both the training and inference processes. This may be because the latent state of the SSM layer in the Mamba model occupies more memory than the attention matrix in Transformer. To verify the hypothesis, this project records the average GPU memory usage by adjusting the dimension of the state space d_{state}, and the results are presented in Figure 5.1(b). It demonstrates that the memory overhead of the Mamba model decreases as the dimension of state d_{state} decreases, and at $d_{\text{state}} = 2^4 = 16$ ($d_{\text{state}} = 2^6 = 64$ for the Mamba Base model), it reaches a similar value compare to Transformer model, which aligns with the efficiency benchmark provided by Gu et al [21]. Although the latent space of the SSM requires more memory during training, its space complexity remains constant, while the attention matrix has a space complexity of $O(L^2)$ during training concerning sequence length L. When using the KV cache during inference, this can be optimized to $O(L)$. This indicates that the memory usage gap between the two models should gradually decrease as the input sequence length increases. The results of the inference memory usage experiments, shown in Figure 5.1(c), confirm this hypothesis: with sequence length increasing, the slope of the Transformer's curve is larger. This suggests that Mamba may be more competitive regarding memory usage when handling longer sequences. Consequently, further research could explore applying Mamba to

	Short (2-19)	Medium (20-30)	Long (31-80)	Overall
Transformer Base	**25.50 ± 1.53**	**25.31 ± 1.01**	**26.22 ± 0.93**	**25.82 ± 0.63**
Mamba Base	25.00 ± 1.44	24.03 ± 1.05	18.30 ± 1.14	21.21 ± 0.72
Mamba Attention	24.89 ± 1.51	25.02 ± 1.03	25.22 ± 0.96	25.05 ± 0.64

Table 5.4: SacreBLEU Score with a Standard Deviation of MA Models compared to Transformers and Mamba Base Models on the WMT14 EN-DE Test Set for Different Source Sentence Lengths.

document-level NMT to validate this assumption.

Overall, the Mamba Base model achieves an inference speed of 5-7 × faster than the Transformer and a faster converge speed. However, it also has a slower training speed and larger GPU memory overhead on the WMT14 EN-DE dataset while it is flexible to trade-off efficiency and performance by adjusting the state space dimension d_{state}. Nevertheless, the result also demonstrates Mamba's potential for processing longer sequences, which requires further research on document-level datasets.

5.2 Attention Enhanced model

After comparing and analyzing the performance of the Mamba Base model, it is evident that its inability to fit the encoder-decoder architecture efficiently leads to struggles in capturing long-distance dependencies. Therefore, this project proposes the Mamba with Attention Model (MA), which features a structure similar to that of the Transformer and achieves competitive translation quality compared to the Transformer model. The BLEU scores of the MA model and other previous models evaluated on different sequence length subsets are presented in Table 5.4.

The MA model achieved a score of 25.05 on the entire test set, significantly surpassing the Mamba Base model's score of 21.21, with a gap of less than 1 compared to the Transformer's score of 25.83. In the short subset, the MA model maintained the same strong performance as the Mamba Base model. In the medium and long subsets, the MA model reached scores similar to the Transformer, improving by 1 point and nearly 7 points compared to the Mamba Base model, respectively. Overall, the MA model achieved scores comparable to the Transformer across the entire test set and all subsets, narrowing the score gap to below 1 in each bucket. Compared to the Mamba Base model, the MA model employs cross-attention, allowing it to extract features without treating the source and target sequences as a single sequence, better capturing dependencies and improving performance. For efficiency, the MA model presents almost equivalent training memory usage (57.49 GiB), training WPS (1.8×10^4) and Inference speed, which suggests that the employment of cross-attention in the MA model on this dataset did not affect the model's efficiency.

The MA model is a reasonable solution that applies Mamba to the sentence-level MT domain, combining the efficiency of the Mamba model's hardware-aware algorithms with the cross-attention model's ability to handle dependencies between two sequences. However, for document-level translation tasks, the efficiency of the MA model may be constrained by the quadratic complexity of cross-attention concerning sequence length. This limitation may require further research to develop an implicit cross-attention block based on Mamba to address the issue.

5.3 Ablation

To investigate the impact of various components on the performance of the MA model, this project conducted a series of ablation experiments, ensuring that all ablated models maintained a parameter count between 66M and 72M. Specifically, three sets of experiments were conducted, where different components of the Transformer were used to replace the Mamba blocks:

1. Replacing the first and second Mamba layers in the encoder and decoder of the MA model with self-attention and MLP layers, respectively. Additionally, to maintain the model parameter count between 66M and 72M, this project also adjusted the stack size of the self-attention Mamba model to $N = 7$ and reduced the MLP's hidden dimension to 1536 ($\frac{3}{4}$ of the original dimension 2048).

2. Replacing the MA model's encoder and decoder with those of the Transformer, respectively. The employed Transformer component in this experiment includes position embeddings, whereas the Mamba does not incorporate them.

3. In the embedding, position encoding is added to explicitly input the order information between tokens into the model. This helps eliminate interference caused by different embeddings in the Transformer and Mamba models.

The experimental results in Table 5.5 indicate that replacing the Mamba block with either the self-attention or MLP layer leads to an overall BLEU score improvement

	Short	Medium	Long	Overall
	(2-19)	(20-30)	(31-80)	
Transformer Base	25.50 ± 1.53	**25.31 ± 1.01**	**26.22 ± 0.93**	**25.82 ± 0.63**
Mamba Base	25.00 ± 1.44	24.03 ± 1.05	18.30 ± 1.14	21.21 ± 0.72
Mamba Attention	24.89 ± 1.51	25.02 ± 1.03	25.22 ± 0.96	25.05 ± 0.64
Mamba-MLP	**25.54 ± 1.49**	24.79 ± 1.04	25.71 ± 0.96	25.38 ± 0.66
Self-Attention-Mamba	25.30 ± 1.48	24.24 ± 1.08	25.87 ± 0.91	25.26 ± 0.62
Mamba$_{EN}$ − Transformer$_{DE}$	24.32 ± 1.45	24.57 ± 1.06	25.28 ± 0.91	24.92 ± 0.65
Transformer$_{EN}$ − Mamba$_{DE}$	24.61 ± 1.50	24.30 ± 1.04	25.35 ± 0.91	24.92 ± 0.63
Mamba Attention Positional	24.61 ± 1.49	24.28 ± 1.09	25.37 ± 0.93	24.89 ± 0.65

Table 5.5: SacreBLEU Score with a Standard Deviation for Ablation Experiments on the WMT14 EN-DE Test Set, for Different Source Sentence Lengths.

of 0.2 to 0.4, and the model utilizing self-attention achieves a BLEU score of 25.54 on the short subset, slightly exceeding Transformer's score of 25.50. However, all models in the ablation experiments show that their BLEU score across all subsets and the full test set fall within one standard deviation of the MA model's results. This suggests that using Transformer components to replace the Mamba block does not yield a significant improvement or impact on translation quality, as all models achieve performance competitive to the Transformer. Therefore, the Mamba layer can serve as an efficient linear-complexity hidden-attention model to replace the self-attention layer and MLP layer in the Transformer, achieving similar translation quality, faster inference speed, and better scalability in sequence length.

5.4 Linguistic Phenomenon Analysis

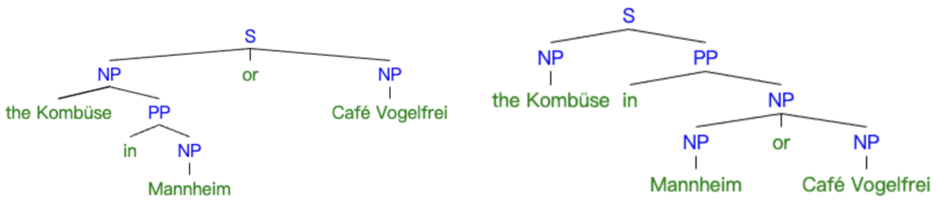
To gain a deeper understanding of the characteristics of the MA model, this project also analyzes the specific translations with different word-level and sentence-level linguistic phenomena and compares with the Transformer's translations.[1] This project first identified the key linguistic phenomena and then selected sentences that specifically exhibit these features to systematically test the model's performance on various linguistic phenomena. The example sentences are shown in Table 5.6 and Appendix B, where S means source sentence, T means target reference, MA is MA's translation,

[1] The reference resource of the analyses in this section is the Cambridge EN-DE Dictionary: https://dictionary.cambridge.org/dictionary/german-english/.

	Phenomenon	Example
Word level	Unseen Word	S: Beautiful animals and delicious **tarts entice** T: Schöne Tiere und leckere **Torten locken** MA: Schöne Tiere und leckere **Torten locken** zum ersten Mal TR: Schöne Tiere und **köstliche Tintenfische**
	Synonym/ Semantics	S: Dog-lovers **victorious** T: Hundefreunde **erfolgreich** MA: Hundeliebhaber **siegreich** TR: Die Hunde-Liebhaber **siegreich**
	Proper Noum/ Acronym	S: **RBS** suspends two forex traders T: **RBS** suspendiert zwei Devisenhändler MA: **RBS** suspendiert zwei Devisenhändler TR: **RBS** suspendiert zwei Devisenhändler
	Morphemes	S: We **see** customers from all walks of life. Witnesses **saw** two people sitting in the car. T: Zu uns kommen Kunden aus jeder sozialen Schicht. Zeugen **sahen** zwei Menschen in dem Auto sitzen. MA: Wir **sehen** Kunden aus allen Lebensbereichen. Zeugen **sahen** zwei Personen im Auto sitzen. TR: Wir **sehen** Kunden aus allen Bereichen des Lebens. Zeugen **sahen** zwei Leute im Auto sitzen.
Sentence Level: Pros	Constituent Structure	S:There are vegan restaurants opening up, such as **the Kombüse in Mannheim or Café Vogelfrei**. T:Vegane Restaurants entwickeln sich, wie zum Beispiel **die Kombüse in Mannheim oder das Café Vogelfrei**. MA: Es eröffnen sich veganische Restaurants wie **die Kombüse in Mannheim oder das Café Vogelfrei**. TR: **In Mannheim und im Café Vogelfre**i öffnen sich veganische Restaurants wie **die Kombüse**.
	Interrogative Sentence	S: **How did** the universe come about and what does it consist of? T: **Wie ist** das Universum entstanden und woraus besteht es? MA: **Wie ist** das Universum entstanden und worin besteht es? TR: **Wie entstand** das Universum und worin besteht es?
	Passive Voice	S: One hundred people **were brought** out of the building to safety. T: Hunderte Menschen **wurden** aus dem Gebäude in Sicherheit **gebracht**. MA: Einhundert Menschen **wurden** aus dem Gebäude in Sicherheit **gebracht**. TR: Hundert Menschen **wurden** aus dem Gebäude in Sicherheit **gebracht**.
Sentence Level: Cons	Adverbial Clause	S: Before Friday 's **Bundesliga match** against **VfB Stuttgart**, the **'Ultras'** responded with silence - initially. T: Vor dem Freitagsspiel der **Fußball-Bundesliga** gegen den **VfB Stuttgart** reagierten die **Ultras** mit einem Schweigen - zunächst. MA: Der **VfB Stuttgart** hat vor Freitag mit Schweigen auf die **"Ultras"** geantwortet. TR: Vor dem **Bundesligaspiel** am Freitag gegen den **VfB Stuttgart** reagierten die **Ultras** mit Schweigen - zunächst.
	Emphatic Inversion	S: Only when the psychological strain becomes severe **do people** give it consideration. T: Erst wenn der Leidensdruck wirklich groß ist, **mache man** sich Gedanken. MA: Erst wenn die psychologische Belastung stark wird , **werden die Menschen** berücksichtigt. TR: Erst wenn die psychologische Belastung stark wird, **wird sie** berücksichtigt.

Table 5.6: Translation Example for MA and Transformer with Different Word-level and Sentence-level Linguistic Phenomenon. Green means good examples while Red is bad.

and TR means Transformer's translation. The analysis reveals that the MA model effectively handles word-level features such as synonyms, morphemes, and acronyms, demonstrating better performance than the Transformer in managing unseen words. Furthermore, the MA model excels at processing simple statements, common interrogative structures, alignment, and passive voice while accurately identifying ambiguous constituent structures that often confuse the Transformer. However, when handling sentences with inverted structure, such as emphatic inversions and adverbial clauses, the MA model struggles to convey adequate semantics and capture complex dependencies. For word-level linguistic phenomenon, the MA model exhibits superior handling of unseen words compared to the Transformer. For example, in the given sentences in Table 5.6, the bolded words "tarts" and "entice" did not appear in the training set, but the MA model successfully provided the same translations as the reference: "Torten" and "locken". In contrast, the Transformer model translated these words as "köstliche



(a) Proper Syntax Tree (also for MA model) (b) Syntax Tree for Transformer’s Translation

Figure 5.2: Syntax Trees for the German Constituent ”the Kombüse in Mannheim or Café Vogelfrei” for MA and Transformer Models

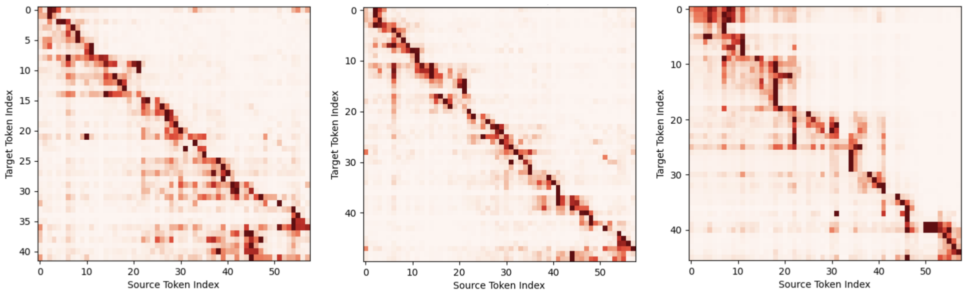
Tintenfische” (delicious squid), which completely deviates from the original meaning of the sentence. Additionally, the MA model demonstrates the ability to select the most contextually appropriate tokens among synonyms. For instance, in the synonym example, MA chooses ”siegreich” over ”erfolgreich” (both of which appear in the training set) as the translation for ”victorious” since the choice of reference ”erfolgreich” would lean more toward the meaning of ”successful”. Furthermore, MA effectively retains proper nouns from the source sequence and manages the complex morphological inflections of German when handling changes in person and tense. For example, the MA model preserves the acronym ”RBS” from the example sentence while correctly using ”sehen” and ”sahen” to correspond to ”see” and ”saw”, respectively, thus achieving alignment in both person and tense. Therefore, the MA model performs excellently in handling these word-level linguistic phenomena during testing, indicating that it has a sufficient understanding of the source sentence’s semantics. It effectively manages local details and some long-distance dependencies while demonstrating better generalization capabilities compared to the Transformer model.

For sentence-level linguistic phenomena, the MA model can effectively parse and disambiguate the syntactic structure of sentences while Transformer struggling. In the given example on the ”Constituent Structure” row, the constituent ”the Kombüse in Mannheim or Café Vogelfrei” has two possible parsing results (shown in Figure 5.2). The first syntax tree treats it as two noun phrases (NPs) connected by the conjunction ”or”, while the second interprets it as a single NP with a prepositional phrase (PP). In this context, the constituent should be parsed as two coordinated NPs since both ”Kombüse” and ”Café Vogelfrei” refer to different restaurants. The Transformer incorrectly parses it as the latter while the MA model successfully captures the dependencies between the tokens and performs disambiguation correctly, which indicates the MA model has stronger pattern recognition capability. Besides, the MA model can also handle common sentence types such as interrogative sentences and sentences with passive voice (Shown in Table 5.6, where “How did” corresponds to “Wie ist,” and “were brought” corresponds to “wurden gebracht”). However, it struggles when dealing with inverted structured sentences, such as emphatic inversion and adverbial clauses. These types of sentences often place the adverbial before the subject of the main clause, which can lead to interference in the model’s selection of the action subject and object in the main clause due to the subject and object present in the adverbial. For instance, when handling the example with the adverbial clause, the MA model mistakenly interprets the object “VfB Stuttgart” in the adverbial clause as the subject in the main clause, overlooking the actual subject “Ultras.” In the case of the emphatic inversion, the MA model misinterprets “people,” the subject of the action “consider,” as the object being considered. The Transformer model correctly handles conditional adverbials but encounters the same issues as the MA model when dealing with inverted emphasis sentences. The aforementioned sentence-level analysis indicates that the MA model effectively eliminates syntactic ambiguities within constituents but struggles to accurately identify the action subject and object within complex sentence structures. This suggests that while the MA model captures fine-grained token patterns better than the Transformer, it slightly lags in managing complex long-term dependencies.

Overall, the MA model demonstrates strong performance in handling word-level and most sentence-level linguistic phenomena, except that it struggles to detect long-term dependencies in complex sentences. Particularly, the MA model outperforms the Transformer in handling unseen words and constituent disambiguation. This indicates that the MA model has a robust ability to understand and capture fine-grained features and exhibits great generalization capability, which may be due to the convolution layer before SSM in the Mamba. However, it lags behind the Transformer in understanding and processing overall sentence structures. Therefore, further research may need to explore ways to reduce information loss when Mamba compresses historical information to enhance the model’s performance in complex sentence structures.

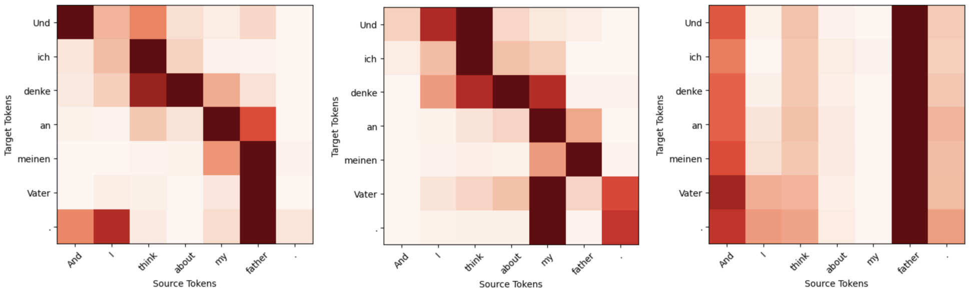
5.5 Attention Distribution

Recent studies [1, 10] have indicated that the Mamba model is an efficient implicit attention model. Therefore, to further explore its hidden-attention efficiency, this project has used attention heatmaps to visualize the model’s attention distribution, revealing its performance and potential advantages under different input conditions. This project



(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure 5.3: Attention Heatmap for a Long Sample (58 Tokens) Comparing Mamba Base, MA, and Transformer Models.



(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure 5.4: Attention Heatmap for a Short Sample (7 tokens, ”And I think about my father.”) Comparing Mamba Base, MA, and Transformer Models.

selected sentences from the short and long subsets and then plotted the corresponding attention heatmaps for the Mamba Base, MA, and Transformer models for analysis. The complete visualization results are presented in Appendix C. Figure 5.3 showcases a set of typical results for long sequences, where it is evident that the heatmap of the MA model is sharp, focusing on only a few specific words, while the Mamba Base model’s results are blurred. This means each target token of the Mamba Base Model attends to more source tokens, which indicates that the Mamba Base model is harder to correctly focus on specific parts of the source sequence when processing long sequences[57]. Additionally, compared to the MA model, the Transformer tends to exhibit more blurring at the start of the sequence, which may hurt the quality of translation. In handling short sequences, the Mamba Base Model overcomes the issue of attention blurriness, presenting a clear and uniform attention heatmap similar to the MA model. In contrast, while the Transformer’s heatmap is also clear, nearly all target words focus on only a few key information-rich words, such as ”father” in the example, rather than exhibiting the uniform attention distribution seen in MA-based models. This indicates that Mamba, as a hidden-attention model, can extract finer-grained features and demonstrate stronger representational capabilities when processing inter-sequence dependencies. However, this ability also makes it more susceptible to noise, potentially leading to overfitting. This conclusion further explains the phenomenon where the MA model outperforms the Transformer in disambiguation but faces challenges when dealing with inverted sentence structures.

5.6 Discussion

This section highlights that while the Mamba Base Model significantly outperforms the Transformer in inference speed and demonstrates better scalability in memory usage for varying sequence lengths, it struggles to effectively handle long sequence dependencies. This issue is tackled by incorporating a cross-attention block in the decoder, leading to translation quality comparable to the baseline Transformer's. However, despite the MA model maintaining similar efficiency metrics, including inference speed and memory usage, on the WMT14 EN-DE dataset, the introduction of cross-attention undoubtedly affects the model's scalability for longer sequences. In light of these findings, this section will discuss the following two questions:

Why does Mamba excel in Language Modeling (LM) but not in MT?

An intuitive question arises: Mamba has been shown to achieve state-of-the-art performance in LM, but why does it struggle with long sequence inputs in MT tasks? This project posits that two primary factors contribute to this issue. Firstly, MT tasks are more challenging than LM, requiring the model to have a stronger ability to capture and recognize long-term dependencies. LM only needs to consider the dependencies between previous target tokens, while MT must account for dependencies among source tokens as well as those between source and target tokens. The second reason is that the implementation of the Mamba only supports processing a single sequence, which aligns well with the characteristics of LM. When dealing with MT tasks, there is currently no good solution to aggregate features from both source and target sequences, so Mamba can only concatenate the two sequences into one for processing, making it difficult for the model to extract the relationships between the two sequences. After applying cross-attention, the model's translation quality reached the same level as the Transformer's, which supports this hypothesis. Therefore, in order for Mamba to better handle MT, or more generally, tasks that use an encoder-decoder architecture, a future research direction is to implement a Mamba-based block that can combine features from two sequences, similar to the cross-attention block.

Potential of Mamba-based Implicit Cross-Attention

The MA model employs cross-attention, which may limit the efficiency advantages of the Mamba model as sequence lengths increase. Therefore, is there a method to implement a Mamba-based Implicit Cross-Attention Block to address this issue? Theoretically, it is feasible, but further research is needed.

During the procedure of this project, the latest research by Mamba's authors, Gu et al.[10], highlighted the duality between structured SSM (S4, S6, etc.) and masked attention used in the auto-regressive model, masked diffusion model[15], etc.

$$y = M \circ (QK^T)V \qquad \text{(Mask Attention)} \tag{5.1}$$

$$y = A_M \circ (CB^T)X \qquad \text{(structured SSM)} \tag{5.2}$$

Where M is the mask of the attention matrix in Transformer, and A_M is a matrix transformed (detail shown in Appendix D) by A in structured SSM. Additionally, the formulas omit the softmax function and the scaling factor in the masked attention for simplicity in computation. This set of formulas reveals the unified form between masked attention and structured SSM. Moreover, in Mamba, the matrices B and C are also input-dependent, which allows it to establish an equivalence between the components of masked attention and structured SSM, where Q is equivalent to C, K is equivalent to B, and V is equivalent to the input X. Thus, it is theoretically sufficient to define the matrix C as a function of another sequence to obtain an implicit cross-attention variant of Mamba that simultaneously processes dependencies between two sequences. Here is the equation given two sequences X_1 and X_2:

$$y = A_M \circ (S_C(X_1) \cdot S_B^T(X_2))X_2 \tag{5.3}$$

Where S_c and S_B are linear projections to make the B and C data-dependent in the Mamba block. However, a constraint of utilizing this duality is that the lengths of the two sequences must be equal, which poses a challenge because the Mamba-based models in this project are auto-regressive. In these models, the sequence lengths of the encoder output and decoder input during the generation process are inherently different. As a result, the current implementation of Mamba makes it challenging to integrate this module within the auto-regressive model used in this project. Nevertheless, inspired by recent studies [11, 37, 5, 12], employing an encoder-decoder diffusion model as a replacement for traditional auto-regressive methods in translation generation could be a potential further direction. This approach can potentially overcome the issue of differing lengths between the encoder output and decoder input.

Chapter 6

Conclusion and Further Direction

6.1 Conclusion

This project seeks to answer whether the Mamba model, known for its high efficiency, hidden-attention property, and linear complexity relative to sequence length, can become a viable replacement for attention mechanisms in NMT systems. Specifically, the project aims to determine if the Mamba model can provide more efficient memory usage and faster execution speeds without sacrificing translation quality. Therefore, this project researched existing NMT models and datasets, then selected the WMT14 EN-DE dataset for training and the Transformer as the dominant baseline, which utilizes both self-attention and cross-attention mechanisms. Then, this project implemented an encoder-decoder model based solely on stacked Mamba blocks (Mamba Base Model), as well as another model that employed cross-attention (MA Model). By evaluating the model's translation quality and efficiency, the results show that the Mamba base model achieved an inference speed of 5-7× faster than the Transformer but struggled to translate long sequence sentences. In contrast, the MA model attained competitive translation quality compared to the Transformer while maintaining efficiency. Additionally, the project conducted ablation experiments on the MA model, where various components of the Transformer were used to replace those in the MA model. None of these replacements impacted the translation quality, indicating that Mamba can effectively replace both the self-attention and the MLP block. To obtain a deeper understanding of the MA model, this project also assessed the model's capabilities in handling specific linguistic phenomena and visualized the implicit attention distribution. It was found that the MA model can handle word-level and most sentence-level linguistic phenomena well, especially performing better when processing unseen words and ambiguity constituents than the Transformer, though it slightly underperforms in managing complex long-range dependencies. Meanwhile, the MA model exhibits a clearer and uniform attention distribution in the visualized heatmap, which indicates a robust ability to extract local fine-grained features.

Overall, the Mamba model is an efficient linear variant of self-attention. The model, which consists of Mamba and cross-attention, achieves translation quality that is competitive to the Transformer on sentence-level translation tasks and even excels in handling ambiguity and unseen words. It also offers faster inference speed and better scalability when processing long sequences. Additionally, adjusting the state space dimension allows for a flexible trade-off between translation quality and efficiency. These properties make Mamba a more efficient replacement for the self-attention and MLP modules in the Transformer, while replacing the cross-attention block still requires further research.

6.2 Limitation and Further Direction

While the Mamba model can efficiently replace the self-attention layer with linear complexity concerning sequence length, the Mamba model is designed to handle only a single sequence and cannot compute correlations between tokens of two sequences like cross-attention can. Therefore, cross-attention remains an irreplaceable component in encoder-decoder auto-regressive NMT systems (such as the MA Model). This limitation results in computational and memory bottlenecks as the sequence length increases, like the Transformer. Consequently, future research will focus on two main directions:

- Develop an encoder-decoder diffusion NMT system with Mamba's cross-attention variant shown in equation 5.3 to replace the original auto-regressive model. This approach allows the model to generate the entire sequence at once rather than generating it token by token, enabling a target length predictor [18, 25] to effectively achieve equal sequence and target lengths. Furthermore, this research direction has significant scalability in other fields, as it can also be applied to text summarization, multi-modal applications, etc.

- Due to the relatively short sequence lengths in sentence-level MT tasks, the Mamba's efficiency for long sequences is not adequately demonstrated. Therefore, the MA model and the diffusion NMT implemented in future research should also be tested on document-level datasets. However, it is important to note models of document-level and sentence-level tasks involve architectural differences.

Bibliography

[1] Ameen Ali, Itamar Zimerman, and Lior Wolf. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*, 2024.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58, 2014.

[5] Linyao Chen, Aosong Feng, Boming Yang, and Zihui Li. Xdlm: Cross-lingual diffusion language model for machine translation, 2023.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] NVIDIA Corporation. A PyTorch Extension: Tools for easy mixed precision and distributed training in Pytorch. https://github.com/NVIDIA/apex, 2024.

[8] Tri Dao. causal-conv1d: Causal depthwise conv1d in CUDA, with a PyTorch interface. https://github.com/Dao-AILab/causal-conv1d, 2024.

[9] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

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

[11] Do Huu Dat, Do Duc Anh, Anh Tuan Luu, and Wray Buntine. Discrete diffusion language model for long text summarization, 2024.

[12] Yunus Demirag, Danni Liu, and Jan Niehues. Benchmarking diffusion models for machine translation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 313–324, 2024.

[13] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, 2017.

[14] Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

[15] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer, 2024.

[16] Yingbo Gao, Christian Herold, Zijian Yang, and Hermann Ney. Is encoder-decoder redundant for neural machine translation?, 2022.

[17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.

[18] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models, 2019.

[19] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[20] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

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

[22] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020.

[23] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[24] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.

[25] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. Non-autoregressive neural machine translation, 2018.

[26] Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael R Lyu, and Shuming Shi. Towards understanding neural machine translation with word importance. *arXiv preprint arXiv:1909.00326*, 2019.

[27] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[28] Tzu-Wei Huang. TensorBoardX: tensorboard for pytorch (and chainer, mxnet, numpy, ...). https://github.com/lanpa/tensorboardX, 2023.

[29] Lukasz Kaiser, Aidan N Gomez, and Francois Chollet. Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*, 2017.

[30] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. Findings of the 2023 conference on machine translation (wmt23): Llms are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, 2023.

[33] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in neural information processing systems*, 34:9204–9215, 2021.

[34] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation, 2020.

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

[36] Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. A survey on document-level neural machine translation: Methods and evaluation. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

[37] Eliya Nachmani and Shaked Dovrat. Zero-shot translation using diffusion models, 2021.

[38] Toshiaki Nakazawa, Kazutaka Kinugawa, Hideya Mino, Isao Goto, Raj Dabre, Shohei Higashiyama, Shantipriya Parida, Makoto Morishita, Ondřej Bojar, Akiko Eriguchi, et al. Overview of the 10th workshop on asian translation. In *Proceedings of the 10th Workshop on Asian Translation*, pages 1–28, 2023.

[39] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020.

[40] University of Edinburgh. Edinburgh compute and data facility web site. http://www.ecdf.ed.ac.uk, 2024.

[41] University of Edinburgh. Edinburgh parallel computing centre web site. https://www.epcc.ed.ac.uk/, 2024.

[42] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[43] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[45] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5:606–624, 2023.

[46] Maja Popović. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395, 2015.

[47] Matt Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

[48] Yanyuan Qiao, Zheng Yu, Longteng Guo, Sihan Chen, Zijia Zhao, Mingzhen Sun, Qi Wu, and Jing Liu. Vl-mamba: Exploring state space models for multimodal learning, 2024.

[49] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*, 2020.

[50] Elizabeth Salesky, Marcello Federico, and Marine Carpuat. Proceedings of the 20th international conference on spoken language translation (iwslt 2023). In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, 2023.

[51] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[53] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[54] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[55] Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[56] Arnold Tustin. A method of analysing the behaviour of linear systems in terms of time series. *Journal of the Institution of Electrical Engineers-Part IIA: Automatic Regulators and Servo Mechanisms*, 94(1):130–142, 1947.

[57] Ali Vardasbi, Telmo Pessoa Pires, Robin M Schmidt, and Stephan Peitz. State spaces aren't enough: Machine translation needs attention. *arXiv preprint arXiv:2304.12776*, 2023.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[59] Shuo Wang, Zhaopeng Tu, Zhixing Tan, Wenxuan Wang, Maosong Sun, and Yang Liu. Language models are good translators, 2021.

[60] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

[61] Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. *arXiv preprint arXiv:2004.11867*, 2020.

Appendix A

Complete Translation Quality Evaluation Results

This appendix provides the ChrF and COMET scores for all models conducted in the project, tested on different length subsets, while the BLEU scores are already presented in the main text. The ChrF and COMET results generally align with the BLEU score in the main text, except that the Transformer model outperforms all the Mamba-based models in the COMET metric, which indicates a better semantic extraction capability.

A.1 ChrF Score

	Short (2-19)	Medium (20-30)	Long (31-80)	Overall
Transformer Base	54.99 ± 1.06	**55.99 ± 0.77**	**57.38 ± 0.64**	**56.53 ± 0.44**
Mamba Base	54.65 ± 1.05	54.76 ± 0.82	45.42 ± 1.36	49.80 ± 0.79
Mamba Attention	54.58 ± 1.11	55.55 ± 0.79	56.39 ± 0.69	55.79 ± 0.47
Mamba-MLP	**55.23 ± 1.07**	55.33 ± 0.84	56.89 ± 0.66	56.10 ± 0.46
Self-Attention-Mamba	55.05 ± 1.11	55.39 ± 0.78	57.12 ± 0.64	56.20 ± 0.45
$\text{Mamba}_{EN} - \text{Transformer}_{DE}$	54.47 ± 1.06	55.58 ± 0.82	56.71 ± 0.64	55.95 ± 0.47
$\text{Transformer}_{EN} - \text{Mamba}_{DE}$	53.73 ± 1.04	54.88 ± 0.78	56.38 ± 0.62	55.45 ± 0.44
Mamba Attention Positional	53.83 ± 1.14	54.78 ± 0.80	55.92 ± 0.72	55.19 ± 0.47

Table A.1: ChrF score with a standard deviation for all models test on the WMT14 EN-DE test set, for different source sentence lengths.

A.2 COMET Score

	Short (2-19)	Medium (20-30)	Long (31-80)	Overall
Transformer Base	**0.841 ± 0.124**	**0.832 ± 0.104**	**0.801 ± 0.100**	**0.825 ± 0.111**
Mamba Base	0.784 ± 0.126	0.768 ± 0.108	0.675 ± 0.168	0.741 ± 0.146
Mamba Attention	0.789 ± 0.130	0.770 ± 0.106	0.738 ± 0.112	0.765 ± 0.119
Mamba-MLP	0.795 ± 0.121	0.773 ± 0.107	0.746 ± 0.103	0.771 ± 0.113
Self-Attention-Mamba	0.793 ± 0.123	0.772 ± 0.107	0.747 ± 0.106	0.771 ± 0.114
$\text{Mamba}_{EN} - \text{Transformer}_{DE}$	0.786 ± 0.128	0.771 ± 0.110	0.738 ± 0.108	0.765 ± 0.118
$\text{Transformer}_{EN} - \text{Mamba}_{DE}$	0.775 ± 0.135	0.761 ± 0.115	0.737 ± 0.106	0.758 ± 0.121
Mamba Attention Positional	0.787 ± 0.125	0.770 ± 0.107	0.739 ± 0.108	0.766 ± 0.115

Table A.2: COMET score with a standard deviation for all models test on the WMT14 EN-DE test set, for different source sentence lengths.

Appendix B

More Linguistic Phenomena Examples

This appendix provides more evidence for the linguistic phenomena analysis in the main text, where RED means bad translation, GREEN mean good translation and **bold texts** means keywords for the linguistic phenomena.

B.1 Word-Level

Unseen Words:

Example 1:

S: Many critics of **veganism** warn in particular of the lack of vitamin B12.

T: Insbesondere vor dem Mangel an Vitamin B12 warnen viele **Vegansimus**-Kritiker.

MA: Viele Kritiker des **Veganismus** warnen insbesondere vor dem Mangel an Vitamin B12.

TR: Viele Kritiker des **Veganismus** warnen insbesondere vor dem Mangel an Vitamin B12.

Example 2:

S: Nothing is more **quintessentially** Halloween than haunted houses.

T: Nichts gehört **mehr zu** Halloween als Häuser, in denen es spukt.

MA: Nichts ist mehr **quintessenz** als Halloween Häuser verfolgt.

TR: Nichts ist **eher** Halloween als Haunted Häuser.

Example 3:

S: However, their repertoire also includes emotive **waltzes** and a full big band sound.

T: Zu ihrem Repertoire gehören aber auch gefühlvolle **Walzer** und ein satter Big-Band-Sound.

MA: Ihr Repertoire umfasst jedoch auch emotionale **Walzer** und einen großen Band-sound.

TR: Zu ihrem Repertoire gehören aber auch emotionale **Walzen** und ein ganzer Big Band Sound.

Synonym:

Example 1:

S: All those involved will be **happy** with that evaluation.

T: Damit werden alle Beteiligten **leben können**.

MA: Alle Beteiligten werden mit dieser Bewertung **zufrieden** sein.

TR: Alle Beteiligten werden mit dieser Bewertung **zufrieden** sein.

Example 2:

S: **July**.

T: **Wiederkehr feiern**.

MA: **Juli**.

TR: **Juli**.

Example 3:

S: This and another **bedroom** were completely burnt out.

T: Dieses und ein weiteres **Zimmer** brannten vollständig aus.

MA: Dieses und ein weiteres **Schlafzimmer** wurden komplett ausgebrannt.

TR: This and another **bedroom** were completely burnt out.

Proper Noun and Acronym:

Example 1:

S: **Edward Snowden**, as witnessed by Hans-Christian Ströbele

T: **Edward Snowden** bezeugt durch Hans-Christian Ströbele

MA: **Edward Snowden**, wie von Hans-Christian Ströbele beobachtet.

TR: **Edward Snowden**, wie er von Hans-Christian Ströbele gesehen wurde

Example 2:

S: **NSA** revelations boost corporate paranoia about state surveillance

T: **NSA**-Enthüllungen verstärken Firmenparanoia wegen staatlicher Überwachung

MA: Die Enthüllungen der **NSA** fördern die Paranoia der Unternehmen hinsichtlich der staatlichen Überwachung

TR: **NSA**-Enthüllungen verstärken die Paranoia der Unternehmen in Bezug auf staatliche Überwachung

Example 3:

S: Delta and **JetBlue** were among the airliners who have already submitted plans.

T: Unter diesen Fluggesellschaften waren auch Delta und **JetBlue**.

MA: Delta und **JetBlue** waren unter den Airlinern, die bereits Pläne vorgelegt haben.

TR: Delta und **JetBlue** gehörten zu den Fluglinien, die bereits Pläne eingereicht haben.

Morphemes:

Example 1:

S: Both ideas **were rejected**.

T: Beides wurde **wieder verworfen**.

MA: Beide Ideen **wurden abgelehnt**.

TR: Beide Ideen **wurden abgelehnt**.

Example 2:

S: The **accused** initially **remained** silent.

T: Die **Angeklagten schwiegen** zum Auftakt.

MA: Die **Angeklagten schwiegen** zunächst.

TR: Die **Beschuldigten schwiegen** zunächst.

Example 3:

S: **Children 's dreams** come true

T: **Kinderträume** werden wahr

MA: **Kinderträume** werden wahr.

TR: **Kinderträume** werden wahr

B.2 Sentence-Level

Constituent Structure:

Example 1:

S: Except: In **the stomachs of those in the passenger and back seats**, hunger strikes.

T: Nur: In **den Bäuchen der Leute auf dem Beifahrer- und Rücksitz** macht sich der Hunger bemerkbar.

MA: In **den Magen der in den Passagier- und Rücksitzen**, Hungerstreiks.

TR: Ausser: **In den Magen der im Passagier und im Rücken Sitze**, Hungerstreik.

Example 2:

S: The **three ship unloaders on the bridge and the second transport belt** could see this rise to 10 million.

T: Die **drei Schiffsentlader auf der Brücke sowie das zweite Transportband** könnten bis zu 10 Millionen schaffen.

MA: Die **drei Schiffsentlader auf der Brücke und der zweite Transportgurt** konnten diesen Anstieg auf 10 Millionen sehen.

TR: Die **drei Entlader auf der Brücke und der zweite Transportgürtel** konnten diesen Anstieg auf 10 Millionen sehen.

Example 3:

S: Canadian plane and train maker Bombardier Inc reported a 15 percent fall in net profit on Thursday, pressured by **fewer aircraft orders and deliveries in the third quarter and contract issues** in its train unit.

T: Der kanadische Flugzeug- und Eisenbahnhersteller Bombardier Inc meldete am Donnerstag einen 15-prozentigen Rückgang des Nettogewinns, nachdem er durch **rückläufige Bestellungen und Auslieferungen bei Flugzeugen im dritten Quartal sowie Vertragsprobleme** in der Eisenbahnsparte unter Druck geraten war.

MA: Der kanadische Flugzeug- und Bahnhersteller Bombardier Inc berichtete über einen Rückgang des Nettogewinns um 15 Prozent, unter dem Druck von **weniger Flugzeugbestellungen und Auslieferungen im dritten Quartal und Vertragsproblemen** in seiner Zugeinheit.

TR: Der kanadische Flugzeug- und Bahnhersteller Bombardier Inc meldete am Donnerstag einen Rückgang von 15 Prozent am Nettogewinn, unter dem Druck von **weniger Flugzeugaufträgen und Auslieferungen im dritten Quartal sowie von Vertragsemissione**n im Eisenbahnbereich.

Interrogative Sentence:

Example 1:

S: **How did** the universe come about and what does it consist of?

T: **Wie ist** das Universum entstanden und woraus besteht es?

MA: **Wie ist** das Universum entstanden und worin besteht es?

TR: **Wie entstand** das Universum und worin besteht es?

Example 2:

S: **Is** Europe’s elite ready to do business with Britain?

T: **Ist** Europas Elite bereit, mit Großbritannien Geschäfte zu machen?

MA: **Ist** die Elite Europas bereit, mit Großbritannien Geschäfte zu machen?

TR: **Ist** Europas Elite bereit, Geschäfte mit Großbritannien zu machen?

Example 3:

S: **What are** the basic physical laws of the Universe?

T: **Was sind** die grundlegenden physikalischen Gesetze des Universums?

MA: **Was sind** die grundlegenden physikalischen Gesetze des Universums?

TR: **Was sind** die grundlegenden physikalischen Gesetze des Universums?

Passive Voice:

Example 1:

S: The evaluations **were** already **made** on Thursday.

T: Die Bewertungen **wurden** bereits am Donnerstag **vorgenommen**.

MA: Die Bewertungen **wurden** bereits am Donnerstag **abgegeben**.

TR: Die Bewertungen **wurden** bereits am Donnerstag **vorgenommen**.

Example 2:

S: Google **is accused** of infringing seven patents.

T: Google **wird** in sieben Fällen der Patentverletzung **bezichtigt**.

MA:Google **wird beschuldigt**, sieben Patente verletzt zu haben.

TR: Google **wird vorgeworfen**, sieben Patente verletzt zu haben.

Example 3:

S: Should this election **be decided** two months after we stopped voting?

T: Sollten diese Wahlen zwei Monate nach dem Ende der Stimmabgabe **entschieden werden**?

MA:Sollte diese Wahl zwei Monate nach unserer Abstimmung **entschieden werden**?

TR: Sollte diese Wahl zwei Monate, nachdem wir die Abstimmung **gestoppt**?

Adverbial Clause:

Example 1:

S: After five years of robust growth since the global financial crisis, and cheap credit fuelled by loose monetary policy in advanced economies, lower- and middle-income families are turning to **pawn shops** to make up the difference as their economies slow.

T: Nach fünf Jahren robusten Wachstums seit der globalen Finanzkrise und billigen Krediten aufgrund einer lockeren Finanzpolitik in den entwickelten Wirtschaftsräumen suchen Familien mit geringeren und mittleren Einkommen **Pfandhäuser** auf, um so bei stotternder Wirtschaft den Unterschied auszugleichen.

MA: Nach fünf Jahren robusten Wachstums seit der globalen Finanzkrise und billigen Krediten, die von lockerer Geldpolitik in den hoch entwickelten Volkswirtschaften angeheizt werden, wenden sich Familien mit niedrigen und mittleren Einkommen an die **Spielerläden**, um die Differenz auszugleichen, da ihre Wirtschaft langsam voranschreitet.

TR: Nach fünf Jahren robusten Wachstums seit der globalen Finanzkrise und billigen Krediten, die in entwickelten Volkswirtschaften durch eine lockere Geldpolitik angeheizt wurden, wenden sich Familien mit niedrigen und mittleren Einkommen an **Pfandhäuser**, um den Unterschied auszugleichen, wenn sich ihre Volkswirtschaften verlangsamen.

Example 2:

S: When designing the early Internet services, the focus lay on **making communication possible**.

T: Bei der Konzeption der frühen Internetdienste stand im Vordergrund, **Kommunikation möglich zu machen**.

MA: Bei der Gestaltung der frühen Bei der Gestaltung der frühen Internet-Dienste lag der Schwerpunkt auf der **Kommunikation**.

TR: Bei der Gestaltung der frühen Internet-Dienste lag der Fokus auf **der Ermöglichung der Kommunikation.**

Example 3:

S: With her dog **Woody** competing in the Class 1 competition, **Susi Höpp** was subject to the critical scrutiny of the judge.

T: **Susi Höpp** stellte sich mit ihrem **Woody** in der Klasse 1 den kritischen Blicken des Leistungsrichters.

MA: **Susi Höpp wurde von Woody** in der Klasse 1 unter kritischer Kontrolle des Richters gestellt.

TR: Mit ihrem Hund **Woody**, der in der Klasse 1 konkurrierte, unterlag **Susi Höpp** der kritischen Prüfung des Richters.

Emphatic Inversion:

Example 1:

S: "**Not only have you** kept countless records for us, but you have also done so much running around for us, and for this we offer our sincere thanks," said Choir Chairman Erich Schlotmann.

T: "**Du hast nicht nur** viel für uns aufgeschrieben, sondern hast auch so manche Runde für uns gedreht, dafür unser herzliches Dankeschön", so der Vorsitzende des Chores Erich Schlotmann.

MA: "**Sie haben uns nicht nur** unzählige Rekorde geführt, sondern Sie haben uns auch so umhergelaufen, und dafür bedanken wir uns herzlich", sagte Choir-Vorsitzender

Erich Schlotmann.

TR: ”**Sie haben nicht nur** unzählige Aufzeichnungen für uns geführt, sondern Sie haben auch so viel für uns getan, und dafür möchten wir uns herzlich bedanken”, sagte Erich Schlotmann, Chorvorsitzender.

Example 2:

S: Only when goalkeeper, Roman Weidenfeller, was the first BVB player to step onto the field, **did cheers briefly erupt**, as is usually the case.

T: Nur als Torhüter Roman Weidenfeller wie immer als erster BVB-Spieler den Platz betrat, **brandete kurzzeitiger Jubel auf**.

MA: Erst als der Torhüter, Roman Weidenfeller, als erster BVB-Spieler auf das Spielfeld stieß, **brach kurz ein Jubel aus**, wie es normalerweise der Fall ist.

TR: Erst als der Torhüter, Roman Weidenfeller, als erster BVB-Spieler auf das Feld kam, **kam es zu einem kurzen Eklat**, wie es normalerweise der Fall ist.

Example 3:

S: Indeed, **such is demand** across parts of southeast Asia - where household debt is rising - that ValueMax, where she is carrying out her transaction, this week became the third pawnshop to list on the Singapore stock exchange.

T: Tatsächlich **ist die Nachfrage** in Teilen Südostasiens - wo die Verschuldung der Haushalte zunimmt - **so groß**, dass ValueMax, wo sie ihren Tausch vorgenommen hat, diese Woche das dritte Pfandhaus wurde, das an der singapurischen Börse gelistet ist.

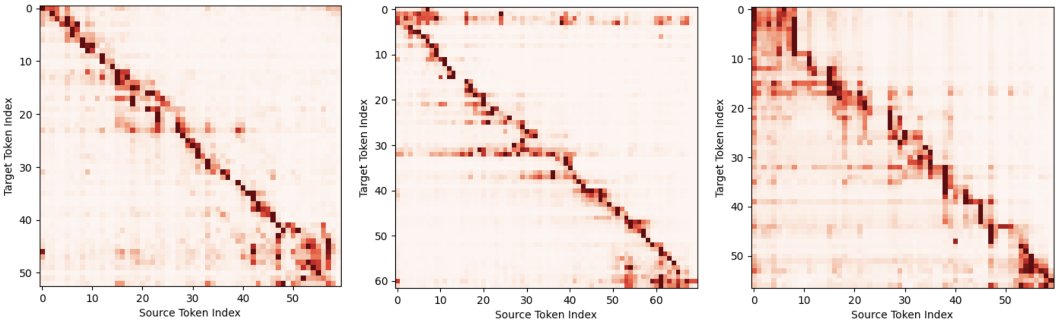
MA: Tatsächlich **ist diese Nachfrage** in Teilen Südostasiens - wo die Verschuldung der Haushalte steigt - **so hoch**, dass ValueMax, wo sie ihre Transaktion durchführt, diese Woche zum dritten Einzelhändler wurde, der an der Börse in Singapur verzeichnet wurde.

TR: Tatsächlich **ist die Nachfrage** in Teilen Südostasiens - wo die Haushaltsschulden steigen - **so groß**, dass ValueMax, wo sie ihre Transaktion durchführt, diese Woche zum dritten Pfandladen wurde, der an der Börse von Singapur auflistet.

Appendix C

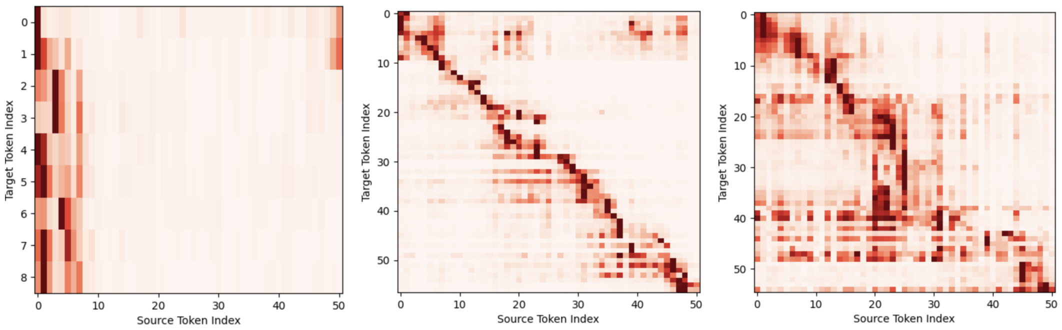
More Attention Distribution Visualization Examples

This appendix presents additional attention distribution visualizations. The results are consistent with the analysis in the main text, showing that the Mamba Base model struggles with long sentences, often producing overly short translations, while the Transformer tends to generate blurred attention patterns. The MA model, however, demonstrates a clear and uniform attention distribution. For short sequences, both the Mamba Base and MA models exhibit more detailed attention distributions, indicating a more fine-grained feature extraction capability, where the transformer only attends to a few headwords of sentences.



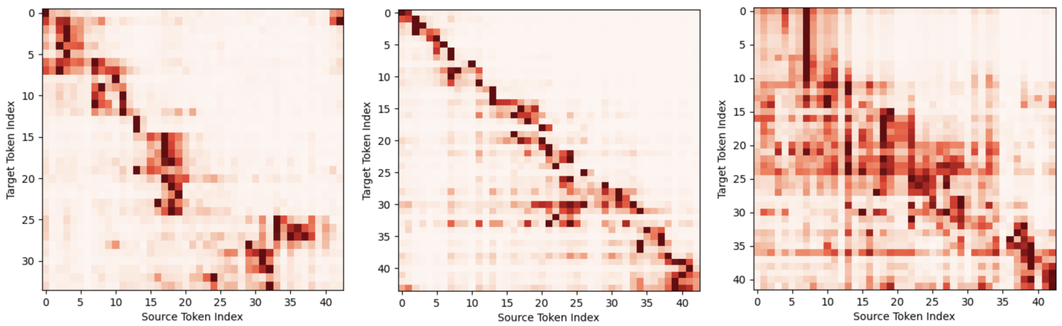
(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.1: Attention Heatmap for a Long Sample (70 Tokens)



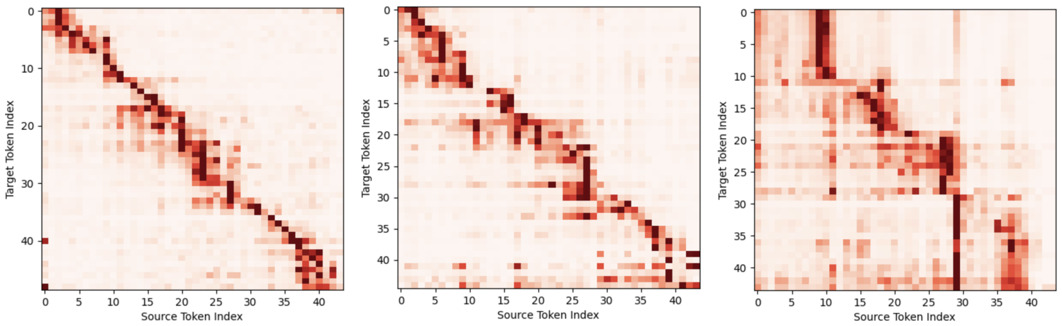
(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.2: Attention Heatmap for a Long Sample (51 Tokens)



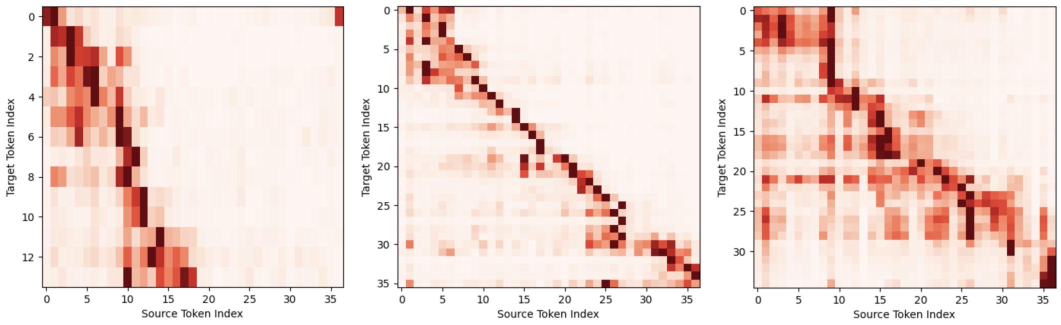
(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.3: Attention Heatmap for a Long Sample (43 Tokens)



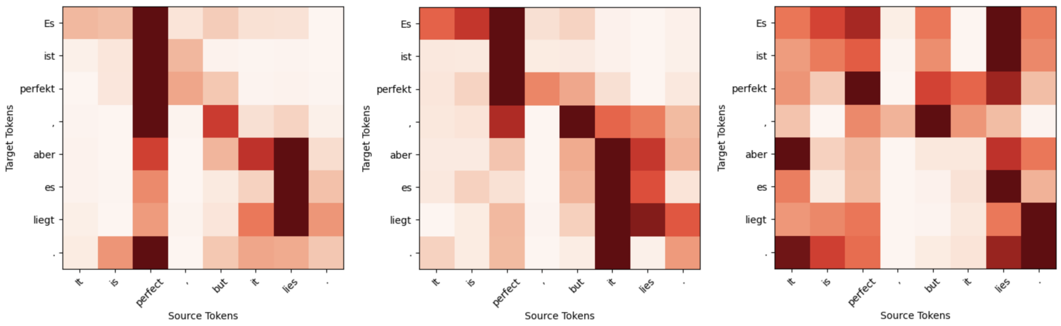
(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.4: Attention Heatmap for a Long Sample (44 Tokens)



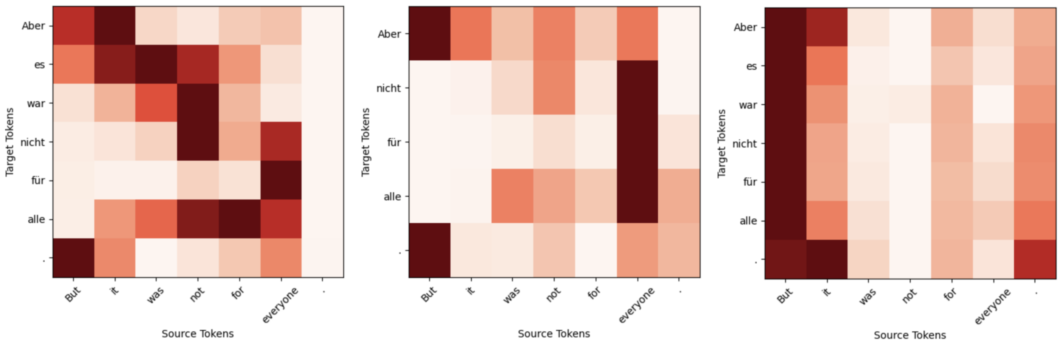
(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.5: Attention Heatmap for a Long Sample (37 Tokens)



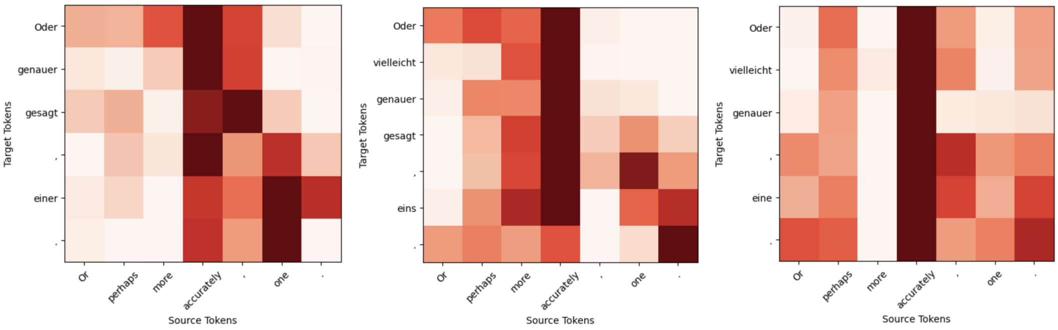
(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.6: Attention Heatmap for a Short Sample (8 tokens, ”It is perfect, but it lies.”)



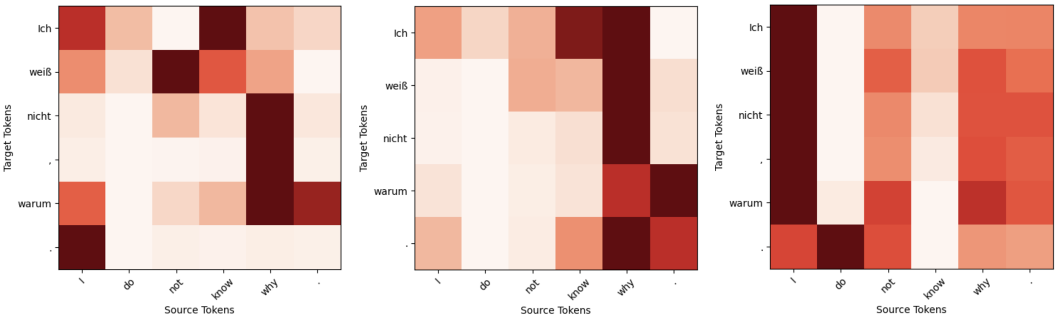
(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.7: Attention Heatmap for a Short Sample (7 tokens, ”But it was not for everyone.”)



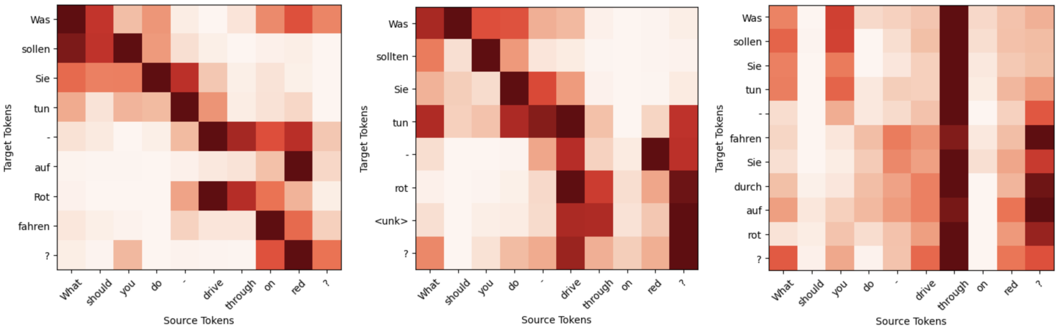
(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.8: Attention Heatmap for a Short Sample (7 tokens, ”Or perhaps more accurately, one.”)



(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.9: Attention Heatmap for a Short Sample (6 tokens, ”I do not know why.”)



(a) Mamba Base Model (b) Mamba Attention Model (c) Transformer Model

Figure C.10: Attention Heatmap for a Short Sample (10 tokens, ”What should you do, drive through on red?”)

Appendix D

Linear Transformation for Matrix A

This appendix will introduce the detail of the linear transformation applied on matrix A to get mask matrix A_M in Equation 5.2.
Given the equation of SSM introduced in Chapter 2:

$$h_k = Ah_{k-1} + Bx_k, \quad y_k = Ch_k \tag{D.1}$$

Expand the state equation (Left) through iteration and obtain the following equation:

$$h_k = A_k \dots A_1 B_0 x_0 + A_k \dots A_2 B_1 x_1 + A_k B_{k-1} x_{k-1} + B_k x_k \tag{D.2}$$

$$= \sum_{i=0}^{k} A^{\times}_{k:s} B_i x_i. \tag{D.3}$$

Multiplying by C to obtain another expression of SSM:

$$y_k = \sum_{i=0}^{k} C_k^{\top} A^{\times}_{k:i} B_i x_i \tag{D.4}$$

$$y = Mx \tag{D.5}$$

$$M_{ji} = C_j^{\top} A_j \cdots A_{i+1} B_i = C_j^{\top} G_{ji} B_i \tag{D.6}$$

$$G_{ji} = A_j \cdots A_{i+1} \tag{D.7}$$

In the newest research of the Mamba model, Gu et al.[10] indicate when instantiating matrix A for each time step in an extremely structured way: $A = aI$, where a is a scalar, and I is an identity matrix. The matrix G can be converted into the following form, which is mask matrix A_M introduced in Equation 5.2:

$$A_M = \begin{bmatrix} 1 & & & & \\ a_1 & 1 & & & \\ a_2 a_1 & a_2 & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ a_{k-1} \dots a_1 & a_{k-1} \dots a_2 & \dots & a_{k-1} & 1 \end{bmatrix} \tag{D.8}$$