

DocMamba: Efficient Document Pre-training with State Space Model

Pengfei Hu^{1*}, Zhenrong Zhang^{1,2*†}, Jiefeng Ma¹, Shuhang Liu¹, Jun Du^{1‡}, Jianshu Zhang²

¹NERC-SLIP, University of Science and Technology of China

²iFLYTEK Research

{pengfeihu, zZR666, jfma, liush19}@mail.ustc.edu.cn, jundu@ustc.edu.cn, jszhang6@iflytek.com

Abstract

In recent years, visually-rich document understanding has attracted increasing attention. Transformer-based pre-trained models have become the mainstream approach, yielding significant performance gains in this field. However, the self-attention mechanism’s quadratic computational complexity hinders their efficiency and ability to process long documents. In this paper, we present DocMamba, a novel framework based on the state space model. It is designed to reduce computational complexity to linear while preserving global modeling capabilities. To further enhance its effectiveness in document processing, we introduce the Segment-First Bidirectional Scan (SFBS) to capture contiguous semantic information. Experimental results demonstrate that DocMamba achieves new state-of-the-art results on downstream datasets such as FUNSD, CORD, and SORIE, while significantly improving speed and reducing memory usage. Notably, experiments on the HRDoc confirm DocMamba’s potential for length extrapolation.

Code — <https://github.com/Pengfei-Hu/DocMamba>

Introduction

With the prosperity of commercial activities in today’s society, a broad range of documents are used to convey information, leading to a growing demand for document processing (Cui et al. 2021). In order to reduce the labor-intensive workflows associated with this, Visually-rich Document Understanding (VrDU) (Xu et al. 2020a) is drawing considerable attention from both academia and industry. It aims to automate information extraction from documents (Zhang et al. 2024b; Hu et al. 2024) and support various applications.

In recent years, Transformer-based (Vaswani et al. 2017) pre-training models have made substantial advancements in VrDU and become the mainstream practice. The pioneering model, LayoutLM (Xu et al. 2020a), encodes both textual and layout information through an architecture similar to BERT (Devlin et al. 2018). Subsequent research (Li et al. 2021; Xu et al. 2020b; Huang et al. 2022) has incorporated additional visual models as image encoders, enabling

*These authors contributed equally.

†Work done during an internship at iFLYTEK Research.

‡Corresponding author.

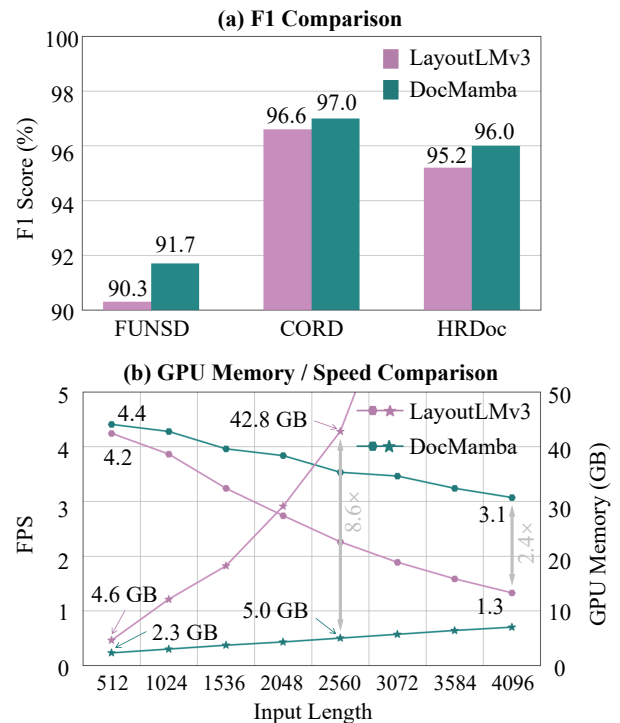


Figure 1: Performance and efficiency comparisons between LayoutLMv3 (Huang et al. 2022) and our DocMamba.

the joint modeling of visual, textual, and layout information. However, despite the significant boost provided by Transformer, its quadratic complexity with respect to input length limits its ability to handle long texts. For instance, the context size in the LayoutLM series (Xu et al. 2020a,b; Huang et al. 2022) is restricted to 512. Consequently, when processing text-dense documents, these models often need to incorporate a sliding window strategy, which can lead to the loss of global information and an increase in processing time.

To achieve sub-quadratic complexity, one promising approach is the substitution of the Transformer with State Space Models (SSMs) (Gu et al. 2022a). They originated from the foundational classic state space model (Kalman 1960), and are notable for their capabilities in linear-time inference, highly parallelized training, and robust perfor-

mance in tasks requiring long-context processing. Examples include the linear state space layers (LSSL) (Gu et al. 2021a) and the structured state-space sequence model (S4) (Gu, Goel, and Ré 2021). A recent addition to this category, Mamba (Gu and Dao 2023), has demonstrated exceptional results through its selective mechanism and hardware-aware design. Unlike self-attention mechanism in that each token interacts with all others within the context, Mamba enables each token to garner contextual knowledge solely through a compressed hidden state, thereby reducing the quadratic complexity to linear. Mamba has shown performance comparable to the Transformer in various fields (Lieber et al. 2024; Zhu et al. 2024). Given the inherently longer sequences produced by documents, a natural question arises: Can Mamba work well for VrDU?

Motivated by this, we introduce DocMamba, a purely SSM-based model tailored for VrDU. It boasts linear complexity relative to input length, making it ideal for long documents. While vanilla Mamba is designed to process 1-D sequences, tokens in documents exhibit complex 2-D layouts and form continuous semantic content alongside their neighbors. Thus, it is necessary to consecutively process tokens belonging to the same segment (e.g., titles, paragraphs, captions). For this purpose, we design the Segment-First Bidirectional Scan (SFBS). Initially, we leverage existing document layout analysis systems (Zhong, Tang, and Yepes 2019) to extract segments. DocMamba then sequentially scans all tokens within one segment before shifting to the next. Considering that incorporating context from both directions enhances the performance of language models (Devlin et al. 2018), we adopt the bidirectional scan strategy following Vim (Zhu et al. 2024). Furthermore, due to the inherent positional information within SSMs, DocMamba does not require 1-D position embeddings, which are indispensable in Transformer-based models. This feature endows DocMamba with the potential for length extrapolation.

We evaluate the performance of the pre-trained DocMamba using several publicly available benchmark datasets in downstream tasks. As depicted in Figure 1 (a), DocMamba surpasses the strong baseline LayoutLMv3 (Huang et al. 2022) at the base scale with a similar number of parameters across three datasets: the FUNSD dataset (Jaume, Ekenel, and Thiran 2019) for form understanding, the CORD (Park et al. 2019) dataset for receipt understanding, and the HRDoc (Ma et al. 2023) for semantic unit classification. Moreover, as shown in Figure 1 (b), tests on the HRDoc dataset show DocMamba has a faster inference speed and less GPU memory usage than LayoutLMv3. Especially with larger input lengths, DocMamba can save up to 88.3% of GPU memory and work 2.4 times faster, which can reduce the application costs significantly. This also proves the linear computational complexity of DocMamba. Furthermore, when the input length is restricted to 512 during both pre-training and fine-tuning, DocMamba still yields impressive results when the token length of test samples reaches 2560 for semantic unit classification on the HRDoc dataset. This validates DocMamba’s potential in length extrapolation. In conclusion, our research underscores the potential of SSMs as a powerful competitor with Transformer for VrDU, offer-

ing a simple yet effective baseline for future research.

Our main contributions are listed as follows:

- We delve into the SSM-based VrDU and propose a novel method, DocMamba, which exhibits linear complexity with respect to input length.
- We introduce the Segment-First Bidirectional Scan (SFBS) to enable Mamba, initially designed for 1-D sequences, to effectively process document tokens that possess complex 2-D layouts.
- Extensive experiments demonstrate that DocMamba exhibits promising performance compared to strong Transformer-based models, while maintaining faster speeds, lower memory consumption, and the potential for length extrapolation.

Related Work

Visually-rich Document Understanding

Early research (Yang et al. 2017; Hu et al. 2022) in VrDU typically utilizes unimodal or multimodal models with shallow fusion techniques. In recent years, the advent of pre-training techniques has revolutionized this field. BERT (Devlin et al. 2018) uses masked language models to obtain pre-trained deep bidirectional representations within pure text. Inspired by BERT, LayoutLM (Xu et al. 2020a) introduces 2-D spatial coordinate embeddings in addition to 1-D positional and text embeddings, thus simultaneously modeling the interaction between text and layout information within a singular framework. Furthermore, LayoutLMv2 (Xu et al. 2020b) adapts the standard Transformer by integrating a spatial-aware self-attention mechanism, and concatenates visual tokens with textual tokens to enhance text-image interactions. LayoutLMv3 (Huang et al. 2022) suggests learning cross-modal alignment with unified text and image masking. Additionally, various model architectures (Appalaraju et al. 2021; Gu et al. 2022b), attention mechanisms (Hong et al. 2022; Zhang et al. 2022) and self-supervised tasks (Tu et al. 2023; Luo et al. 2023; Zhang et al. 2024c; Yao, Li, and Xiao 2024) have been explored. However, nearly all of these methods are based on Transformer, which has a quadratic complexity concerning input length, thus posing challenges when processing lengthy documents.

State Space Models

State Space Models (SSMs) serve as a fundamental model applied across various fields such as control theory (Raibert 1977), signal processing (Rao and Arun 1992), and applied economics (Schulz and Werwatz 2004). Recently, SSMs have garnered renewed attention within the deep learning community (Gu et al. 2021a; Gu, Goel, and Ré 2021; Smith, Warrington, and Linderman 2023), demonstrating notable proficiency in capturing long-range dependencies. They afford highly efficient computation, either as a recurrence or convolution operation, with linear or near-linear scalability in sequence length. Mamba (Gu and Dao 2023), in particular, distinguishes itself by incorporating a time-varying selection mechanism and a hardware-aware parallel algorithm. The significant potential demonstrated by Mamba has inspired a succession of studies in areas like NLP (Lieber et al.

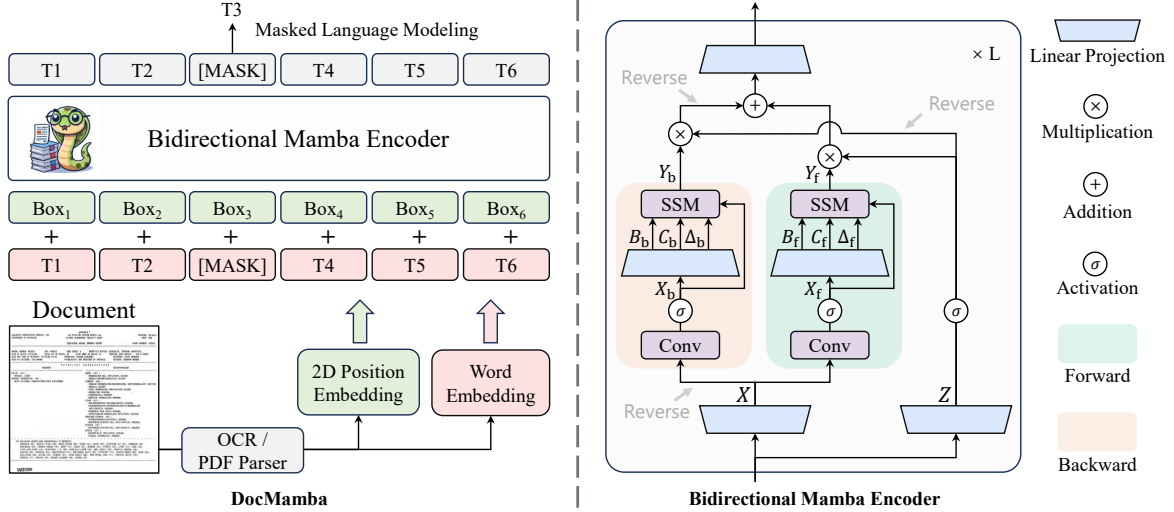


Figure 2: Framework of DocMamba (left) and Bidirectional Mamba Encoder (right).

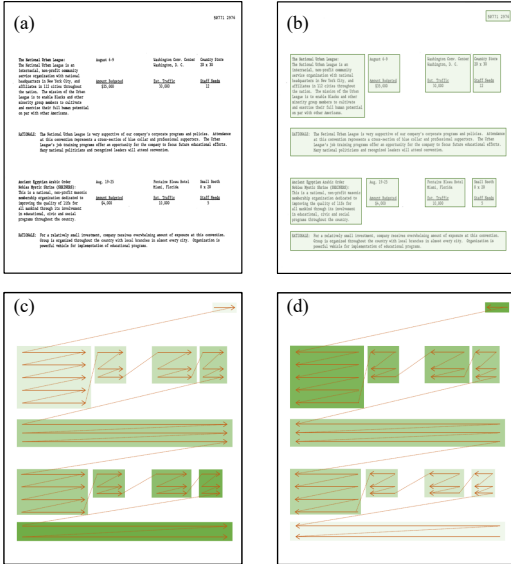


Figure 3: Depiction of Segment-First Bidirectional Scan.

2024; Dat et al. 2024), video understanding (Li et al. 2024; Lu, Salah, and Poppe 2024; Yao et al. 2024), speech processing (Zhang et al. 2024a), and more. However, the application of SSMs for VrDU still remains unexplored.

Preliminaries

State Space Model. The classical SSM represents a continuous system that maps an input $x(t) \in \mathbb{R}$ to an output $y(t) \in \mathbb{R}$ through an implicit latent state $h(t) \in \mathbb{R}^N$. This can be typically formulated as follows:

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t) \\ y(t) &= Ch(t) \end{aligned}$$

Here, $A \in \mathbb{R}^{N \times N}$ denotes the evolution matrix, while $B \in \mathbb{R}^{N \times 1}$ and $C \in \mathbb{R}^{1 \times N}$ denote the input and output mapping matrices, respectively.

Discrete SSM. For integration into deep learning models, SSM requires discretization. Specifically, A , B are transformed into their discretized counterparts \bar{A} , \bar{B} using a timescale parameter $\Delta \in \mathbb{R}$ (Gu, Goel, and Ré 2021). This transformation commonly utilizes the Zero-Order Hold (ZOH) method, defined by:

$$\begin{aligned} \bar{A} &= \exp(\Delta A) \\ \bar{B} &= (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B \end{aligned}$$

This allows the discrete SSM to be represented as:

$$\begin{aligned} h_t &= \bar{A}h_{t-1} + \bar{B}x_t \\ y_t &= Ch_t \end{aligned}$$

Mamba. As evident from the above, the parameters within SSM remain invariant with respect to the input. Mamba (Gu and Dao 2023) identifies this as a fundamental limitation of SSM. In response, Mamba introduces a selection mechanism by setting B , C and Δ as functions of x_t , which allows for propagating or forgetting information throughout the sequence depending on the current token. Additionally, to ensure GPU efficiency, Mamba employs a hardware-aware algorithm within the selective SSM.

Method

This section delineates the core components of our DocMamba, as depicted in Figure 2. Initially, we introduce the Segment-First Bidirectional Scan to enhance DocMamba’s ability to understand tokens in documents. Following this, we introduce the model architecture in detail. The final part illustrates the pre-training of DocMamba, including the training objective and several effective training strategies.

Segment-First Bidirectional Scan

Vanilla Mamba, which is well-suited for the 1-D sequences, captures long-range dependencies by updating the hidden state based on the current token at each step. However, tokens in documents exhibit complex 2-D spatial layouts and share continuous semantic information in conjunction with their neighbors. Therefore, we design the Segment-First Bidirectional Scan (SFBS) to derive 1-D token sequences from documents, as demonstrated in Figure 3.

Specifically, given a document image as illustrated in Figure 3 (a), an off-the-shelf document layout analysis system (Zhong, Tang, and Yepes 2019) is first employed to extract segments such as titles, paragraphs, and captions as depicted in Figure 3 (b). The tokens within each segment are then separately arranged in an order that primarily descends along the Y-axis and then the X-axis. The order of scanning the segments follows a similar pattern. Furthermore, a bidirectional scanning strategy is adopted, as it enables each token in the document to gain global information. The final scanning orders are demonstrated in Figure 3 (c) and (d), where the lighter regions mark the initiation of SFBS, and the darker regions denote its termination.

Model Architecture

DocMamba employs a multi-layer bidirectional Mamba structure as the backbone, taking text and layout information as input. Document images are preprocessed using PaddleOCR¹ to attain the words and corresponding 2-D positions. Detailed descriptions are as follows.

Word Embedding. The text content is tokenized using Byte-Pair Encoding (BPE)(Sennrich, Haddow, and Birch 2015). Each sequence always begins with a specific classification token ([CLS]). Unlike Transformer-based models that necessitate the addition of a 1-D positional embedding to denote word order within a sentence, DocMamba disregards 1-D positional embedding due to the inherent nature of the sequential order within SSMs. Therefore, the i -th word embedding can be formulated as:

$$t_i = \text{TokenEmb}(w_i)$$

2-D Position Embedding. Given the significant influence of a word’s spatial location within a document on its semantic representation, 2-D positional embedding is employed to model these relative spatial positions. Following standard practice (Zhang et al. 2022; Huang et al. 2022; Hu et al. 2022), a document page is considered a coordinate system originating at the top-left. All coordinates are normalized and discretized to integers within the range [0, 1000]. The normalized coordinate of the i -th text token’s four vertices is denoted as $\text{poly}_i = (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, proceeding clockwise from the upper left corner. For the t -th element in poly_i , its embedding can be obtained by:

$$e_{i,t} = \text{PosEmb2D}_{xy}(\text{poly}_{i,t}) + \text{CoordTypeEmb}(t)$$

where PosEmb2D_{xy} is shared between X-axis and Y-axis, and CoordTypeEmb represents the type embedding associated with each coordinate in poly_i . The i -th 2-D position

embedding is the concatenation of $e_{i,1} \sim e_{i,8}$:

$$l_i = \text{Concat}[e_{i,t}], t = 1, \dots, 8$$

Bidirectional Mamba Encoder. The input embeddings $S^0 = \{s_1^0, s_2^0 \dots s_N^0\}$ are computed by summing the word and 2-D position embeddings:

$$s_i^0 = t_i + l_i$$

These input embeddings are then processed through multi-layer bidirectional Mamba blocks. Specifically, the output from the previous layer, S^{m-1} , is fed into the m -th layer, getting the output S^m with a residual connection:

$$S^m = \text{BiMambaBlock}(S^{m-1}) + S^{m-1}$$

BiMambaBlock denotes the bidirectional Mamba block as illustrated on the right part of Figure 2. For the m -th layer, the input S^{m-1} is first normalized and linearly projected to X and Z . X is subsequently processed in both forward and backward directions. In the forward process, X passes through a 1-D convolution layer followed by an activation function to produce X_f . X_f is then linearly projected to generate the B_f , C_f , and Δ_f . These components, along with X_f , are fed into the SSM to compute the discrete \bar{A}_f and \bar{B}_f , leading to the SSM’s output Y_f . The backward output, Y_b , is similarly produced by reversing X from $[x_1; x_2; \dots; x_N]$ to $[x_N; x_{N-1}; \dots; x_1]$. The parameters for the forward and backward directions are not shared. Finally, Y_f and Y_b are gated by Z and summed to produce the output of the current block through a linear layer.

Pre-training Strategy

Following standard procedure (Xu et al. 2020a; Zhang et al. 2022), we employ Masked Language Modeling (MLM) as the pre-training task. This task enables the learning of language representation incorporating layout embedding cues. In the pre-training phase, each token is independently and randomly masked with a given probability P_{mask} , while the associated layout information remains intact. Masked tokens are replaced with a special symbol [MASK]. The output representations of the masked tokens from the encoder are fed into a classifier over the entire vocabulary.

Contrary to prior Transformer-based models that maintain a constant batch size and input length during pre-training, DocMamba is capable of dynamically adjusting the batch size based on the input length. Specifically, we allocate the sequences into non-overlapping buckets based on their lengths, with each bucket covering a range of 64. Within each bucket, input sequences are truncated to the same size. Given the input sequence of length l , we assign the batch size b through $b = k/l$, where k is a constant. This formula is effective because of the linear GPU memory consumption of DocMamba. This approach enhances the efficiency of the pre-training process and empowers the model to dynamically handle document contents of varying lengths.

Experiments

Datasets

We select several datasets to evaluate the performance of DocMamba, including FUNSD (Jaume, Ekenel, and Thiran

¹<https://github.com/PaddlePaddle/PaddleOCR>

Model	Parameters	Modality	FUNSD (F1 \uparrow)	CORD (F1 \uparrow)	SROIE (F1 \uparrow)
BERT (Devlin et al. 2018)	110M	T	60.3	89.7	91.0
RoBERTa (Liu et al. 2019)	125M	T	66.5	93.5	-
Mamba (Gu and Dao 2023)	130M	T	47.5	74.3	77.0
Mamba* (Gu and Dao 2023)	130M	T	58.3	85.3	83.2
LayoutLM (Xu et al. 2020a)	160M	T+L	79.3	-	94.4
BROS (Hong et al. 2022)	110M	T+L	83.1	95.7	95.5
LiLT (Wang, Jin, and Ding 2022)	-	T+L	88.4	96.1	-
SelfDoc (Li et al. 2021)	-	T+L+I	83.4	-	-
UDoc (Gu et al. 2021b)	272M	T+L+I	87.9	98.9 \dagger	-
TILT (Powalski et al. 2021)	230M	T+L+I	-	95.1	97.7 \ddagger
DocFormer (Appalaraju et al. 2021)	183M	T+L+I	83.3	96.3	-
XYLayoutLM (Gu et al. 2022b)	-	T+L+I	83.4	-	-
LayoutLMv2 (Xu et al. 2020b)	200M	T+L+I	82.8	95.0	<u>96.3</u>
LayoutLMv3 (Huang et al. 2022)	133M	T+L+I	<u>90.3</u>	<u>96.6</u>	-
DocMamba	135M	T+L	91.7	97.0	96.8

Table 1: Comparison with existing methods. “T/L/I” stands for “text/layout/image” modality. \dagger : UDoc split the CORD into 626/247 receipts for training/test, deviating from the official 800/100 split for training/test, so the score is not directly comparable. \ddagger : TILT employed extra supervised data for pre-training, making its score not directly comparable as well. *: To keep a fair comparison with DocMamba, we use the same data as DocMamba to pretrain the vanilla Mamba from scratch.

2019), CORD (Park et al. 2019), SROIE (Huang et al. 2019) and HRDoc (Ma et al. 2023).

FUNSD. The FUNSD dataset is a noisy scanned document dataset for form understanding, containing 149 training samples and 50 testing samples. It defines the entity extraction task aimed at extracting values for predefined keys: “question”, “answer”, “header” or “other”.

CORD. The CORD dataset is used for key information extraction from receipts, comprising 800 training samples, 100 validation samples, and 100 test samples. It includes 30 semantic labels under 4 categories: “company”, “date”, “address”, and “total”.

SROIE. The SROIE dataset is another receipt understanding dataset, consisting of 626 training receipts and 347 test receipts. The task is the same as CORD.

HRDoc. The HRDoc dataset is designed for the hierarchical reconstruction of academic document structures. We use the HRDoc-Hard subset, which includes 1,000 training documents and 500 testing documents. Our focus is on semantic unit classification, aiming to categorize each unit into one of 14 categories: “title”, “author”, “mail”, “affiliation”, “section”, “first-line”, “para-line”, “equation”, “table”, “figure”, “caption”, “page-footer”, “page-header”, and “footnote”. HRDoc contains text-dense documents, and we use it to validate DocMamba’s potential for length extrapolation.

Implementation Details

DocMamba employs a 24-layer bidirectional Mamba encoder with a hidden size of 768 and an intermediate size of 1,536. For the SSM within each layer, we use the default hyperparameters from Mamba (Gu and Dao 2023), setting the state dimension to 16. The coordinates of [CLS] are zeros.

Pre-training. We use 10 million pages from the IIT-CDIP Test Collection 1.0 (Lewis et al. 2006), a large-scale scanned document image dataset, to pre-train DocMamba. The con-

stant k for computing the varying batch size of a single GPU is 20,480. For example, the batch size is set to 40 for an input length of 512. For the MLM task, following the settings in BERT (Devlin et al. 2018), we randomly mask 15% of all input tokens. Out of these, 80% are replaced by [MASK], 10% are replaced by random tokens from the vocabulary, and 10% remain unchanged. We adopt distributed training and mixed-precision training to reduce memory costs and speed up training procedures. DocMamba is pre-trained using the Adam optimizer (Kingma and Ba 2014) with a learning rate of 5×10^{-5} for 500,000 steps. The learning rate is warmed up over the first 10% steps and then linearly decayed. Pre-training is conducted on 8 Tesla A40 48GB GPUs.

Finu-tuning. We treat FUNSD, CORD, and SROIE as sequential labeling tasks, using BIO tags for each entity field. We use the officially-provided images and OCR annotations and build a dropout layer and a linear layer above the output representations. DocMamba is fine-tuned on these datasets for 1,000 steps with a learning rate 2×10^{-5} and a batch size of 16. For HRDoc, we directly predict the categories for each unit, using a learning rate of 2×10^{-5} , a batch size of 48 for 2,000 steps.

Comparison With State-of-the-Art Methods

Comparison of F1 scores. Table 1 illustrates the performance of various methods in form and receipt understanding. These methods can be categorized by the modalities used in pre-training. “T” represents pure text models like BERT (Devlin et al. 2018) and RoBERTa (Liu et al. 2019). “T+L” means text and layout models such as LayoutLM (Xu et al. 2020a) and BROS (Hong et al. 2022). “T+L+I” denotes models that incorporate text, layout, and image modalities, including LayoutLMv2 (Xu et al. 2020b), SelfDoc (Li et al. 2021), and LayoutLMv3 (Huang et al. 2022). Some methods offer different versions, such as base and large,

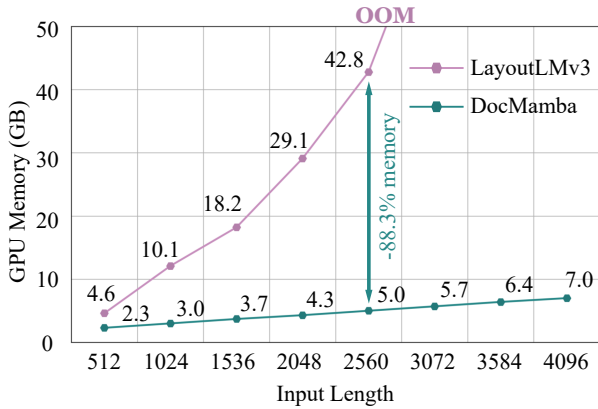


Figure 4: Comparison of GPU memory usage between LayoutLMv3 (Huang et al. 2022) and DocMamba.

due to variations in parameter sizes. To ensure a fair comparison, we opt for the base versions of previous methods, as they maintain a similar number of parameters to that of DocMamba. The entity-level F1 score serves as our evaluation metric. Despite the absence of an image modality in DocMamba, it still outperforms all other methods, including the “T+L+I” models across all three datasets (FUNSD by + 1.4%, CORD by + 0.4%, SROIE by + 0.5%). These results attest to DocMamba’s competitive performance against Transformer-based models, underscoring the substantial potential of SSMs in VrDU.

Comparison of Speed and Memory Usage. Among earlier Transformer-based methods, LayoutLMv3 stands out for its impressive performance and unified structure, making it our primary baseline method. To contrast the speed and memory consumption of DocMamba and LayoutLMv3, we choose HRDoc as the evaluation dataset for semantic unit classification. We use the official implementation of LayoutLMv3 available on Hugging Face² for benchmarking. Figure 4 illustrates the memory consumption of both models during inference with the batch size set to 16. The memory consumption of LayoutLMv3 escalates rapidly, resulting in an Out-of-Memory situation when the input length reaches 3,072. Conversely, DocMamba’s memory consumption grows in a linear manner with the input length, saving 88.3% of memory when the input length attains 2,560. Figure 5 displays the inference speed of both models during inference. Batch size is set to 8 to avoid Out-of-Memory caused by LayoutLMv3. As the input length increases, the Frames Per Second (FPS) of LayoutLMv3 declines sharply. When the input length reaches 4096, DocMamba’s FPS becomes 2.4 times higher than that of LayoutLMv3. These results affirm the efficiency of DocMamba in processing text-dense documents, and also validate DocMamba’s linear computational complexity.

Comparison of Length Extrapolation. Transformers lack an inherent mechanism to consider the order of tokens in a sequence. To address this, many Transformer-based methods in VrDU, such as LayoutLMv3, utilize a learned 1-D position embedding with a prefixed length, which leaves them

²<https://huggingface.co/microsoft/layoutlmv3-base>

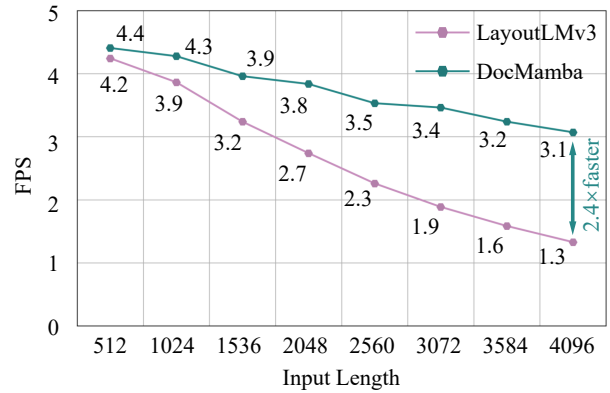


Figure 5: Comparison of Frames Per Second (FPS) between LayoutLMv3 (Huang et al. 2022) and DocMamba.

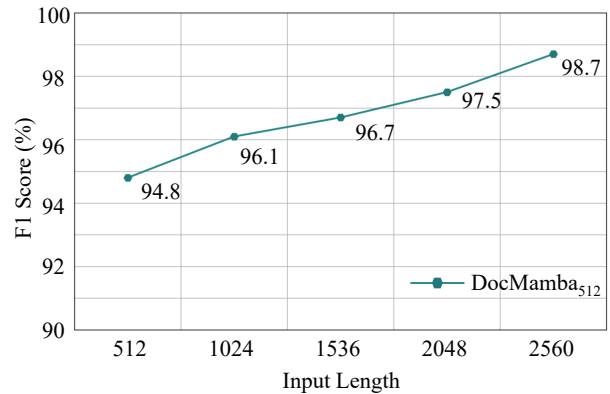


Figure 6: F1 scores of DocMamba₅₁₂ on HrDoc (Ma et al. 2023) with varying input lengths.

incapable of length extrapolation. In contrast, SSMs naturally capture sequential and temporal dependencies without a 1-D position embedding requirement, thus endowing DocMamba with length extrapolation potential. We test this feature through the task of semantic unit classification on the HRDoc dataset. We divide document pages based on their length, and select 5 non-overlapping sub-datasets, each spanning a length range of 512. During both pre-training and fine-tuning, we restrict the input length to 512 to obtain the model, DocMamba₅₁₂. The results are illustrated in Figure 6. As the input length increases, the F1 score of DocMamba also sees an upward trend, since the models can leverage longer contexts to yield more precise predictions. This confirms the potential of DocMamba for length extrapolation.

Ablation Study

Impact of Segment-First Bidirectional Scan. Tokens within documents exhibit complex 2-D spatial layouts. Consequently, we introduce the Segment-First Bidirectional Scan (SFBS) to convert these layouts into 1-D token sequences prior to inputting them into the SSM. To validate the effectiveness of SFBS, we contrast it with the Word-First Bidirectional Scan (WFBS) on FUNSD. Specifically, WFBS

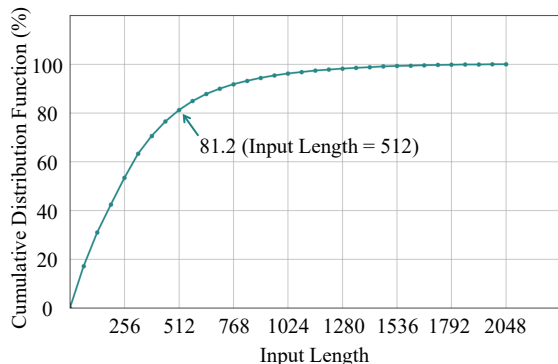


Figure 7: The cumulative distribution function of input lengths of DocMamba during pre-training.

Scan Strategy	Granularity	FUNSD
SFBS	Segment	91.7
WFBS	Word	80.8

Table 2: Ablation study of the Segment-First Bidirectional Scan (SFBS) and Word-First Bidirectional Scan (WFBS).

utilizes word-level granularity, and organizes tokens directly based on their own Y-axis and X-axis. The order of scanning follows a similar pattern to SFBS. The comparative results are shown in Table 2. It is clearly evident that the performance of WFBS significantly lags behind SFBS. This can be attributed to SFBS disrupting the sequence of tokens in forms, thereby inhibiting their ability to generate a continuous semantic flow.

Impact of Input Length in Pre-training. As introduced in the earlier sections, different from previous Transformer-based methods using a fixed pre-training input length, DocMamba employs a variable input length during pre-training. Figure 7 showcases the cumulative distribution function of input lengths during pre-training, ranging from 64 to 2,048. To investigate the effect of varying input length, following LayoutLMv3, we limit the input length during pre-training to a maximum of 512 while keeping other settings the same, leading to a new model, DocMamba₅₁₂. The results are presented in Table 3. We can make two observations: (1) Increasing the input length is beneficial, as the performance of DocMamba₅₁₂ on the FUNSD and CORD datasets falls short by 0.9 and 0.4 points respectively. (2) Even when the pre-training input length is confined to a maximum of 512, DocMamba₅₁₂ still surpasses LayoutLMv3.

Impact of Number of Layers. In VrDU, the popularity of Transformer-based models is partially due to their ability to deepen the network by stacking additional layers, facilitating more comprehensive feature learning. Thus, we also explore DocMamba’s scalability by adjusting the encoder’s layer count to 12, 18, 24, and 30. For experimental efficiency, all models are pre-trained for 10 epochs using the MLM task. The results are presented in Figure 8. A steady increase in the number of parameters can be observed with

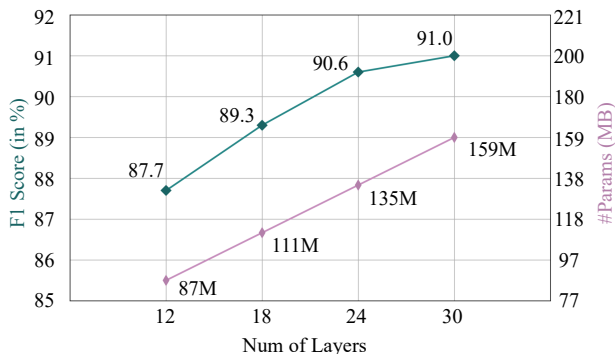


Figure 8: F1 scores on FUNSD and parameter counts across different layer numbers.

Model	FUNSD	CORD	SROIE
LayoutLMv3	90.3	96.6	-
DocMamba ₅₁₂	90.8	96.6	96.8
DocMamba	91.7	97.0	96.8

Table 3: Ablation study of the varying input length. The input length of DocMamba₅₁₂ is limited to 512.

the rise in layer counts. In addition, DocMamba’s F1 score also exhibits a progressive climb, verifying its scalability. This result aligns well with the findings of Mamba in other fields (Zhu et al. 2024; Li et al. 2024).

Limitation

DocMamba’s central limitation is its omission of image modality. This decision stems from the observation that DocMamba, employing only text and layout, could already outperform Transformer-based models that incorporate text, layout, and image modalities. This is sufficient to demonstrate the competitive potential of SSM against the Transformer in VrDU. We leave the incorporation of image modality in SSM-based methods to future research in VrDU.

Conclusion

In this study, we propose DocMamba, a model based on the SSM that does not rely on the self-attention mechanism. This reduces computational complexity to linear, making it suitable for processing text-dense documents. We also introduce Segment-First Bidirectional Scan, which is used to extract 1-D token sequences from documents. In addition, DocMamba combines text and layout information using a multi-layer bidirectional Mamba encoder. Experiments conducted on publicly available datasets, including FUNSD, CORD, and SROIE, show that DocMamba outperforms previous Transformer-based models, with faster speed and less memory usage. Further, outcomes on HRDoc validate DocMamba’s capacity for length extrapolation. This study highlights the potential of SSM as a powerful tool for understanding visually-rich documents and provides a simple yet effective baseline for future research.

References

- Appalaraju, S.; Jasani, B.; Kota, B. U.; Xie, Y.; and Manmatha, R. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, 993–1003.
- Cui, L.; Xu, Y.; Lv, T.; and Wei, F. 2021. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*.
- Dat, D. H.; Anh, D. D.; Luu, A. T.; and Buntine, W. 2024. Discrete Diffusion Language Model for Long Text Summarization. *arXiv preprint arXiv:2407.10998*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Gu, A.; Goel, K.; Gupta, A.; and Ré, C. 2022a. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35: 35971–35983.
- Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.
- Gu, A.; Johnson, I.; Goel, K.; Saab, K.; Dao, T.; Rudra, A.; and Ré, C. 2021a. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34: 572–585.
- Gu, J.; Kuen, J.; Morariu, V. I.; Zhao, H.; Jain, R.; Barmaliotis, N.; Nenkova, A.; and Sun, T. 2021b. Unidoc: Unified pretraining framework for document understanding. *Advances in Neural Information Processing Systems*, 34: 39–50.
- Gu, Z.; Meng, C.; Wang, K.; Lan, J.; Wang, W.; Gu, M.; and Zhang, L. 2022b. Xylayoutlm: Towards layout-aware multimodal networks for visually-rich document understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4583–4592.
- Hong, T.; Kim, D.; Ji, M.; Hwang, W.; Nam, D.; and Park, S. 2022. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10767–10775.
- Hu, P.; Ma, J.; Zhang, Z.; Du, J.; and Zhang, J. 2024. Count, decompose and correct: A new approach to handwritten Chinese character error correction. *Pattern Recognition*, 111110.
- Hu, P.; Zhang, Z.; Zhang, J.; Du, J.; and Wu, J. 2022. Multimodal tree decoder for table of contents extraction in document images. In *2022 26th international conference on pattern recognition (ICPR)*, 1756–1762. IEEE.
- Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, 4083–4091.
- Huang, Z.; Chen, K.; He, J.; Bai, X.; Karatzas, D.; Lu, S.; and Jawahar, C. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1516–1520. IEEE.
- Jaume, G.; Ekenel, H. K.; and Thiran, J.-P. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, 1–6. IEEE.
- Kalman, R. E. 1960. A new approach to linear filtering and prediction problems.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lewis, D.; Agam, G.; Argamon, S.; Frieder, O.; Grossman, D.; and Heard, J. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 665–666.
- Li, K.; Li, X.; Wang, Y.; He, Y.; Wang, Y.; Wang, L.; and Qiao, Y. 2024. VideoMamba: State Space Model for Efficient Video Understanding. *arXiv:2403.06977*.
- Li, P.; Gu, J.; Kuen, J.; Morariu, V. I.; Zhao, H.; Jain, R.; Manjunatha, V.; and Liu, H. 2021. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5652–5660.
- Lieber, O.; Lenz, B.; Bata, H.; Cohen, G.; Osin, J.; Dalmedigos, I.; Safahi, E.; Meirum, S.; Belinkov, Y.; Shalev-Shwartz, S.; et al. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Lu, H.; Salah, A. A.; and Poppe, R. 2024. VideoMambaPro: A Leap Forward for Mamba in Video Understanding. *arXiv preprint arXiv:2406.19006*.
- Luo, C.; Cheng, C.; Zheng, Q.; and Yao, C. 2023. Geolayoutlm: Geometric pre-training for visual information extraction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7092–7101.
- Ma, J.; Du, J.; Hu, P.; Zhang, Z.; Zhang, J.; Zhu, H.; and Liu, C. 2023. Hrdoc: Dataset and baseline method toward hierarchical reconstruction of document structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1870–1877.
- Park, S.; Shin, S.; Lee, B.; Lee, J.; Surh, J.; Seo, M.; and Lee, H. 2019. CORD: a consolidated receipt dataset for post-OCR parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.
- Powalski, R.; Borchmann, Ł.; Jurkiewicz, D.; Dwojak, T.; Pietruszka, M.; and Pałka, G. 2021. Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer. In *ICDAR*.

- Raibert, M. H. 1977. *Motor control and learning by the state space model*. Ph.D. thesis, Massachusetts Institute of Technology.
- Rao, B. D.; and Arun, K. 1992. Model based processing of signals: A state space approach. *Proceedings of the IEEE*, 80(2): 283–309.
- Schulz, R.; and Werwatz, A. 2004. A state space model for Berlin house prices: Estimation and economic interpretation. *The Journal of Real Estate Finance and Economics*, 28: 37–57.
- Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Smith, J. T.; Warrington, A.; and Linderman, S. 2023. Simplified State Space Layers for Sequence Modeling. In *The Eleventh International Conference on Learning Representations*.
- Tu, Y.; Guo, Y.; Chen, H.; and Tang, J. 2023. Layoutmask: Enhance text-layout interaction in multi-modal pre-training for document understanding. *arXiv preprint arXiv:2305.18721*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, J.; Jin, L.; and Ding, K. 2022. Lilt: A simple yet effective language-independent layout transformer for structured document understanding. *arXiv preprint arXiv:2202.13669*.
- Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020a. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 1192–1200.
- Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; et al. 2020b. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.
- Yang, X.; Yumer, E.; Asente, P.; Kraley, M.; Kifer, D.; and Lee Giles, C. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5315–5324.
- Yao, J.; Lai, Y.; Kou, H.; Wu, T.; and Liu, R. 2024. QE-BEV: Query evolution for bird’s eye view object detection in varied contexts. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2927–2935.
- Yao, J.; Li, C.; and Xiao, C. 2024. Swift Sampler: Efficient Learning of Sampler by 10 Parameters. *arXiv preprint arXiv:2410.05578*.
- Zhang, X.; Zhang, Q.; Liu, H.; Xiao, T.; Qian, X.; Ahmed, B.; Ambikairajah, E.; Li, H.; and Epps, J. 2024a. Mamba in Speech: Towards an Alternative to Self-Attention. *arXiv preprint arXiv:2405.12609*.
- Zhang, Z.; Hu, P.; Ma, J.; Du, J.; Zhang, J.; Yin, B.; Yin, B.; and Liu, C. 2024b. SEMv2: Table separation line detection based on instance segmentation. *Pattern Recognition*, 149: 110279.
- Zhang, Z.; Liu, S.; Hu, P.; Ma, J.; Du, J.; Zhang, J.; and Hu, Y. 2024c. UniTabNet: Bridging Vision and Language Models for Enhanced Table Structure Recognition. *arXiv preprint arXiv:2409.13148*.
- Zhang, Z.; Ma, J.; Du, J.; Wang, L.; and Zhang, J. 2022. Multimodal pre-training based on graph attention network for document understanding. *IEEE Transactions on Multimedia*, 25: 6743–6755.
- Zhong, X.; Tang, J.; and Yepes, A. J. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International conference on document analysis and recognition (ICDAR)*, 1015–1022. IEEE.
- Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.