Paphos, Cyprus

School Of Economics, Business And Computer Science

Department of Computer Science

Adapting S4 for high-dimensional data

Artem Makoian

Thesis advisor: Professor Zach Anthis, Alexander Avdiushenko

May 2024

Abstract

The advancement of machine learning models for sequence data has seen significant progress with the development of the Structured State Space Sequence (S4) architecture [6]. Initially designed for natural language processing (NLP) tasks, S4 has demonstrated superior performance over traditional transformers [22], particularly in handling long-context dependencies. While transformers have successfully migrated from NLP to computer vision, becoming the state-of-the-art (SoTA) in various tasks, they still face challenges with long-context information. This thesis explores the potential of S4 in the realm of computer vision by adapting it to high-dimensional data. Our primary focus is on evaluating S4ND in computer vision tasks involving high-resolution images and long-context information. We tested S4ND [14] on the ChesapeakeRSC [16] dataset for aerial segmentation, which presents a complex challenge due to the high resolution and vast spatial context of the images. Additionally, we extended our evaluation to various video tasks, assessing S4ND's capability in capturing temporal dependencies across frames (LVU [25], Breakfast [10], COIN [20]). The experimental results reveal that S4ND outperforms existing models in handling high-dimensional data with long-context dependencies. This suggests that S4ND has the potential to redefine state-of-the-art performance in computer vision tasks, particularly where long-context information is critical. Our findings pave the way for further research and optimization of S4-based models in high-dimensional, long-context applications across different domains. Moreover, we proposed ViS4NDmer architecture based on ViS4mer [9] and S4ND that outperforms previous models on video long-context benchmarks. The code can be found here: https://github.com/MakArtKar/s4nd-experiments.

Contents

Introduction

Motivation

The field of machine learning has seen remarkable advancements with the development of various architectures designed to handle sequential data. The Structured State Space Sequence (S4) model [6] is one of the latest innovations in this domain, offering significant improvements over traditional models, especially in tasks requiring the handling of long-context dependencies. Initially, S4 was introduced for natural language processing (NLP) tasks, where it has demonstrated exceptional performance, surpassing even the highly-regarded transformer models in scenarios involving extensive context.

Transformers, originally developed for NLP, have successfully migrated to the field of computer vision, becoming the state-of-the-art (SoTA) solution for numerous vision tasks. Despite their success, transformers still encounter challenges when dealing with long-context information, such as long-range dependencies in sequences. This limitation presents an opportunity to explore the application of the S4 architecture in computer vision, particularly for tasks that involve high-dimensional data and require the capture of long-context dependencies.

Problem Statement

Given the success of the S4 model in NLP tasks and the limitations of transformers in handling long-context information in computer vision, this thesis aims to investigate the adaptation of the S4 architecture to the domain of computer vision. Specifically, the research focuses on the development and evaluation of the S4ND [14] (N-dimensional S4) architecture, designed to manage high-dimensional data and capture long-context dependencies effectively.

Objectives

1 To evaluate S4ND in computer vision tasks, particularly those involving high-resolution images and long-context dependencies.

2 To test the performance of S4ND on the ChesapeakeRSC dataset for aerial segmentation, a complex task due to the high resolution and extensive spatial context.

3 To extend the evaluation of S4ND to various video tasks, assessing its capability to handle temporal dependencies across frames.

4 To compare the performance of S4ND with existing state-of-the-art models, particularly transformers, and analyze its advantages and limitations.

5 To propose a novel architecture for long-range video tasks based on S4ND.

Contribution

This thesis makes the following contributions to the field of machine learning and computer vision:

1 A comprehensive evaluation of S4ND in computer vision tasks involving high-resolution images and long videos.

2 Empirical evidence demonstrating the superior performance of S4ND over existing models, particularly in scenarios requiring the handling of long-context information.

3 ViS4NDmer - a novel architecture for processing long-range videos effectively leveraged spatial and temporal dependencies.

1 Literature Review

1.1 Models

1.1.1 RNN

Recurrent Neural Networks (RNNs) [17] were designed to process the sequential data for NLP tasks. The main advantage compared to Convolutional Neural Networks (CNNs) is that they can remember the information from previous tokens longer than CNN with it local context.

Two most popular versions of blocks in RNN are LSTM [8] and GRU [3]. These blocks help to prevent vanishing gradient problem which prevents loosing long-term dependencies information. They have the gating mechanism that updates the hidden state with knowledge of the input token.

In an LSTM, there are three gates: the input gate, the forget gate, and the output gate. The input gate controls how much new information is added to the hidden state, the forget gate controls how much of the previous hidden state is retained, and the output gate controls how much of the current hidden state is used to make a prediction. The gates are controlled by sigmoid functions, which output values between 0 and 1 that determine the degree of gate opening or closing.

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \\
o_t &= \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \\
\tilde{C}_t &= \tanh(W_C \cdot x_t + U_C \cdot h_{t-1} + b_C) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
h_t &= o_t \odot \tanh(C_t)
\end{aligned}
$$

In a GRU, there are two gates: the update gate and the reset gate. The update gate controls how much of the previous hidden state is retained, and the reset gate controls how much of the current input is incorporated into the hidden state. The gates are controlled by a single sigmoid function and a reset gate function, respectively.

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z)$$
$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r)$$
$$\tilde{h}_t = \tanh(W \cdot x_t + U \cdot (r_t \odot h_{t-1}) + b)$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

Both LSTM and GRU have been shown to perform well on a variety of sequence prediction tasks. LSTM is more effective for longer sequences and GRU is faster to train and requiring fewer parameters.

1.1.2 Transformers

Attention

Transformer [22] is an architecture, which relies entirely on self-attention mechanisms to draw global dependencies between input and output. This approach eliminates the need for recurrent layers, which are typically used in sequence-to-sequence models. The Transformer architecture consists of an encoder and a decoder, each composed of multiple identical layers. Each layer in the encoder has two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. The decoder has an additional sub-layer for multi-head attention over the encoder's output.

The key innovation is the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence when forming representations, regardless of their positions. Positional encodings are added to the input embeddings to preserve the order of the sequence. The multi-head attention mechanism enhances the model's ability to focus on different parts of the input for each word, providing more nuanced understanding and better capturing context.

Overall, the Transformer architecture simplifies the model structure, improves parallelization, and achieves superior performance on tasks such as machine translation. This model has set new benchmarks in natural language processing and has become a foundation for many subsequent advancements in the field.

ViT

ViT [24] is a model for visual tasks based on the transformer model, originally designed

for natural language processing. Instead of processing the entire image at once, the ViT model divides the image into fixed-size patches, treats each patch as a token, and then processes these tokens through a standard transformer architecture.

The process begins by dividing an image into smaller patches, flattening them, and embedding them into vectors. Positional embeddings are then added to these vectors to retain spatial information. This sequence of vectors is fed into a transformer model, which uses self-attention mechanisms to analyze the relationships between the patches. A special classification token is appended to the sequence, and its output is used for the final image classification after being processed through a multi-layer perceptron.

The ViT approach leverages the transformer's ability to capture global context and dependencies within the image, leading to competitive performance on various image recognition benchmarks. This method demonstrates that transformers can be effectively adapted for vision tasks, offering a new way to approach image classification problems.

1.1.3 SSM

Gu et al. introduced the State-Space Model (SSM) [7] a novel approach to enhance deep learning models based on State-Space equations by integrating different types of neural network architectures. The approach leverages the strengths of recurrent neural networks (RNNs), convolutional neural networks (CNNs), and continuous-time models by incorporating Linear State-Space (LSS) layers.

The LSS layers are designed to handle both temporal and spatial data effectively, bridging the gap between the capabilities of RNNs, which are good at handling sequential data, and CNNs, which excel at processing spatial data. By integrating these layers, the model can process data continuously over time, which is beneficial for tasks that require understanding dynamic and evolving data patterns.

This combined architecture allows the model to capture complex dependencies and interactions within the data, providing a more comprehensive understanding of temporal and spatial features. The paper demonstrates that this approach not only improves the performance of various tasks but also offers greater flexibility and efficiency in handling different types of data, reducing the limitations associated with using RNNs, CNNs, or continuous-time models in isolation.

1.1.4 S4

S4 [6] is a new architecture based on SSMs for handling long sequences in data efficiently. The core of the approach is the use of structured state spaces, which combines elements from both classical state-space models and modern neural network techniques. This method is designed to capture long-range dependencies in sequences more effectively than traditional models.

By structuring the state spaces in a specific way, the model can maintain and process information over long sequences without the typical exponential growth in computational complexity. This results in a more scalable and efficient approach to modeling long sequences, addressing the challenges posed by long-term dependencies and large-scale data. The paper demonstrates that this structured state-space model outperforms existing methods in various tasks, particularly those involving lengthy sequential data, by providing better performance and greater computational efficiency.

1.1.5 S4ND

S4ND [14] is a new model for processing images and videos by treating them as multi-dimensional signals and applying state space models to understand their underlying structures. The approach aims to address the complexity and high dimensionality inherent in visual data by leveraging the mathematical foundations of state space models, which have been traditionally used in fields like control theory and signal processing.

The framework begins with the recognition that images and videos can be viewed as signals that vary across multiple dimensions—spatial dimensions in the case of images and additional temporal dimensions in the case of videos. This perspective allows the authors to apply state space models, which are adept at handling such multidimensional data, to capture the intricate patterns and dependencies present in visual content.

For images, the model considers the spatial dimensions (height and width) as state variables. The state space model is designed to capture the spatial dependencies and variations within the image, treating it as a signal that evolves across these dimensions. This involves representing the image as a set of states, where each state corresponds to a specific region or pixel of the image. The transitions between these states are governed by the spatial relationships and patterns observed in the image.

For videos, the model extends this concept to include the temporal dimension, treating the video as a signal that evolves not only across spatial dimensions but also over time. This results in a more complex state space model that captures both spatial and temporal dependencies. Each

frame of the video is represented as a state, and the transitions between states account for changes both within frames (spatial transitions) and between frames (temporal transitions).

The training process for the state space models involves learning the parameters that define these transitions. This is done by optimizing the model to accurately represent the observed data, which includes images or sequences of video frames. The training data consists of annotated images and videos that provide the necessary ground truth for the model to learn from.

One of the key strengths of this approach is its ability to capture long-range dependencies and contextual information, which are critical for understanding complex visual content. By treating images and videos as multidimensional signals, the state space model can effectively manage the high dimensionality and extract meaningful patterns that might be missed by traditional models.

The authors demonstrate the effectiveness of their framework through a series of experiments on image and video datasets. The results show that the S4ND model outperforms existing methods in various tasks, such as image classification, object detection, and video analysis. This success is attributed to the model's ability to leverage the multidimensional nature of visual data and the robust mathematical foundation provided by state space models.

Overall, the paper presents a comprehensive approach to modeling images and videos using state spaces, offering a novel perspective on visual data processing that combines the strengths of signal processing and modern machine learning techniques. This approach opens up new possibilities for accurately and efficiently understanding and analyzing complex visual content.

1.1.6 ViS4mer

ViS4mer [9] is a novel model to classifying long movie clips by leveraging state-space models. The primary challenge addressed in this work is the complexity and variability inherent in long movie clips, which can span several minutes and contain diverse scenes, characters, and narrative structures.

The approach begins with the need to represent the movie clips in a way that captures their temporal dynamics and contextual information effectively. To achieve this, the authors propose the use of state-space models, which are mathematical models that describe a system by a set of variables and their transitions over time. In this context, the system is the sequence of frames in a movie clip, and the state-space model captures the progression and relationships between these frames.

The process starts with the extraction of features from each frame of the movie clip. These

features include visual information, such as objects and actions, as well as audio cues and textual elements like subtitles or dialogue. By combining these multimodal features, the model obtains a comprehensive representation of each frame.

Next, the state-space model is employed to capture the temporal dependencies and transitions between frames. This model treats each frame as a state and defines the transitions between states based on the extracted features. The state-space model is designed to handle the long-range dependencies that are common in movie clips, ensuring that the model can understand the context and flow of the narrative over extended periods.

To train the state-space model, the authors use a method that involves learning the parameters of the model from a large dataset of annotated movie clips. This training process involves optimizing the model to predict the correct labels for the movie clips, which can include genres, themes, or specific content-related categories.

The model's ability to handle long-range dependencies and its integration of multimodal features make it particularly suited for classifying long movie clips. By understanding the relationships and transitions between frames, the state-space model can accurately capture the essence of the movie clip and classify it according to the defined categories.

The paper presents several experiments and results demonstrating the effectiveness of the state-space video model in classifying long movie clips. The model is shown to outperform traditional approaches that do not account for the temporal dynamics and multimodal nature of movie clips. This approach represents a significant advancement in video classification, particularly for long-form content where capturing the narrative flow and context is crucial for accurate classification.

1.1.7 VideoBERT

VideoBERT [19] is a novel approach to integrating video and language data for improved representation learning. The primary goal of the model is to bridge the gap between visual content in videos and associated textual descriptions or dialogue, enabling more effective understanding and generation of multimodal content.

VideoBERT builds upon the BERT (Bidirectional Encoder Representations from Transformers) architecture, which has shown significant success in natural language processing tasks. However, unlike BERT, which focuses solely on text, VideoBERT extends this model to handle both video and language data simultaneously.

The approach begins with preprocessing the video data. Videos are divided into short clips,

and from each clip, key frames are extracted. These frames serve as visual tokens, akin to words in a sentence. Alongside this, any available textual data, such as subtitles or descriptions, are tokenized into word tokens.

Next, the model employs a tokenizer specifically designed for the visual domain, converting frames into a sequence of visual tokens. These tokens, combined with the word tokens from the text, form a unified sequence representing both the video and the associated language content.

To train VideoBERT, the model leverages a masked token prediction task similar to that used in BERT. In this task, a portion of the tokens in the sequence (both visual and textual) are masked, and the model is trained to predict these masked tokens based on the surrounding context. This method allows the model to learn deep contextual relationships between video frames and textual elements, fostering a joint understanding of both modalities.

Additionally, the model incorporates a segment embedding technique to distinguish between visual and textual tokens within the sequence. This distinction helps the model recognize and process the different types of information appropriately.

Through this training process, VideoBERT learns to create rich, joint representations of video and language data. These representations capture the nuanced interactions between the visual and textual components, making them useful for a variety of downstream tasks, such as video captioning, question answering about videos, and cross-modal retrieval.

The paper highlights the effectiveness of VideoBERT through several experiments, demonstrating its ability to outperform baseline models on tasks that require a deep understanding of both video content and associated language. By unifying video and language representation learning, VideoBERT sets a new standard for multimodal models and opens up new possibilities for applications that require integrated video and language processing.

1.1.8 Object Transformer

The paper "Towards Long-Form Video Understanding" [25] focuses on developing methods to better comprehend and analyze long-form videos, which are videos that typically span several minutes to hours. The approach detailed in the paper addresses the challenges associated with processing and understanding such extensive content. Traditional video analysis methods often struggle with the length and complexity of long-form videos due to the sheer volume of data and the intricate narrative structures these videos can possess.

The authors propose a novel framework that breaks down the problem into more manageable components. This framework leverages a combination of temporal segmentation, multimodal

feature extraction, and hierarchical modeling to effectively process and interpret long-form videos.

Temporal segmentation involves dividing the video into coherent segments or scenes, which makes it easier to analyze smaller chunks of data instead of the entire video at once. This segmentation is achieved using advanced techniques that can identify scene boundaries based on changes in visual content, audio cues, and other contextual information.

Multimodal feature extraction is a crucial step in the process, as it involves capturing relevant features from different modalities present in the video, such as visual, audio, and textual information. By integrating features from these various sources, the framework can gain a more holistic understanding of the video's content.

Hierarchical modeling then comes into play to manage the complexity of the video data. This modeling approach organizes the extracted features and segmented scenes into a structured representation that mirrors the video's narrative flow. The hierarchical structure allows the framework to capture both high-level context and detailed information, enabling a deeper understanding of the video.

Furthermore, the paper discusses the use of advanced machine learning techniques, such as transformers and attention mechanisms, to enhance the processing and interpretation of the segmented and multimodal data. These techniques help in capturing long-range dependencies and interactions within the video, which are critical for understanding the overarching narrative and finer details.

The authors also highlight the importance of creating and utilizing large-scale annotated datasets for training and evaluating their models. These datasets provide the necessary ground truth for developing robust algorithms capable of handling the intricacies of long-form video content.

Overall, the approach described in the paper represents a significant step towards more effective and efficient long-form video understanding. By combining temporal segmentation, multimodal feature extraction, and hierarchical modeling, along with advanced machine learning techniques, the framework aims to provide a comprehensive solution to the challenges posed by long-form video analysis.

1.2 Datasets

1.2.1 ChesapeakeRSC

The Chesapeake Region of Spatial Context (ChesapeakeRSC) [16] dataset is a high-resolution aerial imagery dataset designed for various remote sensing tasks. It includes detailed annotations

for land cover classification and change detection. The dataset is notable for its high spatial resolution, covering multiple states in the Chesapeake Bay watershed area. It comprises images collected from multiple sources, including NAIP (National Agriculture Imagery Program) and NLCD (National Land Cover Database), providing rich spatial and temporal context.

The dataset is structured to facilitate research in high-resolution image segmentation and other computer vision tasks that require understanding complex spatial patterns. It includes diverse land cover types, such as urban areas, forests, water bodies, and agricultural lands, making it a valuable resource for training and evaluating machine learning models aimed at environmental monitoring and land use analysis.

ChesapeakeRSC is particularly challenging due to the variability in land cover types and the need for models to capture fine-grained details and long-range spatial dependencies. This makes it an ideal testbed for advanced machine learning architectures, such as the S4ND model, which aims to handle high-dimensional data with long-context dependencies efficiently.

1.2.2 LVU

The paper "Understanding Long-form Movie Videos" [25] presents an approach to analyzing and understanding the complex structure of full-length movies. It introduces methods to capture the narrative flow and intricate details of movies by segmenting them into scenes and detecting key events. The approach involves using deep learning models, including convolutional neural networks (CNNs) and transformers, to extract visual and temporal features from the videos. These models help in identifying characters, actions, and interactions throughout the movie.

A significant contribution of the paper is the introduction of the Long-Form Video Understanding (LVU) Benchmark. The LVU Benchmark is designed to evaluate the effectiveness of models in handling long-form video content. It includes a diverse set of tasks such as action detection, temporal localization, video classification, and understanding character interactions. By providing a comprehensive set of evaluation criteria, the LVU Benchmark aims to push the development of models that can understand the rich and complex narrative structures typical of long-form movies.

1.2.3 COIN

The COIN [20] dataset is a large-scale collection designed for comprehensive analysis of instructional videos. It contains over 11,800 videos spanning 180 different tasks, covering a wide array of activities from various domains such as cooking, DIY projects, and makeup tutorials. Each

video in the COIN dataset is meticulously annotated, providing detailed step-by-step descriptions of the instructional process. These annotations include temporal segments that mark the start and end times of each step, along with corresponding descriptions, making it possible to analyze the sequence and structure of the tasks being performed.

The dataset is structured to facilitate the development and evaluation of models for tasks such as action segmentation, action recognition, and temporal localization. By offering a diverse and extensive set of instructional videos, the COIN dataset aims to advance research in understanding complex, sequential, and multimodal information inherent in instructional content. Its rich annotations and variety of tasks provide a robust foundation for training machine learning models to automatically interpret and assist with instructional videos. This makes the COIN dataset a valuable resource for researchers in computer vision and machine learning, particularly those focused on enhancing the capabilities of AI in understanding and processing instructional material.

1.2.4 Breakfast

The Breakfast dataset [10] is a large-scale collection of videos specifically designed to facilitate research in the area of human activity recognition, particularly in the context of daily activities associated with breakfast preparation. It consists of 1,712 videos capturing various individuals performing a wide range of breakfast-related activities in natural settings. The dataset spans 10 different categories of breakfast activities, including making coffee, frying eggs, making juice, and preparing tea. Each of these activities is recorded from multiple viewpoints, providing a diverse and rich set of video data.

The videos are meticulously annotated with temporal segments that label specific actions within each activity. These annotations include detailed information about the start and end times of each action, allowing researchers to analyze and develop models for temporal action segmentation and recognition. The dataset's comprehensive annotations and varied video content make it a valuable resource for advancing the state-of-the-art in video analysis and understanding complex human activities in real-world scenarios.

The Breakfast dataset is widely used for benchmarking purposes in the field of computer vision and machine learning, particularly for tasks such as action recognition, sequence modeling, and temporal segmentation. Its extensive coverage of everyday activities and detailed annotations provide a robust foundation for training and evaluating machine learning models aimed at understanding and interpreting human actions in video data.

2 Background

2.1 SSM

SSM arcitecture is based on SSM equations

$$\begin{cases} x'(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

They are discretized to work with sequences and transform into

$$\begin{cases} x_t = \overline{A}x_{t-1} + \overline{B}u_t \\ y_t = \overline{C}x_t \end{cases}$$

This equation is similar to RNN architecture but without activation functions. This allows us to represent transitions as convolution $K * u$, where $K = \left(\overline{CB}, \overline{CAB}, \overline{CA}^2\overline{B} \dots\right)$ and count it with Fast Fourier Transform (FFT) [18] effectively.

This raw approach doesn't work properly and can't surpass the RNNs. The reason is that during calculating the convolution kernel we have to put the matrix in the power of sequence length that can be very long. This results into gradient vanishing and explode during training - the same problem as with RNNs. So, without any tricks raw SSMs work bad, even on pMNIST (a variant of the original MNIST dataset where each image undergoes a fixed random permutation of pixels) it achieves 62% [7].

2.2 S4

S4 architecture is an improvement of previous SSM architecture and surpasses transformers on Long Range Arena [21]. The two main drawbacks of SSM architecture are that we randomly initialized SSM which doesn't perform well and that computing the convolution naively like we've is really slow and memory inefficient.

First improvement [5] is to initialize SSM with Hippo matrices to overcome vanishing and exploding problems in gradient descent.

$$A_{nk} = \begin{cases} (2n+1)^{0.5}(2k+1)^{0.5}, & n > k \\ n+1, & n = k \\ 0, & n < k \end{cases}$$

This trick boosts performance from 62% to 98% on MNIST.

And second, the main improvement, is to effectively count the convolution kernel. We can do this if we restrict matrix A making A the sum of orthogonal and low-dimensional matrices. Doing this in naive manner we got $O(N^2L)$ time complexity because of counting A^L in convolution kernel, where $A \in Mat_{N\times N}$. First, we build the generating function (truncated by first L terms) for kernel of convolution instead of directly compute the kernel. And instead of matrix power we get matrix inverse. Then we use a Cauche trick [Pan [15]] and Woodbury identity to replace matrix inversion with weighted sum operation which make the training more stable. And finally paper shows that we can compute the result signal without matrix inversion.

These tricks boost the performance of the architecture and allow it to bit transformers on Long Range Arena.

2.3 S4ND

When transformers surpassed other models in NLP tasks they were adapted to work with images and overcame also CNNs in vision tasks [24]. The same idea was implemented for S4 models but instead ViT approach where spatial dependencies are obtained ineffectively authors proposed S4ND [14] architecture that can work with N-dimensional data effectively use spatial context.

Adapting to Multidimensional Data

In the paper the SSM equations were adapted for K-dimensional data case (Multidimensional SSM). Let $t = (t^{(1)}, \dots, t^{(K)})$ be K-dimensional time-invariant, $u(t)$ and $y(t)$ be the input and output signals $\mathbb{R}^K \to \mathbb{C}$ and $x(t) = \left(x^{(1)}(t), \dots, x^{(N)}(t)\right)$ - the SSM state of dimension $N^{(1)} \times \dots \times N^{(K)}$, where $x^{(\tau)} : \mathbb{R}^K \to \mathbb{C}^{N^{(\tau)}}$.

Multidimensional SSM: given parameters $A^{(\tau)} \in \mathbb{C}^{N^{(\tau)}\times N^{(\tau)}}$, $B^{(\tau)} \in \mathbb{C}^{N^{(\tau)}\times 1}$, $C \in \mathbb{C}^{N^{(1)}\times\dots\times N^{(K)}}$

$$\frac{\partial}{\partial t^{(i)}} x(t) = \left(x^{(1)}(t), \dots, x^{(i-1)}(t), A^{(i)}x^{(i)}(t), x^{(i+1)}(t), \dots, x^{(N)}(t)\right) + B^{(i)}u(t)$$

$$y(t) = \langle C, x(t)\rangle$$

Then authors show that previous system is equivalent to a K-dimensional convolution $y = K * u$ by the kernel

$$K(t) = \langle C, \otimes_{i=1}^{K} \exp(t^{(i)}A^{(i)})B^{(i)}\rangle$$

After that authors mentioned that the number of basis functions $N^{(1)} \times \dots \times N^{(K)}$ grows

exponentially in the dimension, increasing the parameter count of C. They also factorized the kernel into $N^{(1)} \times \ldots \times N^{(K)}$ additives $\{K^{(1)}_{n^{(1)}}(t^{(1)}) \otimes \ldots \otimes K^{(K)}_{n^{(K)}}(t^{(K)}) : n^{(i)} \in [N^{(i)}] \forall i = 1 \ldots K\}$.

Lastly, authors showed that

$$K(t) = \sum_{i=1}^{r} K_i^{(1)}(t^{(1)}) \otimes \ldots \otimes K_i^{(K)}(t^{(K)})$$

$$K_i^{(j)} = C_i^{(j)} \exp(t^{(j)} A^{(j)}) B^{(j)}$$

$$C = \sum_{i=1}^{r} \otimes_{j=1}^{K} C_i^{(j)}$$

In experiments authors chose $r = 1$ taking C as low-rank matrix.

That helped to represent K-dimensional SSM layer as product of K 1-dimensional SSM layers.

Resolution Change and Bandlimiting

S4 model showed the ability to change the frequency of sampling for audio input zero-shot. To preserve this ability for spatial resolution due to the problem of aliasing in the spatial domain authors proposed a technique to avoid it. They applied a low-pass filter to remove frequencies above the Nyquist cutoff frequency.

2.4 ViS4mer

2.4.1 Transformer encoder

Assume we get a video $V \in R^{T \times H \times W \times 3}$ consisting of T frames denoted $f_1, \ldots, f_T$, H - the height of each frame, W - the width of each frame. We will apply encoder $\mathcal{E}$ for each frame independently extracting feature map for each frame.

The same as in ViT [24] we first divide each frame into N non-overlapping patches of size $P \times P$ where $N = \frac{HW}{P^2}$. Then we linearly project each patch to a latent dimension of size D and add a positional embedding to each projection resulting in z_i embedding.

After that the resulting sequence is feeded to thet transformer encoder $\mathcal{E}$. The encoder is a stack of L transformer blocks where each block consists of a multi-head attention (MHA) [22] and a multi-layer perceptron (MLP). Moreover Layer Normalization [11] is applied before each block and a skip-connection is processed after each one.

$$z' = MHA(LN(z_{in})) + z_{in}$$

$$z_{out} = MLP(LN(z')) + z'$$

Let's denote the output of each transformer as $h_1, \ldots, h_T$, where $h_i = \mathcal{E}(f_i) \in R^{H' \times W' \times D}$, where $H' = \frac{H}{P}, W' = \frac{W}{P}$. If we stack all of these outputs together we will get the video representation $X \in R^{T \times H' \times W' \times D}$.

2.4.2 Multi-scale Temporal S4 Decoder

Video feature map obtained from transformer can be hardly processed with transformers because it contains too much tokens. Instead authors utilized the S4 architecture and proposed a temporal multi-scale S4 decoder architecture for complex long-range reasoning. That helped to significantly reduce the computational costs compared to self-attention because of S4 linear computation and memory dependency with respect to the sequence length.

The authors introduced a novel multiscale S4 decoder architecture to effectively integrate the S4 layer into the visual domain. This temporal multiscale decoder comprises several blocks, each designed to function at varying spatial resolutions and channel dimensions. The model begins with high spatial resolution and channel dimension, and gradually decreases these parameters across the blocks. Inspired by successful visual domain models like Feature Pyramid Networks and Swin Transformer, this architecture enables each block to learn features at different scales, which is crucial for understanding complex spatiotemporal dependencies in long videos. Additionally, by handling shorter input sequences with reduced channel dimensions in the deeper blocks, the model effectively minimizes overfitting on the relatively small LVU benchmark. This multiscale approach not only enhances performance but also lowers computational costs and GPU memory usage.

The decoder network $\mathcal{D}$ consists of N blocks, which are defined below:

$$x_{s4} = S4(LN(x_{in}))$$

$$x_{mlp} = MLP(Pooling(x_{s4}))$$

$$x_{skip} = Linear(Pooling(x_{in}))$$

$$x_{out} = x_{mlp} + x_{skip}$$

S4 Layer

The X tensor (output from encoder) flattened to a sequence of $L = T \times H' \times W'$ vectors $x_{in} = (x_1, \ldots, x_L)$, where $x_i \in R^D$ and then passed to the S4 layer, which outputs $x_{s4} \in R^{L \times D}$

Spatiotemporal Resolution Reduction

The space-time resolution of input tensor is reduced by the factor $s_T \times s_H \times s_W$ using a max-

pooling layer where s_T, s_H, s_W are strides corresponded to temporal and spatial dimensions. This allows the model to learn multiscale spatiotemporal representations while also reducing overfitting.

Channel Reduction

To decrease the computational cost and overfitting the channel dimension is reduced by applying an MLP to the pooling layer output.

Skip Connections

Skip connections are also used but because of tensor size mismatching extra pooling is applied to the input to match the feature dimensionalities, after the linear layer is applied to match the channel dimension.

3 Methodology

3.1 S4ND on ChesapeakeRSC

The ChesapeakeRSC dataset is a high-resolution aerial dataset derived from the original Chesapeake dataset by filtering to retain roads heavily covered by tree canopies. This dataset poses a challenge because the tree canopy requires models to utilize long-range context to predict roads that are not directly visible. We addressed this problem using the S4ND model, which benefits from long-range context due to its use of S4 layers.

We followed the same pipeline used to evaluate other models in the "Seeing the Road through the Trees" paper. The model was trained with a standard cross-entropy loss [2] and an AdamW optimizer for 150 epochs. A cosine annealing learning rate schedule without restarts was employed, with a period of 100 epochs, starting at an initial learning rate of 1e-3 and reducing to a minimum of 1e-6. The model's performance was evaluated using the same metrics as in the paper to facilitate comparison with CNNs.

3.2 S4ND on LVU

The LVU benchmark uses publicly available MovieClips [13], which contains approximately 30,000 videos from around 3,000 movies, each about 1-3 minutes long. This benchmark comprises 9 different movie understanding tasks that cover a wide range of aspects of long-form video understanding: **content understanding** (relationship, way of speaking, scene/place), **metadata prediction** (director, genre, writer, year), and **user engagement** (like ratio, view count). Therefore, this benchmark requires models to effectively handle long temporal dependencies.

We followed the same strategy as outlined in the ViS4mer paper [9]. The content understanding and metadata prediction tasks were evaluated using the standard top-1 accuracy metric, while user engagement was assessed with Mean-Square Error (MSE) [12]. We also used standard splits and trained the model on 60-second video clips.

refs

3.3 S4ND on Breakfast and COIN

Next, we evaluated S4ND on the Breakfast [10] and COIN [20] datasets to demonstrate the model's ability to generalize to other domains. These datasets focus on long-range procedural activity classification. The Breakfast dataset contains 1,712 videos, with an average length of 2.32

minutes each, covering 10 complex cooking activities. The COIN dataset consists of 11,827 videos, each averaging 2.36 minutes in length, and includes 180 diverse procedural tasks.

We hypothesized that these datasets are well-suited for testing the model's capability in understanding long-range activities. Following the same pipeline as in [9], we used standard splits and measured performance in terms of activity classification accuracy.

3.4 ViS4NDmer architecture

We researched the S4ND and ViS4mer architectures and identified several drawbacks. While the S4ND model is effective with high-dimensional data, transformers tend to perform better with smaller datasets. ViS4mer, on the other hand, utilizes transformers to independently extract features from video frames. However, its S4 decoder struggles to fully leverage dimensional dependencies, even with scaled architecture tricks, as it was originally designed for 1D data.

Our idea is to replace the S4 decoder with the S4ND architecture, which is better suited for handling long-range data, resulting in the **ViS4NDmer** model. This change retains the advantage of the transformer encoder and enhances the decoder's ability to work with the encoder's multidimensional output.

3.5 ViS4NDmer evaluation

ViS4NDmer was evaluated on the LVU [25], Breakfast [10], and COIN [20] datasets using the same pipeline as for S4ND. This approach allowed for a direct comparison between ViS4NDmer, S4ND, and previous works.

4 Evaluation and Results

4.1 S4ND for aerial images segmentation

We compared S4ND model with CNNs architectures validated on ChesapeakeRSC benchmark 4.1.

Model	Background		Road		Tree Canopy over Road	
	R	P	R	P	R	DWR
FCN	99.4	98.3	64.1	71.5	23.4	10.7
U-Net (ResNet-18)	99.4	99.2	83.6	71.1	63.5	**46.5**
U-Net (ResNet-50)	**99.5**	99.2	84.0	71.8	63.5	45.7
DeepLabV3+ (ResNet-18)	99.4	99.1	82.6	71.9	58.9	41.3
DeepLabV3+ (ResNet-50)	**99.5**	**99.3**	84.8	72.0	63.9	46.1
S4ND	**99.5**	**99.3**	**85.2**	**72.2**	**64.0**	-

Table 4.1: Comparison S4ND to previous CNN models. Test set performance of each method (P=precision, R=recall, DWR=distance weighted recall).

As anticipated, the S4ND model outperformed all other CNN models due to its capacity to handle long-term context effectively. This experiment demonstrates S4ND's superiority over CNN models in tasks that require long-term contextual understanding, such as video benchmarks and 3D scene analysis. The ability of S4ND to maintain and process information over extended periods allows it to excel in these complex scenarios, highlighting its advanced capabilities in capturing and interpreting long-range dependencies within data.

4.2 COIN and Breakfast datasets

Next, we evaluated the S4ND and proposed ViS4NDmer models on video benchmarks. We compared the results with the original ViS4mer and other transformers mentioned in this paper using the Breakfast and COIN datasets. As stated in the paper "Long Movie Clip Classification with State-Space Video Models" [9], which introduced ViS4mer, these datasets are ideally suited to our model's capability in understanding long-range activities.

To ensure a fair comparison, we used the same evaluation pipeline to assess and compare the performance of the S4ND and ViS4NDmer models. This involved using standard splits and measuring activity classification accuracy. Our results for the Breakfast dataset are reported in Table 4.2, and for the COIN dataset in Table 4.3.

Model	Pretraining Dataset	Pretraining Samples	Accuracy (↑)
VideoGraph	Kinetics-400	306K	69.50
Timeception	Kinetics-400	306K	71.30
GHRM	Kinetics-400	306K	75.50
Distant Supervision	HowTo100M	136M	**89.90**
ViS4mer	Kinetics-600	495K	88.17
S4ND	Kinetics-600	495K	88.23
ViS4NDmer	Kinetics-600	495K	<u>88.31</u>

Table 4.2: Compare models on Breakfast dataset.

Model	Pretraining Dataset	Pretraining Samples	Accuracy (↑)
TSN	Kinetics-400	306K	73.40
Distant Supervision	HowTo100M	306K	**90.00**
ViS4mer	Kinetics-600	495K	88.41
S4ND	Kinetics-600	495K	88.59
ViS4NDmer	Kinetics-600	495K	<u>88.73</u>

Table 4.3: Compare models on COIN dataset.

The results indicate that ViS4NDmer slightly outperforms ViS4mer on these datasets due to its enhanced ability to leverage dependencies between temporal and spatial features in the decoder. However, Distant Supervision still outperforms ViS4NDmer, likely because of the much larger pretraining dataset used in Distant Supervision. This highlights the importance of extensive pretraining in achieving superior performance on complex tasks involving long-range activity understanding.

4.3 LVU

Lastly we evaluated S4ND and ViS4ndmer on LVU benchmark.

	Content (↑)			**Metadata (↑)**				**User (↓)**	
	Relation	**Speak**	**Scene**	**Director**	**Genre**	**Writer**	**Year**	**Like**	**Views**
SlowFast+NL [4],[23]	52.40	35.80	54.70	44.90	53.00	36.30	**52.50**	0.38	3.77
VideoBERT [19]	52.80	37.90	54.90	47.30	51.90	38.50	36.10	0.32	4.46
Obj. Transformer [25]	53.10	39.40	56.90	51.20	54.60	34.50	39.10	**0.23**	**3.55**
ViS4mer [9]	57.14	40.79	67.44	62.61	54.71	48.80	44.75	0.26	3.63
S4ND [14]	57.38	40.87	67.63	62.86	54.81	48.81	44.92	0.26	3.64
ViS4NDmer (ours)	**57.53**	**41.15**	**67.80**	**63.02**	**54.93**	**48.97**	45.06	0.25	3.59

Table 4.4: LVU results for S4ND, ViS4NDmer and previous works

	Content (↑)			Metadata (↑)				User (↓)			
	Relation	**Speak**	**Scene**	**Director**	**Genre**	**Writer**	**Year**	**Like**	**Views**	**Sam./s (↑)**	**Mem (↓)**
Self-attention	52.38	37.31	62.79	56.07	52.70	42.26	39.16	0.31	3.83	1.88	41.38
Performer	50.00	38.80	60.46	58.87	49.45	48.21	41.25	0.31	3.93	4.67	5.93
Orthoformer	50.00	39.30	66.27	55.14	**55.79**	47.02	43.35	0.29	3.86	4.85	5.56
S4ND	57.38	40.87	67.63	62.86	54.81	48.81	44.92	0.26	3.64		
ViS4NDmer (ours)	**57.53**	**41.15**	**67.80**	**63.02**	54.93	**48.97**	**45.06**	**0.25**	**3.59**		

Table 4.5: LVU results for S4ND, ViS4NDmer and previous works for efficient transformers

The results indicate that S4ND and ViS4NDmer outperform ViS4mer and other transformer-like models on the LVU benchmark. In terms of size and speed, S4ND and ViS4NDmer are comparable to, or even surpass, ViS4mer.

5 Conclusion

Our work demonstrated that while State Space Model (SSM) architectures may not surpass transformers across most tasks, they possess distinct advantages, particularly their capability to handle long contexts effectively. The ViS4NDmer model delivered the best performance in our experiments, indicating that integrating SSM models with transformers yields superior results.

Furthermore, the S4ND architecture shows promise as a multi-modal framework due to its use of 1D S4 layers. This characteristic makes it a compelling option for applications like text-to-image generation, where handling different types of data within a unified model is advantageous. The inherent design of S4ND to process sequential data efficiently suggests its potential effectiveness in various multi-modal tasks, providing a robust foundation for future research and development in this area.

Acknowledgement

We are grateful to Jet Brains Cadence team for providing GPUs on which experiments were run, we used Cadence plugin [1] for PyCharm.

References

[1] *Cadence plugin*. URL: https://plugins.jetbrains.com/plugin/17764-cadence.

[2] *Cross-Entropy Loss*. URL: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.

[3] Rahul Dey and Fathi M. Salem. "Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks". In: 2017. URL: https://arxiv.org/abs/1701.05923.

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. "Slowfast networks for video recognition". In: 2019. URL: https://arxiv.org/abs/1812.03982.

[5] Albert Gu, Tri Dao, Stefano Ermon, and et al. "HiPPO: Recurrent Memory with Optimal Polynomial Projections". In: 2020. URL: https://arxiv.org/abs/2008.07669.

[6] Albert Gu, Karan Goel, Christopher Ré, and et al. "Efficiently Modeling Long Sequences with Structured State Spaces". In: *ICLR*. 2022. URL: https://arxiv.org/abs/2111.00396.

[7] Albert Gu, Johnson Isys, Karan Goel, and et al. "Combining Recurrent, Convolutional, and Continuous-time Models with Linear State-Space Layers". In: *NeurIPS*. 2021. URL: https://arxiv.org/abs/2110.13985.

[8] Sepp Hochreiter and Jurgen Schmidhuber. "Long Short-term Memory". In: *Neural computation*. 1997. URL: https://arxiv.org/pdf/1805.03716.pdf.

[9] Md Mohaiminul Islam and Gedas Bertasius. "Long Movie Clip Classification with State-Space Video Models". In: *ECCV*. 2022.

[10] Hilde Kuehne, Ali Arslan, and Thomas Serre. "The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities". In: *CVPR*. 2014.

[11] *Layer Normalization*. URL: https://pytorch.org/docs/stable/generated/torch.nn.LayerNorm.html.

[12] *Mean Squared Error*. URL: https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html.

[13] *Movieclips*. URL: https://www.movieclips.com/.

[14] Eric Nguyen, Karan Goel, Albert Gu, and et al. "S4ND: Modeling Images and Videos as Multidimensional Signals Using State Spaces". In: *NeurIPS*. 2022. URL: https://arxiv.org/abs/2210.06583.

[15] Victor Pan. "Fast approximate computations with cauchy matrices and polynomials". In: *Mathematics of Computation*. 2017.

[16] Caleb Robinson, Isaac Corley, Anthony Ortiz, Rahul Dodhia, Juan M. Lavista Ferres, and Peyman Najafirad. "Seeing the roads through the trees: A benchmark for modeling spatial dependencies with aerial imagery". In: 2024.

[17] Robin M. Schmidt. "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview". In: 2014.

[18] Feifei Shen, Zhenjian Song, Congrui Wu, and et al. "Research on the fast Fourier transform of image based on GPU". In: *Mathematical Software*. 2015. URL: https://arxiv.org/abs/1505.08019.

[19] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. "VideoBERT: A Joint Model for Video and Language Representation Learning". In: *ICCV*. 2019. URL: https://arxiv.org/abs/1904.01766.

[20] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. "COIN: A Large-scale Dataset for Comprehensive Instructional Video Analysis". In: *CVPR*. 2019.

[21] Yi Tay, Mostafa Dehghani, Samira Abnar, and et al. "Long Range Arena: A Benchmark for Efficient Transformers". In: 2020. URL: https://arxiv.org/abs/2011.04006.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, and et al. "Attention Is All You Need". In: *Computation and Language*. 2017. URL: https://arxiv.org/abs/1706.03762.

[23] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. "Non-local Neural Networks". In: 2017. URL: https://arxiv.org/abs/1711.07971v3.

[24] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, and et al. "Visual Transformers: Token-based Image Representation and Processing for Computer Vision". In: *CVPR*. 2020. URL: https://arxiv.org/abs/2006.03677.

[25] Chao-Yuan Wu and Philipp Krähenbühl. "Towards Long-Form Video Understanding". In: *CVPR*. 2021.