

Computer Speech & Language
Keyword Mamba: Spoken Keyword Spotting with State Space Models
--Manuscript Draft--

| | |
|------------------------------|--|
| Manuscript Number: | YCSLA-D-25-00738 |
| Article Type: | Research Paper |
| Keywords: | Keyword spotting; Speech command recognition; Mamba; State space model; Temporal domain |
| Corresponding Author: | Qirong Mao Jiangsu University CHINA |
| First Author: | Hanyu Ding |
| Order of Authors: | Hanyu Ding Wenlong Dong Qirong Mao, Ph.D. |
| Abstract: | Keyword spotting (KWS) is an essential task in speech processing. It is widely used in voice assistants and smart devices. Deep learning models like CNNs, RNNs, and Transformers have performed well in KWS. However, they often struggle to handle long-term patterns and stay efficient at the same time. In this work, we present Keyword Mamba, a new architecture for KWS. It uses a neural state space model (SSM) called Mamba. We apply Mamba along the time axis and also explore how it can replace the self-attention part in Transformer models. We test our model on the Google Speech Commands datasets. The results show that Keyword Mamba reaches strong accuracy with fewer parameters and lower computational cost. To our knowledge, this is the first time a state space model has been used for KWS. These results suggest that Mamba has strong potential in speech-related tasks. |
| Opposed Reviewers: | |

Cover Letter

Dear Editor-in-Chief,

We are pleased to submit our manuscript entitled "**Keyword Mamba: Spoken Keyword Spotting with State Space Models**" for consideration in *Computer Speech and Language*. This work presents a novel architecture that introduces the Mamba selective state space model to the task of spoken keyword spotting (KWS), an essential component of modern speech interfaces.

Unlike traditional architectures that rely on convolutional or attention-based models, Keyword Mamba operates in the temporal domain using neural state space models (SSMs), enabling it to efficiently model long-range dependencies in audio sequences. Furthermore, we explore a hybrid variant that replaces self-attention in Transformers with Mamba layers, yielding strong performance gains with reduced computational cost.

We evaluate our models on the Google Speech Commands datasets (V1 and V2), where Keyword Mamba achieves new state-of-the-art accuracy with fewer parameters and lower inference latency compared to existing baselines. To the best of our knowledge, this is the first work applying Mamba to keyword spotting.

We believe that our findings will be of interest to readers in both the speech processing and machine learning communities, offering a fresh perspective on efficient sequence modeling in real-time audio systems.

We confirm that:

- The manuscript has not been published or submitted elsewhere.
- All authors have approved the manuscript and agree to its submission.
- We have disclosed any potential conflicts of interest (see below).

We look forward to your feedback and thank you for considering our work for publication in *Computer Speech and Language*.

Sincerely,

Hanyu Ding,

On behalf of all co-authors

Jiangsu University

Zhenjiang, China

Highlight

- Introduce Keyword Mamba, a novel Mamba-based model for keyword spotting (KWS).
- Apply state space modeling along the time axis to improve long-range speech modeling.
- Replace attention with Mamba in Transformer layers for better efficiency.
- Achieve state-of-the-art performance on Google Speech Commands datasets.
- Demonstrate lower parameter usage while maintaining high accuracy and robustness.

Keyword Mamba: Spoken Keyword Spotting with State Space Models

Hanyu Ding^a, Wenlong Dong^a and Qirong Mao^{a,b,c,*}

^aSchool of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, 212013, China

^bJiangsu Engineering Research Center of Big Data Ubiquitous Perception and Intelligent Agricultural Applications, Zhenjiang, 212013, China

^cProvincial Key Laboratory of Computational Intelligence and New Technologies in Low-Altitude Digital Agriculture, Zhenjiang, 212013, China

ARTICLE INFO

Keywords:

Keyword spotting
Speech command recognition
Mamba
State space model
Temporal domain

ABSTRACT

Keyword spotting (KWS) is an essential task in speech processing. It is widely used in voice assistants and smart devices. Deep learning models like CNNs, RNNs, and Transformers have performed well in KWS. However, they often struggle to handle long-term patterns and stay efficient at the same time. In this work, we present Keyword Mamba, a new architecture for KWS. It uses a neural state space model (SSM) called Mamba. We apply Mamba along the time axis and also explore how it can replace the self-attention part in Transformer models. We test our model on the Google Speech Commands datasets. The results show that Keyword Mamba reaches strong accuracy with fewer parameters and lower computational cost. To our knowledge, this is the first time a state space model has been used for KWS. These results suggest that Mamba has strong potential in speech-related tasks.

1. Introduction

Keyword spotting (KWS) [1] refers to the task of detecting specific keywords in audio streams comprising speech. As a critical entry point for human-computer interaction systems, KWS has been extensively applied in voice assistants, smart home devices, and intelligent cockpit systems [2, 3, 4, 5, 6]. Accurate keyword detection enables these applications to be activated in response to user commands, thereby initiating subsequent operations. Consequently, the advancement of KWS techniques has become a key area of research to enable more seamless human-computer interaction experiences.

In recent years, with the rapid advancement of deep learning technologies, KWS has also experienced significant growth. A major milestone in this development was the introduction of the first deep learning-based speech KWS system [7] in 2014, which marked the beginning of a new era in the field. Fig. 1 shows an illustration of the keyword spotting process using deep learning, in which spoken words are converted into audio signals and processed by neural networks. Since then, the field of KWS has witnessed a surge of research adopting representative neural network architectures, with convolutional neural networks (CNNs) [8, 9, 10], recurrent neural networks (RNNs) [11, 12, 13], and Transformers [14, 15] emerging as the predominant frameworks. Notably, BC-ResNet [10], a convolutional neural network based on broadcasted residual learning, combines the strengths of 1D temporal and 2D spatial convolutions while minimizing additional computational cost, and demonstrates strong performance in the KWS task. On the RNN side, MHAtt-RNN [13] introduces multi-head attention mechanisms, which significantly improve the model's ability to capture key temporal dependencies in speech sequences. In addition, Transformer-based architectures such as the Keyword Transformer (KWT) [15] further advance the field by leveraging self-attention to effectively model long-range contextual information.

Despite the success of these models in KWS applications, several inherent limitations still persist. For instance, the fixed receptive field of CNNs restricts their ability to capture long-range temporal dependencies. RNNs, due to their sequential computation nature, are difficult to parallelize, resulting in slower training and inference speeds. Moreover, Transformer models suffer from quadratic growth in computational complexity with respect to the size of the input context window, which significantly limits their efficiency and scalability when processing long sequences.

To address these inherent challenges, a new architecture based on neural state space models (SSMs) [16, 17, 18] known as Mamba [19], has recently emerged and achieved performance surpassing state-of-the-art methods across a

*Corresponding author at: School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, 212013, China.

*E-mail addresses: mao_qr@ujs.edu.cn (Q. Mao).

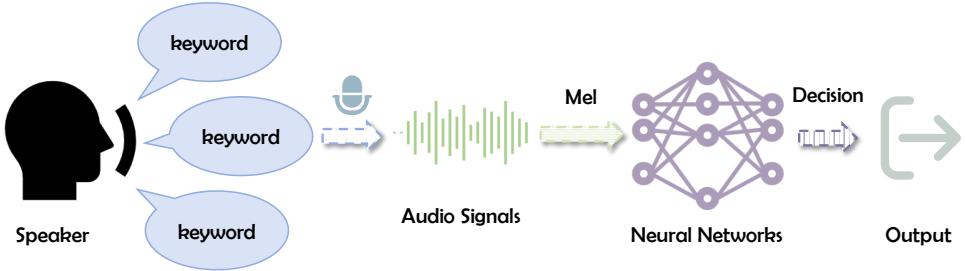


Figure 1: Keyword spotting using deep learning illustration.

range of tasks [20, 21, 22]. In the field of speech processing, recent studies have explored replacing Transformers with Mamba for tasks such as speech enhancement [23] and automatic speech recognition [23], demonstrating promising initial results. Since different speech tasks [24, 25, 26] focus on various characteristics of the speech signal, such as speaker identity, linguistic content, or emotional state, it remains unclear how to effectively apply the Mamba architecture to other speech-related tasks such as spoken KWS.

In this work, inspired by Mamba and Vision Mamba (Vim) [22], we introduce them to KWS by developing a novel architecture called Keyword Mamba (KWM). Our objective is to explore the application of Mamba in KWS. Specifically, we apply Mamba solely along the temporal domain, leveraging its capability to effectively model long-range dependencies in order to enhance KWS performance while maintaining computational efficiency. In addition, to further investigate the potential of Mamba, we attempt to replace the multi-head self-attention (MHSA) module [14] in the Transformer architecture with Mamba, rather than replacing the entire model, aiming to improve the nonlinear modeling capacity of Mamba. We evaluate Keyword Mamba on various versions of the Google Speech Commands datasets [27], demonstrating its effectiveness and robustness in real-world scenarios. To the best of our knowledge, this is the first study to apply a state space-based model to KWS, highlighting the novelty and significance of our approach.

2. Related Works

2.1. State Space Models

SSMs [16, 17, 18, 19] are statistical frameworks designed for modeling time-series data. They are widely used in machine learning and statistics to handle dynamic systems and time-varying processes. SSMs are capable of modeling the evolution of latent system states over time and capturing the relationships between these hidden states and the observed data. Importantly, SSMs not only support long-range dependency modeling but also offer linear computational complexity. Recently, the advantages of SSMs in sequence analysis have gained more and more attention from scholars. Among them, the structured state space sequence model (S4) [17] and the selective state space model (Mamba) [19] are typically represented.

Structured State Space Sequence Model (S4). The S4 is an efficient sequence modeling approach designed to capture long-range dependencies using SSMs. It incorporates three core mechanisms: HiPPO [28] for memory of input history, Diagonal Plus Low-Rank Parameterization for stability and diagonalizability, and Efficient Kernel Computation via FFTs and iFFTs. The S4 significantly reduces computational complexity to $O(N \log(N))$ and achieves strong performance on tasks like path-X in the LRA benchmark. Moreover, its effectiveness extends to various other benchmark tasks as well. Although this model still falls short of state-of-the-art Transformers in terms of performance, it continues to close the gap and provides improved computational efficiency [29].

Selective State Space Model (Mamba). Based on the S4, Mamba is proposed. Mamba is a more efficient state space modeling approach to address the limitations of traditional SSMs in tasks such as selective copying and induction [30], while avoiding the quadratic computational complexity $O(N^2)$ and memory complexity of Transformers. It adopts an input-aware parameterization scheme combined with a simplified selection mechanism and introduces an efficient, hardware-friendly selective scan algorithm. Mamba also employs a gating mechanism to reduce the dimensionality of kernel operations and integrates a gated MLP [31] to further enhance its modeling capability. In both computer vision (CV) [22] and natural language processing (NLP) [32] tasks, Mamba has demonstrated outstanding performance and effectiveness. Meanwhile, with linear-time complexity $O(N)$, Mamba is more efficient than traditional Transformers.

2.2. Visual Mamba

With the ongoing development and adoption of Mamba, the model has been rapidly extended to the visual domain. Vision Mamba (Vim) [22], one of the earliest and most representative visual Mamba models, draws inspiration from Vision Transformers (ViT) [33] by incorporating position embeddings for spatial information encoding and leveraging bidirectional SSMs to process non-causal image sequences. Another variant, VMamba [34], unfolds image patches into sequences along both the horizontal and vertical axes, enabling bidirectional scanning in both directions. Likewise, several other notable works have investigated Mamba-based visual backbones [35, 36, 37, 38], consistently reporting competitive performance across various vision tasks, including classification, detection, and segmentation.

2.3. Speech Mamba

Recent works have explored the use of SSMs and Mamba methods in speech processing tasks. Notably, several studies investigated self-supervised audio Mamba models trained using masked spectrogram modeling. For example, Audio Mamba (AuM) [39] and its variant for audio tagging achieve comparable or superior performance to Audio Spectrogram Transformers (AST) [40], while using only about one-third of the parameters. Furthermore, Mamba has been applied to specific tasks such as speech enhancement [41, 42], automatic speech recognition [43, 44] and speech separation [26, 45, 46]. However, the effective design of Mamba models for deep KWS remains unexplored. Unlike these other tasks, KWS is less sensitive to frequency information [9]. Therefore, we propose applying Mamba solely within the temporal domain. This leverages its ability to model long-range dependencies efficiently, aiming to enhance KWS performance while maintaining efficiency.

3. Proposed Keyword Mamba

The goal of Keyword Mamba is to introduce the advanced SSM (i.e., Mamba) into the KWS task and explore how to effectively apply Mamba in this context. This section begins with the fundamental principles of Mamba, followed by a comprehensive overview of the Keyword Mamba framework and a detailed explanation of its key architectural component (i.e., Mamba Encoder).

3.1. Preliminaries

Mamba is built upon the S4, retaining its core SSM framework while introducing a selective mechanism to enhance its ability to focus on task-relevant features.

3.1.1. Structured State Space Sequence Model (S4)

The core dynamics of SSMs lie in how it updates the latent state from one time step to the next, and how it derives the output sequence from this latent state:

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t), \\ y(t) &= Ch(t). \end{aligned} \tag{1}$$

Here, A denotes the state transition matrix, B is the input-to-state projection matrix, and C is the state-to-output projection matrix. For the rest, $x(t)$ represents the input at time step t , $h(t)$ denotes the latent state, and $y(t)$ is the generated output. Since most practical applications involve discrete data such as text, it is necessary to discretize SSMs. This is achieved by introducing a special fourth parameter, Δ , which is used to convert the continuous parameters A and B into their discrete counterparts:

$$\begin{aligned} \bar{A} &= \exp(\Delta A), \\ \bar{B} &= (\Delta A)^{-1} (\exp(\Delta A) - I) \cdot \Delta B. \end{aligned} \tag{2}$$

Here, \bar{A} and \bar{B} represent the parameters obtained through the discretization of their continuous-time counterparts. After applying discretization, the equations take the following form:

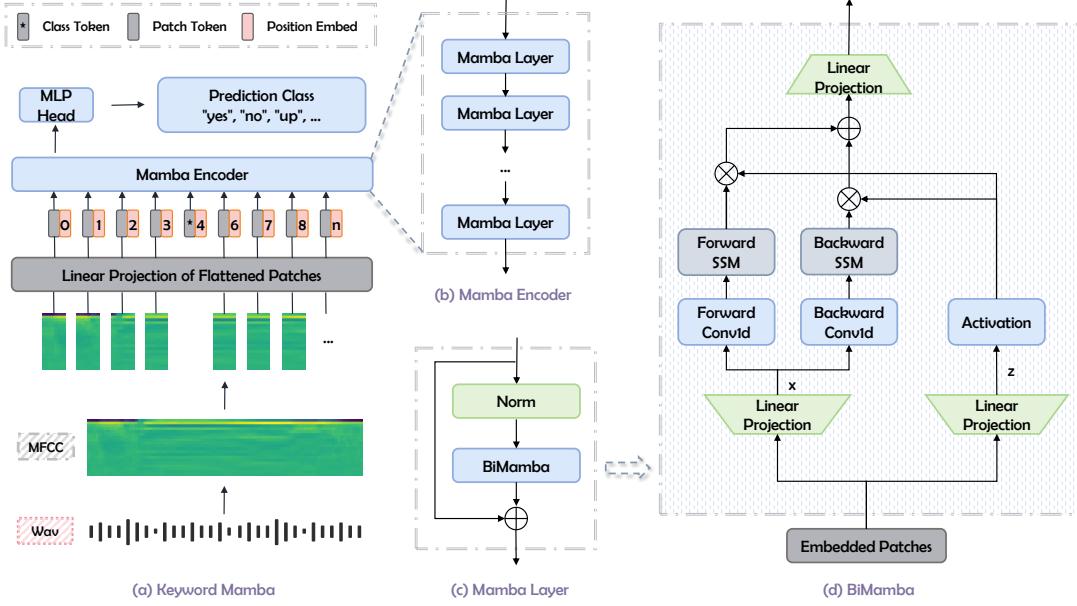


Figure 2: Overview of the Keyword Mamba architecture for keyword spotting: (a) Keyword Mamba; (b) Mamba Encoder of Keyword Mamba; (c) Constituent Layers of Mamba Encoder; (d) Bidirectional Mamba (BiMamba) of Constituent Layers.

$$h'(t) = \bar{A}h(t-1) + \bar{B}x(t), \\ y(t) = Ch(t). \quad (3)$$

To simplify calculations, the repeated application of the equation can be efficiently performed simultaneously using a global convolution approach:

$$y = x \circledast \bar{K}, \\ \text{where } \bar{K} = \left(C\bar{B}, C\bar{AB}, \dots, C\bar{A}^{L-1}\bar{B} \right). \quad (4)$$

Here, L is the length of the input sequence x , \circledast denotes convolution operation, and \bar{K} is the SSM kernel.

3.1.2. Selective State Space Model (Mamba)

Based on the S4, Mamba improves the performance of SSMs by introducing Selective State Space Models, allowing the continuous parameters to vary with the input enhances selective information processing across sequences, which extend the discretization process by selection mechanism [47]:

$$\bar{B} = s_B(x), \quad \bar{C} = s_C(x), \\ \Delta = \tau_A (\text{Parameter} + s_A(x)). \quad (5)$$

Here, $s_B(x)$ and $s_C(x)$ are linear functions that project input x into an N -dimensional space, while $s_A(x)$ broadens a D -dimensional linear projection to the fixed dimensions.

3.2. Keyword Mamba

The Keyword Mamba architecture, as depicted in Fig. 2, begins by transforming an input audio waveform into the MFCC spectrogram [48] $M \in \mathbb{R}^{F \times T}$, where F and T represent the frequency and time dimensions, respectively. The spectrogram is partitioned into a sequence of N patches $X \in \mathbb{R}^{N \times (f \cdot t)}$, with f and t denoting the shape of each patch

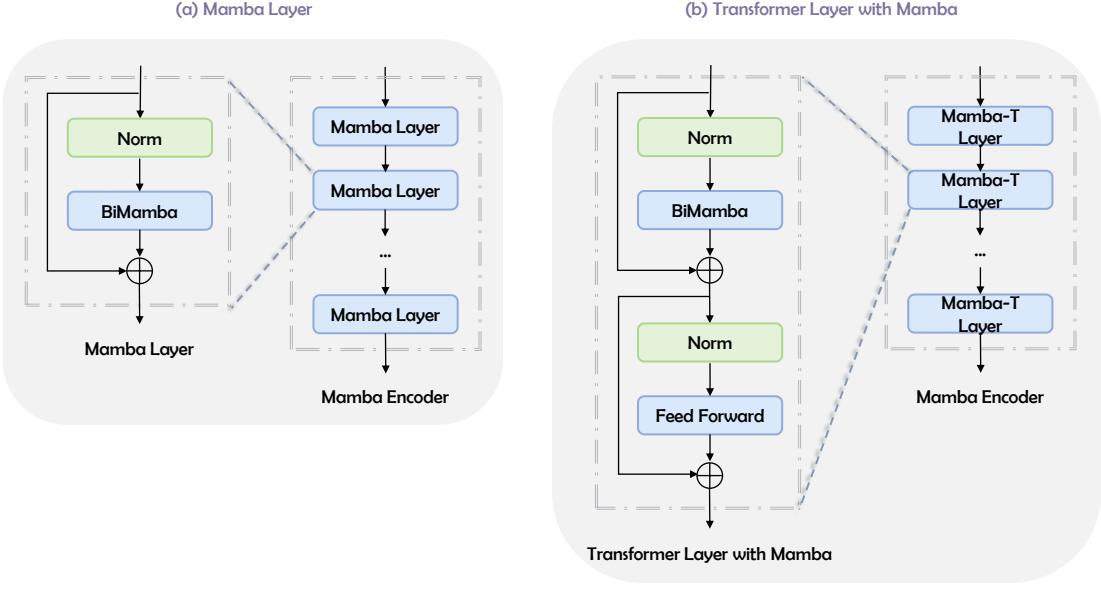


Figure 3: Task-Aware Encoder Designs: (a) Mamba Layer: using stacked BiMamba layers as an alternative to Transformer layers; (b) Transformer Layer with Mamba: replacing MHSA in Transformer layer with BiMamba.

and N calculated as $N = (F/f) \times (T/t)$. Since Mamba is applied exclusively in the time domain, $f = F$, $t = 1$, and $N = T$, which means that after flattening, the sequence of patches can be represented as $X \in \mathbb{R}^{T \times F}$. Then, the sequence is mapped to a higher dimension d using a linear projection matrix $W_0 \in \mathbb{R}^{F \times d}$ in the frequency domain. To learn a global representation of the entire spectrogram, we insert a learnable class embedding [33] $X_{\text{class}} \in \mathbb{R}^{1 \times d}$ into the middle of the projected sequence like Vim [22]. Specifically, the class token is placed between the first and second halves of the sequence after projection. A learnable positional embedding $X_{\text{pos}} \in \mathbb{R}^{(T+1) \times d}$ is then added, and the final input to the Mamba Encoder is defined as:

$$X_0 = [X_{[1:T/2]} W_0; X_{\text{class}}; X_{[T/2]+1:T} W_0] + X_{\text{pos}}. \quad (6)$$

Then, the resulting sequence X_{l-1} is fed into the l -layer of the Mamba Encoder to produce the output X_l . The final output is obtained by normalizing the class token X_L^0 , and then passing it through an MLP head for classification, where L is the number of layers:

$$\begin{aligned} X_l &= \text{Mamba Encoder}(X_{l-1}) + X_{l-1}, \\ f &= \text{Norm}(X_L^0), \quad p = \text{MLP}(f). \end{aligned} \quad (7)$$

3.3. Mamba Encoder

The Mamba Encoder serves as the core architectural component of Keyword Mamba. In this section, we first present the specific Mamba structure (i.e. BiMamba) employed in our work, followed by the Mamba-based encoder designed specifically for the KWS task.

3.3.1. BiMamba

The original Mamba [19] performs causal computations in a unidirectional manner, relying solely on historical information. However, bidirectional processing is essential for speech tasks in which the entire utterance is available [23]. To address this, Mamba requires to support the bidirectional processing, similar to the MHSA module, in order to capture global dependencies within the input signal features. In this study, we employ the Bidirectional Mamba design (BiMamba) [22] inspired by Vim, aiming to enhance the model's ability to capture contextual dependencies in both temporal directions. In other words, BiMamba enables the model to process information from past and future contexts.

The BiMamba block is shown in Fig. 2(d). Specifically, BiMamba employs two sets of SSM and Conv1D modules (one forward and one backward) that share the same input and output projection layers. The input embedding, which is first processed by the input projection layer, is then fed separately into a forward SSM and a backward SSM, resulting in two outputs. After that, the outputs are combined and passed through the output projection layer.

3.3.2. Task-Aware Encoder Designs

For the Mamba Encoder in Keyword Mamba, we explore two different design variants to capture information of various abstraction levels, as shown in Fig. 3: one composed entirely of Mamba layers, and the other incorporating additional nonlinear transformations by integrating Mamba components into Transformer layers.

Mamba Layer. As depicted in Fig. 3(a), the first strategy uses the BiMamba layers independently (i.e., as a direct replacement for the Transformer layer) to construct the Mamba Encoder. The objective is to directly leverage Mamba’s ability to capture long-range dependencies within the input data, thereby enhancing the effectiveness of KWS. This design can better focus on temporal patterns relevant to keyword activation, while maintaining a lightweight architecture suitable for real-time or low-resource scenarios. Algo. 1 illustrates the workflow of the Mamba Layer.

Transformer Layer with Mamba. The second approach employs the BiMamba layer to replace the MHSA modules within the Transformer, where the feed-forward network (FFN) and normalization are used to provide additional nonlinearity. This design choice is primarily motivated by the fact that SSMs are largely composed of linear operations. Although the SiLU [49] activation is used in the residual structure in practice, its capacity to capture high-level information (e.g., semantics) remains limited. Therefore, introducing additional nonlinear components is essential for enhancing Mamba’s ability to model higher-level representations. Algo. 2 illustrates the workflow of the Transformer Layer with Mamba.

4. Experimental Setup

4.1. Datasets and Metrics

We evaluate the performance of the proposed Keyword Mamba model on Google Speech Commands datasets V1 and V2 [27]. V1 of the dataset contains 64,727 snippets of 30 different words from various speakers, whereas V2 contains 105,829 snippets of 35 different words. The 12-label classification task uses 10 words: “up”, “down”, “left”, “right”, “yes”, “no”, “on”, “off”, “go”, and “stop”, in addition to “silence” and “unknown”, where instances of the latter is taken from the remaining words in the dataset, whereas the 30-label task and the 35-label task use all available words. We use the same 80:10:10 train/validation/test data split ratio as [13, 15] for side-by-side comparisons and follow the evaluation criteria of [13] as closely as possible.

For these datasets, the primary metric is accuracy (ACC) [13]. For each experiment, we train the model three times with different random initializations. Therefore, the final results are presented as average values.

4.2. Implementation Details

The model is trained for 200 epochs on V1 and 140 epochs on V2 with a batch size of 128, using the AdamW optimizer. The initial learning rate is set to 0.001, and a cosine learning rate schedule is applied. Additionally, the training includes 10 warmup epochs. For regularization, a weight decay of 0.1 and label smoothing of 0.1 are used. As for the loss function, the cross-entropy loss is used.

During preprocessing, the input audio is converted into a sequence of 40-dimensional MFCC calculated from 30ms window with 10ms stride. Then, we pad the time dimension with zeros to a fixed length of 98 feature vectors per sample. This means that the frequency dimension of MFCC is 40 and the time dimension is 98. To improve generalization, various data augmentation techniques are applied when training. These include time shifting in the range of [-100, 100] ms, resampling within [0.85, 1.15], and adding background noise with a volume of 0.1. In addition, 2 time masks are applied, with sizes in the range of [0, 25], and 2 frequency masks are applied, with sizes in the range of [0, 7] [50].

4.3. Model Variants

Due to the parameter constraints inherent and lightweight requirements in the KWS task, we adopt the same dimensional configurations as KWT [15], setting the model dimensions to 192, 128, and 64, respectively.

Algorithm 1 Mamba Layer Workflow

Require: $\mathbf{X}_{l-1} : (B, L, D)$
Ensure: $\mathbf{X}_l : (B, L, D)$

- 1: $\mathbf{X}'_{l-1} : (B, L, D) \leftarrow \text{Norm}(\mathbf{X}_{l-1})$
- 2: $\mathbf{x} : (B, L, E) \leftarrow \text{Linear}^{\mathbf{x}}(\mathbf{X}'_{l-1})$
- 3: $\mathbf{z} : (B, L, E) \leftarrow \text{Linear}^{\mathbf{z}}(\mathbf{X}'_{l-1})$
- 4: **for** $o \in \{\text{forward, backward}\}$ **do**
- 5: $\mathbf{x}'_o : (B, L, E) \leftarrow \text{SiLU}(\text{Conv1d}_o(\mathbf{x}))$
- 6: $\mathbf{B}_o : (B, L, N) \leftarrow \text{Linear}_o^{\mathbf{B}}(\mathbf{x}'_o)$
- 7: $\mathbf{C}_o : (B, L, N) \leftarrow \text{Linear}_o^{\mathbf{C}}(\mathbf{x}'_o)$
- 8: $\Delta_o : (B, L, N) \leftarrow \log(1 + \exp(\text{Linear}_o^{\Delta}(\mathbf{x}'_o) + \text{Parameter}_o^{\Delta}))$
- 9: $\bar{\mathbf{A}}_o : (B, L, E, N) \leftarrow \Delta_o \otimes \text{Parameter}_o^{\mathbf{A}}$
- 10: $\bar{\mathbf{B}}_o : (B, L, E, N) \leftarrow \Delta_o \otimes \mathbf{B}_o$
- 11: $\mathbf{y}_o : (B, L, E) \leftarrow \text{SSM}(\bar{\mathbf{A}}_o, \bar{\mathbf{B}}_o, \mathbf{C}_o)(\mathbf{x}'_o)$
- 12: **end for**
- 13: $\mathbf{y}'_{\text{forward}} : (B, L, E) \leftarrow \mathbf{y}_{\text{forward}} \odot \text{SiLU}(\mathbf{z})$
- 14: $\mathbf{y}'_{\text{backward}} : (B, L, E) \leftarrow \mathbf{y}_{\text{backward}} \odot \text{SiLU}(\mathbf{z})$
- 15: $\mathbf{X}_l : (B, L, D) \leftarrow \text{Linear}^{\mathbf{X}}(\mathbf{y}'_{\text{forward}} + \mathbf{y}'_{\text{backward}}) + \mathbf{X}_{l-1}$
- 16: **return** \mathbf{X}_l

Algorithm 2 Transformer Layer with Mamba Workflow

Require: $\mathbf{X}_{l-1} : (B, L, D)$
Ensure: $\mathbf{X}_l : (B, L, D)$

- 1: $\mathbf{X}'_{l-1} : (B, L, D) \leftarrow \text{Norm}(\mathbf{X}_{l-1})$
- 2: $\mathbf{x} : (B, L, E) \leftarrow \text{Linear}^{\mathbf{x}}(\mathbf{X}'_{l-1})$
- 3: $\mathbf{z} : (B, L, E) \leftarrow \text{Linear}^{\mathbf{z}}(\mathbf{X}'_{l-1})$
- 4: **for** $o \in \{\text{forward, backward}\}$ **do**
- 5: $\mathbf{x}'_o : (B, L, E) \leftarrow \text{SiLU}(\text{Conv1d}_o(\mathbf{x}))$
- 6: $\mathbf{B}_o : (B, L, N) \leftarrow \text{Linear}_o^{\mathbf{B}}(\mathbf{x}'_o)$
- 7: $\mathbf{C}_o : (B, L, N) \leftarrow \text{Linear}_o^{\mathbf{C}}(\mathbf{x}'_o)$
- 8: $\Delta_o : (B, L, N) \leftarrow \log(1 + \exp(\text{Linear}_o^{\Delta}(\mathbf{x}'_o) + \text{Parameter}_o^{\Delta}))$
- 9: $\bar{\mathbf{A}}_o : (B, L, E, N) \leftarrow \Delta_o \otimes \text{Parameter}_o^{\mathbf{A}}$
- 10: $\bar{\mathbf{B}}_o : (B, L, E, N) \leftarrow \Delta_o \otimes \mathbf{B}_o$
- 11: $\mathbf{y}_o : (B, L, E) \leftarrow \text{SSM}(\bar{\mathbf{A}}_o, \bar{\mathbf{B}}_o, \mathbf{C}_o)(\mathbf{x}'_o)$
- 12: **end for**
- 13: $\mathbf{y}'_{\text{forward}} : (B, L, E) \leftarrow \mathbf{y}_{\text{forward}} \odot \text{SiLU}(\mathbf{z})$
- 14: $\mathbf{y}'_{\text{backward}} : (B, L, E) \leftarrow \mathbf{y}_{\text{backward}} \odot \text{SiLU}(\mathbf{z})$
- 15: $\tilde{\mathbf{X}}_l : (B, L, D) \leftarrow \text{Linear}^{\tilde{\mathbf{X}}}(\mathbf{y}'_{\text{forward}} + \mathbf{y}'_{\text{backward}}) + \mathbf{X}_{l-1}$
- 16: $\tilde{\mathbf{X}}'_l : (B, L, D) \leftarrow \text{Norm}(\tilde{\mathbf{X}}_l)$
- 17: $\mathbf{X}_l : (B, L, D) \leftarrow \text{FeedForward}^{\mathbf{X}}(\tilde{\mathbf{X}}'_l) + \tilde{\mathbf{X}}_l$
- 18: **procedure** FEEDFORWARD(\mathbf{X})
- 19: $\mathbf{X} : (B, L, D)$
- 20: $\mathbf{H} : (B, L, D_f) \leftarrow \text{Linear}_1(\mathbf{X})$
- 21: $\mathbf{G} : (B, L, D_f) \leftarrow \text{GELU}(\mathbf{H})$
- 22: $\mathbf{O} : (B, L, D) \leftarrow \text{Linear}_2(\mathbf{G})$
- 23: **return** \mathbf{O}
- 24: **end procedure**
- 25: **return** \mathbf{X}_l

Table 1

Accuracy comparison of various models on Speech Commands V1 and V2 benchmarks. KWM and KWM-T achieve state-of-the-art performance while maintaining smaller model sizes compared to Transformer-based baselines. The results exceeding existing best methods are highlighted.

| Model | Para | V1-12 | V1-30 | V2-12 | V2-35 |
|-------------------------|-------|--------------|--------------|--------------|--------------|
| Att-RNN [11] | 202K | 95.6 | 96.29 | 96.9 | 93.9 |
| MHAtt-RNN [13] | 743K | 97.2 | 97.02 | 98.0 | 97.27 |
| TC-ResNet14-1.5 [9] | 305K | 96.6 | 96.99 | 97.43 | 95.43 |
| Matchboxnet-3x2x64 [51] | 93K | 97.48 | 97.23 | 97.21 | 97.46 |
| BC-ResNet-8 [10] | 321K | 98.0 | 97.46 | 98.7 | 97.65 |
| ConvMixer [52] | 119K | 97.3 | 97.05 | 98.2 | 96.83 |
| DenseNet-BiLSTM [53] | 250K | 97.5 | / | 97.4 | / |
| SincConv+DSConv [54] | 122K | 96.6 | / | 97.4 | / |
| SincConv+GDSConv [54] | 62K | 96.4 | / | 97.3 | / |
| NoisyDARTS-TC14 [55] | 108K | 96.79±0.30 | / | 97.18±0.26 | / |
| LG-Net6 [56] | 321K | 97.67 | / | 96.79 | / |
| KWT-3 [15] | 5361K | 97.49±0.15 | 96.83 | 98.56±0.07 | 97.69±0.09 |
| KWT-2 [15] | 2394K | 97.27±0.08 | 96.34 | 98.43±0.08 | 97.74±0.03 |
| KWT-1 [15] | 607K | 97.26±0.18 | 95.96 | 98.08±0.10 | 96.95±0.14 |
| KWM-192 | 3.4M | 98.01 | 97.73 | 98.79 | 97.86 |
| KWM-128 | 1.6M | 97.95 | 97.54 | 98.62 | 97.84 |
| KWM-64 | 0.5M | 97.46 | 97.07 | 98.13 | 97.12 |
| KWM-T-192 | 5.2M | 98.05 | 97.80 | 98.91 | 97.89 |
| KWM-T-128 | 2.4M | 97.88 | 97.70 | 98.73 | 97.86 |
| KWM-T-64 | 0.7M | 97.72 | 97.69 | 98.56 | 97.75 |

Since the Mamba Encoder in the Keyword Mamba comprises two distinct variants, for ease of distinction in our subsequent experiments, we denote the inclusion of the Mamba Layer and the Transformer Layer with Mamba as KWM and KWM-T, respectively.

5. Results and Analysis

5.1. Keyword Mamba vs. Other SOTA Models

To ensure a fair comparison with other state-of-the-art (SOTA) models, we set the number of layers in Keyword Mamba to 12, matching the configuration of the KWT series [15]. Table 1 summarizes the average accuracy across four evaluation settings from the Speech Commands V1 and V2 datasets. As shown in the results, Keyword Mamba consistently outperforms existing models across all datasets, establishing new benchmark records. These gains are not only in accuracy but also in parameter count and efficiency.

Relative to Transformer-based models like KWT, Keyword Mamba achieves better accuracy with fewer parameters, demonstrating its efficiency in both training and inference. For instance, while KWT-3 requires over 5M parameters, Keyword Mamba reaches higher accuracy with only 3.4M parameters, confirming its strong parameter efficiency.

In addition, Keyword Mamba shows competitive or superior performance when compared to CNN-based models such as BC-ResNet [10] and MatchboxNet [51]. It also surpasses RNN-based models like Att-RNN [11] and MHAtt-RNN [13], which typically struggle with long-range temporal patterns and parallel computation. It is worth emphasizing that this is the first known application of Mamba to the keyword spotting task. Despite its novelty, Keyword Mamba not only performs well but also demonstrates strong generalization across different subsets. These results clearly support the effectiveness of state space models in speech tasks, especially in learning discriminative and time-aware features from audio signals.

5.2. Keyword Mamba - KWM (Mamba Layer)

To explore whether the success of Mamba in other domains can be transferred to the KWS task, we conduct detailed experiments on the KWM model using pure Mamba layers. Specifically, we vary both the model dimension (192, 128,

Table 2

Effect of Mamba layer depth and model size (dimension) on accuracy. KWM shows strong performance even at low parameter counts, confirming its scalability and efficiency. The best results are **highlighted**.

| Model | Para | V1-12 | V1-30 | V2-12 | V2-35 |
|----------------|------|--------------|--------------|--------------|--------------|
| Dim 192 | | | | | |
| KWM-12 | 3.4M | 98.01 | 97.73 | 98.79 | 97.86 |
| KWM-10 | 2.9M | 97.95 | 97.54 | 98.42 | 97.62 |
| KWM-8 | 2.3M | 97.70 | 97.41 | 98.40 | 97.62 |
| KWM-6 | 1.7M | 97.92 | 97.39 | 98.52 | 97.44 |
| Dim 128 | | | | | |
| KWM-12 | 1.6M | 97.95 | 97.54 | 98.62 | 97.84 |
| KWM-10 | 1.4M | 97.62 | 97.44 | 98.48 | 97.67 |
| KWM-8 | 1.1M | 97.75 | 97.29 | 98.52 | 97.45 |
| KWM-6 | 0.8M | 97.33 | 97.11 | 98.31 | 97.32 |
| Dim 64 | | | | | |
| KWM-12 | 0.5M | 97.46 | 97.07 | 98.13 | 97.12 |
| KWM-10 | 0.4M | 97.27 | 96.98 | 97.92 | 97.09 |
| KWM-8 | 0.3M | 97.43 | 96.71 | 98.01 | 97.00 |
| KWM-6 | 0.2M | 96.74 | 96.52 | 97.45 | 96.92 |

64) and layer depth (12, 10, 8, 6), as shown in Table 2. The results show that KWM maintains high and stable accuracy across different configurations. For example, KWM-192 with 12 layers achieves 98.01% on V1-12 and 98.79% on V2-12. Even when the model is reduced to just 6 layers and a dimension of 64, it still reaches 96.74%–97.45%, which is comparable to many larger CNN or RNN models. This behavior reveals a key strength of Mamba-based designs: robust performance under compression. Unlike Transformer-based or convolutional models that often lose accuracy when downsized, KWM continues to perform well with significantly fewer parameters. This makes it ideal for real-world applications where computational and memory resources are limited, such as mobile or embedded devices. The strong performance of small KWM models is likely due to Mamba’s ability to model long-range dependencies efficiently without relying on attention mechanisms. Since Mamba uses a recurrent state-based structure, it can capture temporal patterns over extended time spans, even in smaller architectures. This explains why deeper or wider configurations bring improvements, but are not essential for competitive accuracy. In summary, this experiment supports our core claim: KWM is scalable and effective across different model sizes, proving that the state space modeling capability of Mamba translates well to the audio domain, especially for KWS tasks.

5.3. Keyword Mamba - KWM-T (Transformer Layer with Mamba)

To further explore how Mamba can enhance KWS, we construct a hybrid model called KWM-T. In this variant, we keep the feed-forward layer from the Transformer structure but replace the multi-head self-attention module with Mamba, as shown in Table 3. This design aims to combine Mamba’s efficient sequence modeling with the Transformer’s strong nonlinear transformation capacity.

We replicate the model size and depth settings used in Section 5.2 to allow direct comparison. As reported in Table 3, KWM-T consistently achieves competitive or even higher accuracy than the original KWM under similar parameter budgets. For example, KWM-T-192 reaches 98.05% on V1-12 and 98.91% on V2-12, surpassing both KWM-192 and most Transformer baselines from Table 1.

Even under reduced settings, such as KWM-T-64, the model achieves 97.72% on V1-12 and 97.75% on V2-35, confirming that this hybrid design maintains strong performance across a wide range of resource levels. This result supports a key finding: Integrating Mamba with Transformer-style feed-forward layers leads to better KWS performance by combining efficient memory modeling with strong nonlinear capacity.

The feed-forward layer helps Mamba go beyond linear state-space behavior, allowing it to better represent complex speech features. Meanwhile, Mamba replaces the attention mechanism, offering lower computational cost and better scalability for longer input sequences. In summary, KWM-T offers a highly effective trade-off between accuracy and complexity, and the results confirm that this hybrid architecture is a promising design choice for future speech models.

Table 3

Effect of Mamba-integrated Transformer depth and size on accuracy. KWM-T models benefit from replacing self-attention with Mamba, achieving high accuracy with moderate complexity.

| Model | Para | V1-12 | V1-30 | V2-12 | V2-35 |
|----------------|------|--------------|--------------|--------------|--------------|
| Dim 192 | | | | | |
| KWM-T-12 | 5.2M | 98.05 | 97.80 | 98.91 | 97.89 |
| KWM-T-10 | 4.3M | 97.85 | 97.86 | 98.62 | 97.85 |
| KWM-T-8 | 3.5M | 97.85 | 97.69 | 98.58 | 97.67 |
| KWM-T-6 | 2.6M | 97.62 | 97.57 | 98.42 | 97.63 |
| Dim 128 | | | | | |
| KWM-T-12 | 2.4M | 97.88 | 97.70 | 98.73 | 97.86 |
| KWM-T-10 | 2.0M | 97.69 | 97.55 | 98.52 | 97.76 |
| KWM-T-8 | 1.6M | 97.49 | 97.64 | 98.48 | 97.64 |
| KWM-T-6 | 1.2M | 97.56 | 97.57 | 98.36 | 97.49 |
| Dim 64 | | | | | |
| KWM-T-12 | 0.7M | 97.72 | 97.69 | 98.56 | 97.75 |
| KWM-T-10 | 0.6M | 97.49 | 97.38 | 98.33 | 97.61 |
| KWM-T-8 | 0.5M | 97.40 | 97.23 | 98.09 | 97.45 |
| KWM-T-6 | 0.4M | 97.49 | 97.00 | 98.17 | 97.22 |

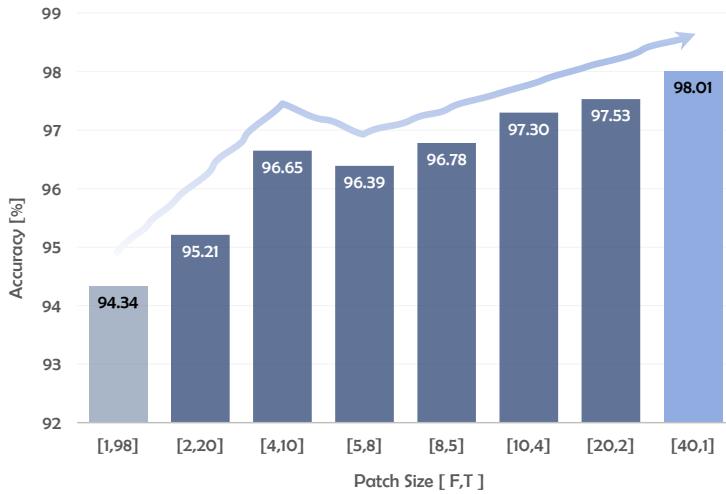


Figure 4: Impact of patch size on keyword spotting accuracy. Larger temporal patches significantly improve performance, suggesting that long-range temporal modeling is critical. Results are based on the Speech Commands V1-12 dataset using KWM-192.

5.4. Ablation Studies

We conduct three ablation studies to understand how different design choices affect the performance of Keyword Mamba. These studies focus on (1) the shape of MFCC patches, (2) the position of the class token, and (3) the directionality of the Mamba components.

Temporal vs. Frequency Modeling: First, we vary the shape of MFCC spectrogram patches fed into the model. Our baseline uses temporal-domain Mamba with patches shaped [40,1], but we also test frequency-domain Mamba and rectangular patches that mix both time and frequency dimensions.

As shown in Fig. 4, the temporal configuration achieves the highest accuracy, reaching 98.01%. This confirms that modeling along the time axis is most effective for keyword spotting. Our first projection layer (e.g., Patching) acts like a

Table 4

Ablation study on class token position and Mamba type. Mid-position tokens and bidirectional Mamba configurations (BiMamba-Bi-Bi) provide the best accuracy, highlighting the importance of token placement and directional flow.

| Ablation Objective | Type | V1-12 |
|----------------------|---------------|--------------|
| class token position | Mid | 98.01 |
| | Head | 97.75 |
| | End | 97.92 |
| mamba type | BiMamba-Bi-Bi | 98.01 |
| | BiMamba-Fo-Bi | 97.69 |
| | Mamba-Fo-Fo | 78.29 |

temporal convolution with a kernel size [40,1] and stride 1, which supports earlier findings that temporal convolutions are especially useful for speech recognition [9].

Class Token Position: Next, we study the effect of class token placement. We test inserting the token at the start, middle, or end of the sequence. Table 4 shows that placing the token in the middle of the input achieves the best result (98.01%), likely because it allows the token to receive information from both earlier and later parts of the sequence.

However, we also find that the class token position has a relatively small impact on performance in keyword spotting. This contrasts with other speech or language tasks, where token position often plays a larger role.

Directional Flow in Mamba Variants: Finally, we compare three types of Mamba configurations based on their directional flow in both the SSM module and Conv1D layers:

- BiMamba-Bi-Bi: bidirectional SSM and Conv1D
- BiMamba-Fo-Bi: only bidirectional SSM
- Mamba-Fo-Fo: unidirectional in both modules

Among the three, BiMamba-Bi-Bi achieves the best performance (98.01%), as shown in Table 4. In contrast, Mamba-Fo-Fo drops significantly to 78.29%, showing that unidirectional modeling lacks important context. These results confirm the importance of bidirectional information flow in speech tasks, where keyword boundaries and surrounding sounds may appear in both past and future contexts [23]. Our findings reinforce the idea that capturing both directions is essential for high-performance speech modeling.

6. Conclusions

In this work, we propose Keyword Mamba, a novel architecture that brings the Mamba framework into the KWS domain. By applying Mamba along the temporal axis, we effectively capture long-range dependencies while maintaining computational efficiency. To further explore its additional nonlinear potential, we integrate Mamba into the Transformer architecture by replacing only the multi-head self-attention (MHSA) module. Experiments we conducted on multiple versions of the Google Speech Commands datasets demonstrate the effectiveness and robustness of our approach. These results highlight the promising potential of Mamba-based architectures in outperforming traditional Transformers for the KWS task. The implementation of this work is available at the following GitHub repository: <https://github.com/dhyzy123/KWM>.

CRediT authorship contribution statement

Hanyu Ding: Writing - original draft, Writing - review & editing, Methodology, Investigation, Validation, Data curation, Software. **Wenlong Dong:** Writing - review & editing, Supervision, Formal analysis. **Qirong Mao:** Writing - review & editing, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No.62176106), the Special Scientific Research Project of the School of Emergency Management of Jiangsu University (No.KY-A-01), the Project of Faculty of Agricultural Engineering of Jiangsu University (No.NGXB20240101), the Key Project of National Natural Science Foundation of China (No.U1836220), and the Jiangsu Key Research and Development Plan Industry Foresight and Key Core Technology (No.BE2020036).

Data availability

Data will be made available on request.

References

- [1] López-Espejo, I., Tan, Z.H., Hansen, J.H., Jensen, J., 2021. Deep spoken keyword spotting: An overview. *IEEE Access* 10, 4169–4199.
- [2] Ng, D., Xiao, Y., Yip, J.Q., Yang, Z., Tian, B., Fu, Q., Chng, E.S., Ma, B., 2023. Small footprint multi-channel network for keyword spotting with centroid based awareness, in: *Interspeech 2023*, pp. 296–300.
- [3] Zhang, Y., Suda, N., Lai, L., Chandra, V., 2017. Hello edge: Keyword spotting on microcontrollers. *arXiv preprint arXiv:1711.07128*.
- [4] Dar, M.A., Pushparaj, J., 2025. Bi-directional lstm-based isolated spoken word recognition for kashmiri language utilizing mel-spectrogram feature. *Applied Acoustics* 231, 110505.
- [5] Peng, T., Xiao, Y., 2025. Dark experience for incremental keyword spotting, in: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE. pp. 1–5.
- [6] Xiao, Y., Peng, T., Das, R.K., Hu, Y., Zhuang, H., 2025. Analytickws: towards exemplar-free analytic class incremental learning for small-footprint keyword spotting, in: *Findings of the Association for Computational Linguistics ACL 2025*, p. 14147–14158.
- [7] Chen, G., Parada, C., Heigold, G., 2014. Small-footprint keyword spotting using deep neural networks, in: *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE. pp. 4087–4091.
- [8] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 2002. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324.
- [9] Choi, S., Seo, S., Shin, B., Byun, H., Kersner, M., Kim, B., Kim, D., Ha, S., 2019. Temporal convolution for real-time keyword spotting on mobile devices, in: *Interspeech 2019*, pp. 3372–3376.
- [10] Kim, B., Chang, S., Lee, J., Sung, D., 2021. Broadcasted residual learning for efficient keyword spotting, in: *Interspeech 2021*, pp. 4538–4542.
- [11] De Andrade, D.C., Leo, S., Viana, M.L.D.S., Bernkopf, C., 2018. A neural attention model for speech command recognition. *arXiv preprint arXiv:1808.08929*.
- [12] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation* 9, 1735–1780.
- [13] Rybakov, O., Kononenko, N., Subrahmany, N., Visontai, M., Laurenzo, S., 2020. Streaming keyword spotting on mobile devices, in: *Interspeech 2020*, pp. 2277–2281.
- [14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems* 30, 5998–6008.
- [15] Berg, A., O'Connor, M., Cruz, M.T., 2021. Keyword transformer: A self-attention model for keyword spotting, in: *Interspeech 2021*, pp. 4249–4253.
- [16] Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* 82, 35–45.
- [17] Gu, A., Goel, K., Re, C., 2022a. Efficiently modeling long sequences with structured state spaces, in: *International Conference on Learning Representations*.
- [18] Gu, A., Goel, K., Gupta, A., Ré, C., 2022b. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems* 35, 35971–35983.
- [19] Gu, A., Dao, T., 2024. Mamba: Linear-time sequence modeling with selective state spaces, in: *First Conference on Language Modeling*.
- [20] Zhang, X., Ma, J., Shahin, M., Ahmed, B., Epps, J., 2025. Rethinking mamba in speech processing by self-supervised models, in: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE. pp. 1–5.
- [21] Shi, Y., Dong, M., Xu, C., 2024. Multi-scale vmamba: Hierarchy in hierarchy visual state space model. *Advances in Neural Information Processing Systems* 37, 25687–25708.
- [22] Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X., 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model, in: *Proceedings of the 41st International Conference on Machine Learning*, PMLR. pp. 62429–62442.
- [23] Zhang, X., Zhang, Q., Liu, H., Xiao, T., Qian, X., Ahmed, B., Ambikairajah, E., Li, H., Epps, J., 2025. Mamba in speech: Towards an alternative to self-attention. *IEEE Transactions on Audio, Speech and Language Processing* 33, 1933–1948.
- [24] Xiao, Y., Das, R.K., 2025. Xlsr-mamba: A dual-column bidirectional state space model for spoofing attack detection. *IEEE Signal Processing Letters* 32, 1276–1280.
- [25] Xiao, Y., Das, R.K., 2024. Tf-mamba: A time-frequency network for sound source localization. *arXiv preprint arXiv:2409.05034*.
- [26] Jiang, X., Han, C., Mesgarani, N., 2025. Dual-path mamba: Short and long-term bidirectional selective structured state space models for speech separation, in: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE. pp. 1–5.
- [27] Warden, P., 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*.

- [28] Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C., 2020. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems* 33, 1474–1487.
- [29] Fu, D.Y., Epstein, E.L., Nguyen, E., Thomas, A.W., Zhang, M., Dao, T., Rudra, A., Ré, C., 2023. Simple hardware-efficient long convolutions for sequence modeling, in: International Conference on Machine Learning, PMLR. pp. 10373–10391.
- [30] Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al., 2022. In-context learning and induction heads. arXiv preprint arXiv:2209.11895 .
- [31] Liu, H., Dai, Z., So, D., Le, Q.V., 2021. Pay attention to mlps. *Advances in neural information processing systems* 34, 9204–9215.
- [32] Waleffe, R., Byeon, W., Riach, D., Norick, B., Korthikanti, V., Dao, T., Gu, A., Hatamizadeh, A., Singh, S., Narayanan, D., et al., 2024. An empirical study of mamba-based language models. arXiv preprint arXiv:2406.07887 .
- [33] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations.
- [34] Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y., 2024. Vmamba: Visual state space model. *Advances in neural information processing systems* 37, 103031–103063.
- [35] Li, S., Singh, H., Grover, A., 2024. Mamba-nd: Selective state space modeling for multi-dimensional data, in: European Conference on Computer Vision, Springer. pp. 75–92.
- [36] Pei, X., Huang, T., Xu, C., 2025. Efficientvmamba: Atrous selective scan for light weight visual mamba, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 6443–6451.
- [37] Zhan, Z., Kong, Z., Gong, Y., Wu, Y., Meng, Z., Zheng, H., Shen, X., Ioannidis, S., Niu, W., Zhao, P., et al., 2024. Exploring token pruning in vision state space models. *Advances in Neural Information Processing Systems* 37, 50952–50971.
- [38] Xie, F., Zhang, W., Wang, Z., Ma, C., 2024. Quadmamba: Learning quadtree-based selective scan for visual state space model. *Advances in Neural Information Processing Systems* 37, 117682–117707.
- [39] Erol, M.H., Senocak, A., Feng, J., Chung, J.S., 2024. Audio mamba: Bidirectional state space model for audio representation learning. *IEEE Signal Processing Letters* 31, 2975–2979.
- [40] Gong, Y., Lai, C.I., Chung, Y.A., Glass, J., 2022. Ssast: Self-supervised audio spectrogram transformer, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 10699–10709.
- [41] Chao, R., Cheng, W.H., La Quatra, M., Siniscalchi, S.M., Yang, C.H.H., Fu, S.W., Tsao, Y., 2024. An investigation of incorporating mamba for speech enhancement, in: 2024 IEEE Spoken Language Technology Workshop (SLT), IEEE. pp. 302–308.
- [42] Quan, C., Li, X., 2024. Multichannel long-term streaming neural speech enhancement for static and moving speakers. *IEEE Signal Processing Letters* 31, 2295–2299.
- [43] Gao, X., Chen, N.F., 2024. Speech-mamba: Long-context speech recognition with selective state spaces models, in: 2024 IEEE Spoken Language Technology Workshop (SLT), IEEE. pp. 1–8.
- [44] Masuyama, Y., Miyazaki, K., Murata, M., 2024. Mamba-based decoder-only approach with bidirectional speech modeling for speech recognition, in: 2024 IEEE Spoken Language Technology Workshop (SLT), IEEE. pp. 1–6.
- [45] Jiang, X., Li, Y.A., Florea, A.N., Han, C., Mesgarani, N., 2025. Speech slytherin: Examining the performance and efficiency of mamba for speech separation, recognition, and synthesis, in: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 1–5.
- [46] Dang, S., Matsumoto, T., Takeuchi, Y., Kudo, H., 2024. U-mamba-net: A highly efficient mamba-based u-net style network for noisy and reverberant speech separation, in: 2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE. pp. 1–5.
- [47] Han, D., Wang, Z., Xia, Z., Han, Y., Pu, Y., Ge, C., Song, J., Song, S., Zheng, B., Huang, G., 2024. Demystify mamba in vision: A linear attention perspective. *Advances in neural information processing systems* 37, 127181–127203.
- [48] Davis, S., Mermelstein, P., 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing* 28, 357–366.
- [49] Elfwing, S., Uchibe, E., Doya, K., 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks* 107, 3–11.
- [50] Park, D.S., Chan, W., Zhang, Y., Chiu, C.C., Zoph, B., Cubuk, E.D., Le, Q.V., 2019. Specaugment: A simple data augmentation method for automatic speech recognition, in: Interspeech 2019, pp. 2613–2617.
- [51] Majumdar, S., Ginsburg, B., 2020. Matchboxnet: 1d time-channel separable convolutional neural network architecture for speech commands recognition, in: Interspeech 2020, pp. 3356–3360.
- [52] Ng, D., Chen, Y., Tian, B., Fu, Q., Chng, E.S., 2022. Convmixer: Feature interactive convolution with curriculum learning for small footprint and noisy far-field keyword spotting, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 3603–3607.
- [53] Zeng, M., Xiao, N., 2019. Effective combination of densenet and bilstm for keyword spotting. *IEEE Access* 7, 10767–10775.
- [54] Mittermaier, S., Kürzinger, L., Waschnec, B., Rigoll, G., 2020. Small-footprint keyword spotting on raw audio data with sinc-convolutions, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 7454–7458.
- [55] Zhang, B., Li, W., Li, Q., Zhuang, W., Chu, X., Wang, Y., 2021. Autokws: Keyword spotting with differentiable architecture search, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 2830–2834.
- [56] Wang, L., Gu, R., Chen, N., Zou, Y., 2021. Text anchor based metric learning for small-footprint keyword spotting, in: Interspeech 2021, pp. 4219–4223.



Click here to access/download
LaTeX Source File
cas-common.sty



Click here to access/download
LaTeX Source File
cas-model2-names.bst



Click here to access/download
LaTeX Source File
cas-sc.cls



Click here to access/download
LaTeX Source File
README.md



Click here to access/download
LaTeX Source File
cas-dc.cls



Click here to access/download
LaTeX Source File
cas-refs.bib



Click here to access/download
LaTeX Source File
manuscript-double.tex





Click here to access/download
LaTeX Source File
kwm-ref.bib



Click here to access/download
LaTeX Source File
manuscript.bbl



Click here to access/download
LaTeX Source File
manuscript.tex



Click here to access/download
LaTeX Source File
cas-dc-templates.pdf



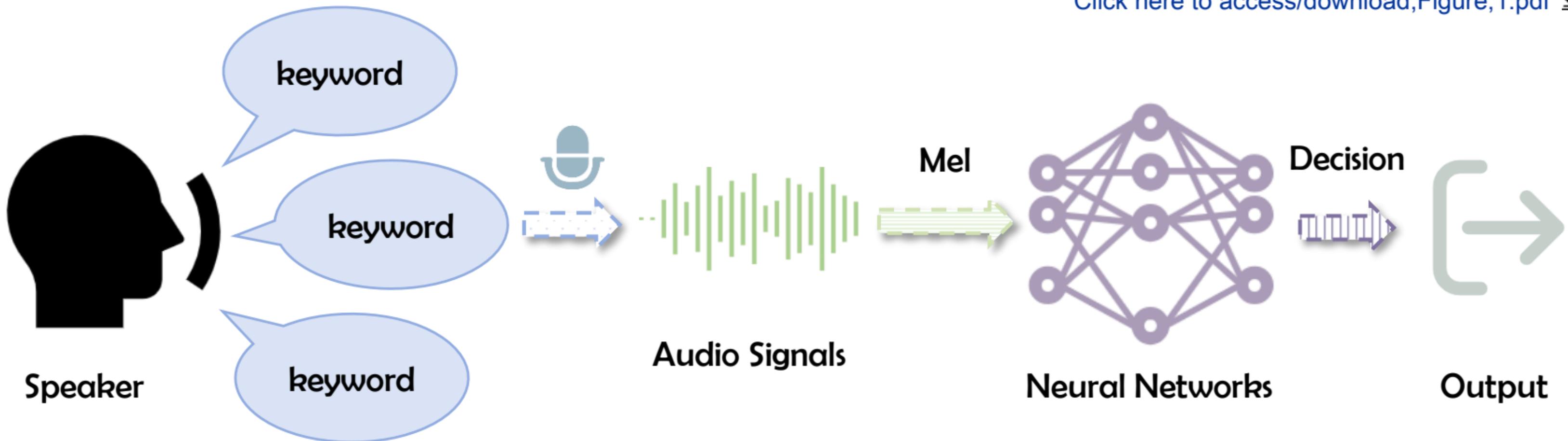


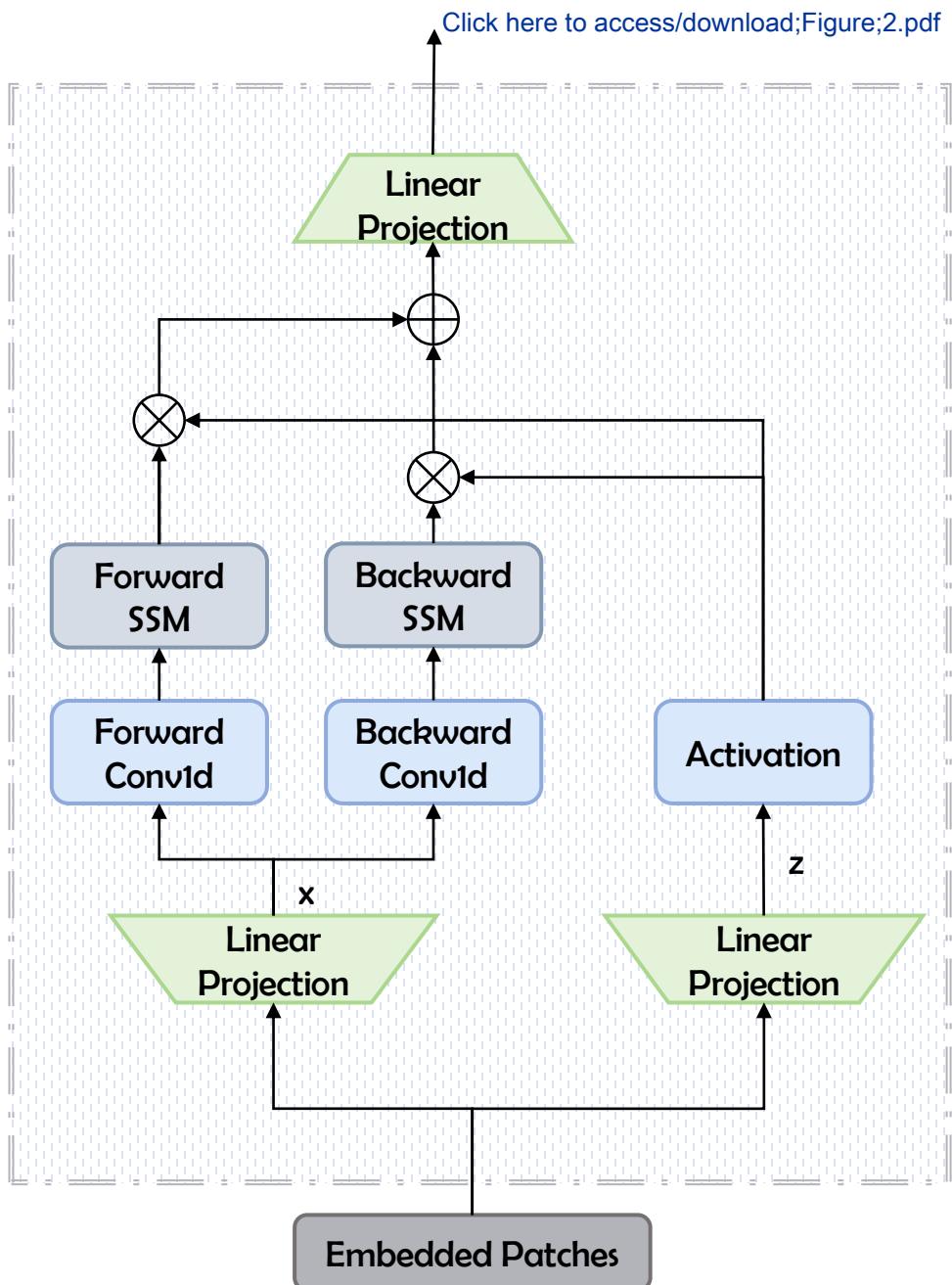
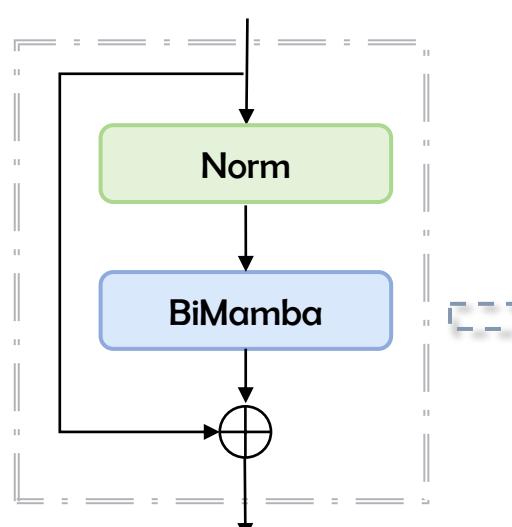
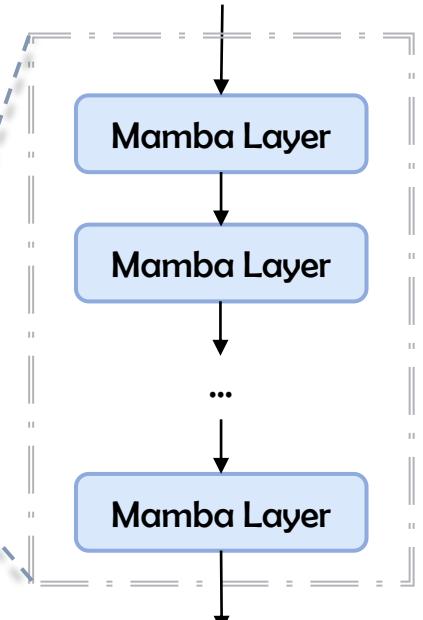
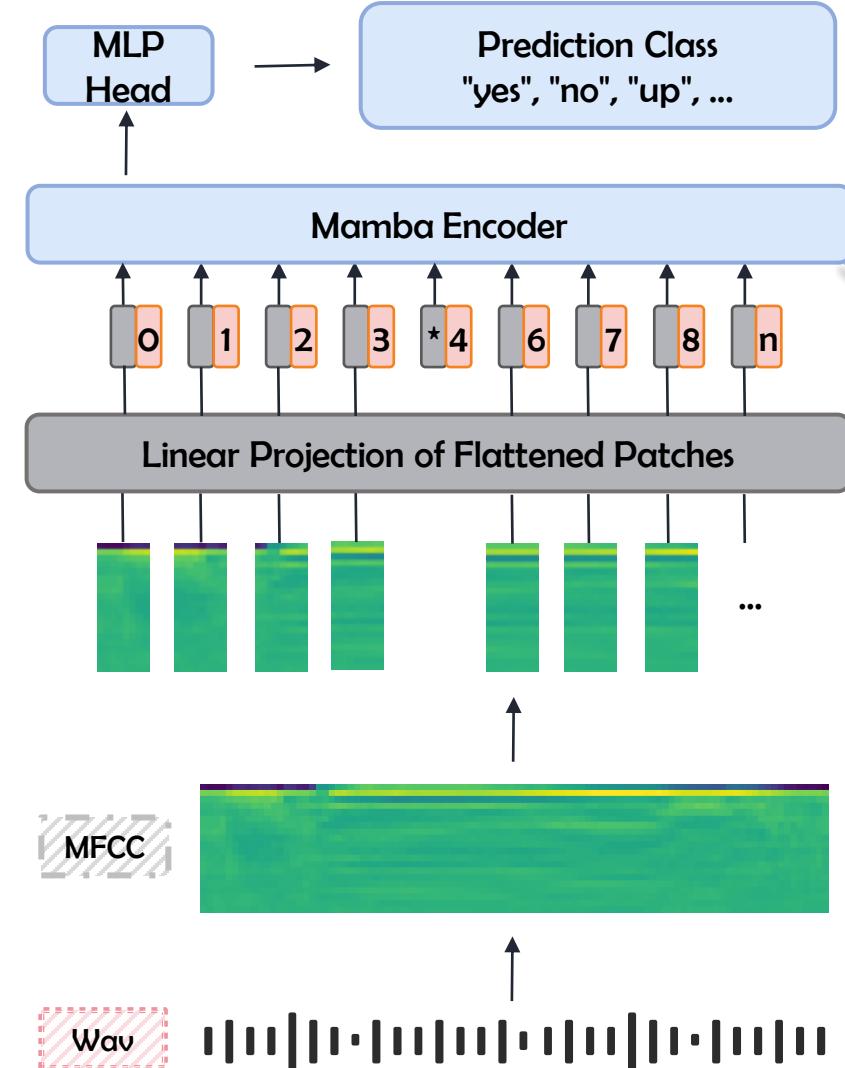
Click here to access/download
LaTeX Source File
image.jpeg



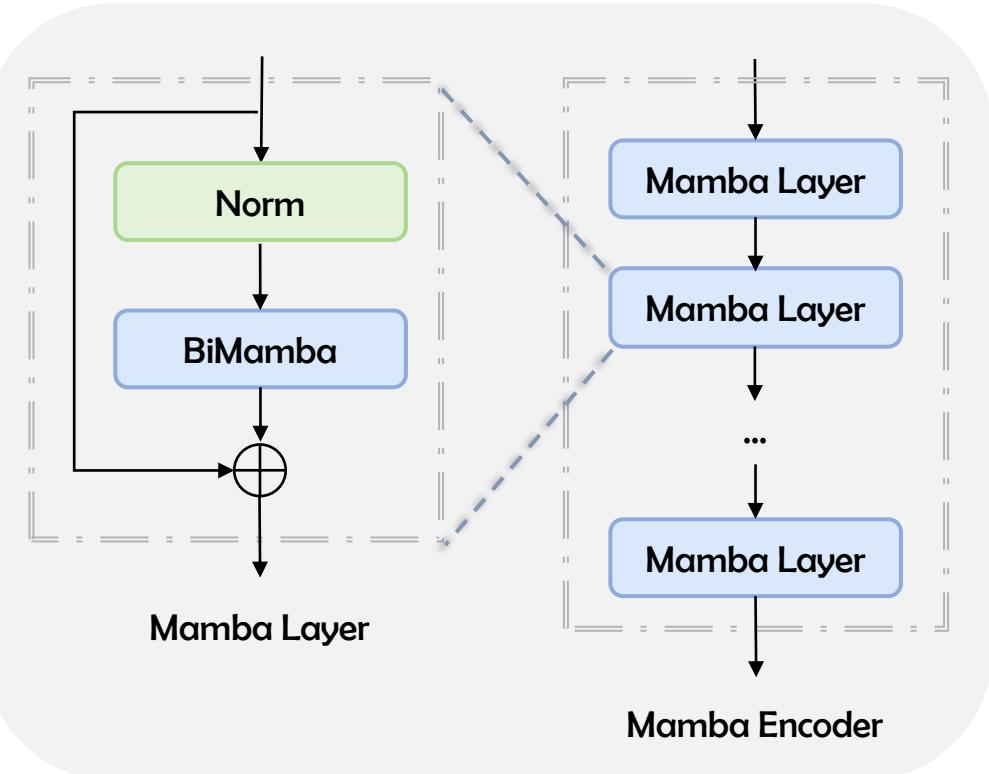


Click here to access/download
LaTeX Source File
cas-sc-template.pdf



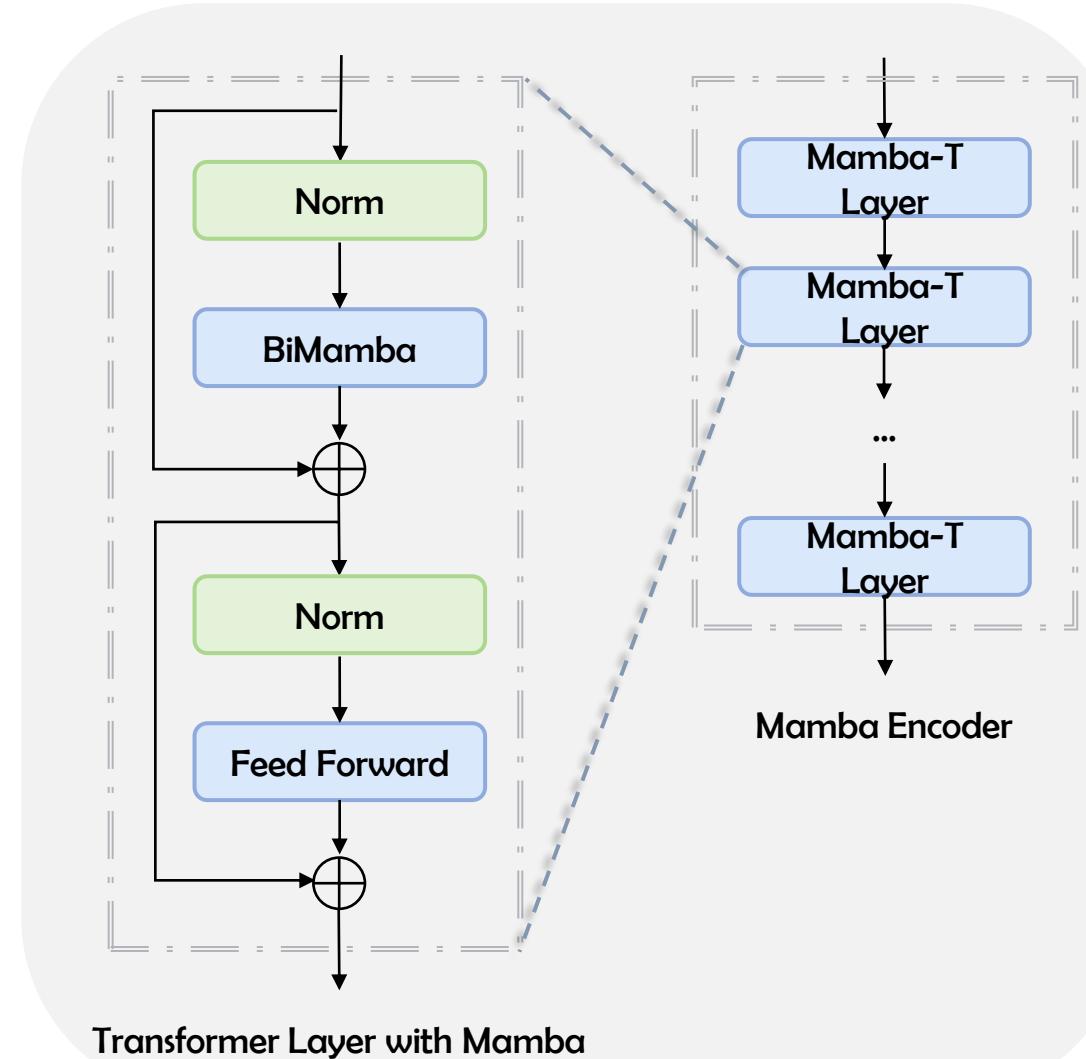


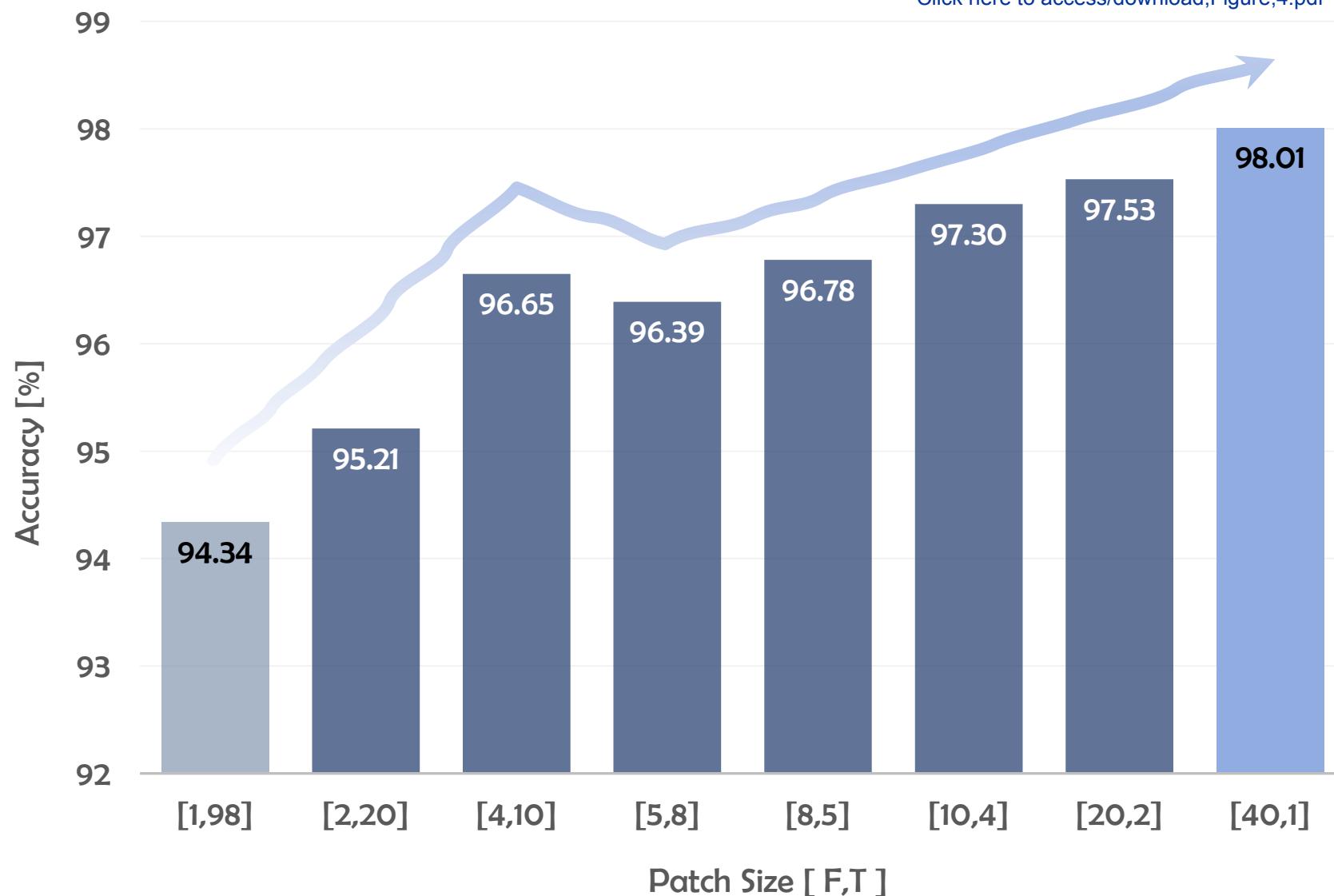
(a) Mamba Layer



Click here to access/download;Figure;3.pdf

(b) Transformer Layer with Mamba





Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Qirong Mao reports financial support was provided by National Natural Science Foundation of China.
Qirong Mao reports financial support was provided by Jiangsu University. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.