


Selective state models are what you need for animal action recognition

Edoardo Fazzari^{a,b,c} , Donato Romano^{a,b}, Fabrizio Falchi^{a,c}, Cesare Stefanini^{a,b}

^a The BioRobotics Institute, Sant'Anna School of Advanced Studies, Viale Rinaldo Piaggio, Pontedera, 56025, Tuscany, Italy

^b Department of Excellence in Robotics and AI, Sant'Anna School of Advanced Studies, Piazza Martiri della Libertà, Pisa, 56127, Tuscany, Italy

^c Institute of Information Science and Technologies, National Research Council of Italy, Via G. Moruzzi, Pisa, 56124, Tuscany, Italy

ARTICLE INFO

Keywords:

Animal action recognition
Deep learning
Selective state models
Computer vision
Mamba
Msqnet

ABSTRACT

Recognizing animal actions provides valuable insights into animal welfare, yielding crucial information for agricultural, ethological, and neuroscientific research. While video-based action recognition models have been applied to this task, current approaches often rely on computationally intensive Transformer layers, limiting their practical application in field settings such as farms and wildlife reserves. This study introduces Mamba-MSQNet, a novel architecture family for multilabel Animal Action Recognition using Selective Space Models. By transforming the state-of-the-art MSQNet model with Mamba blocks, we achieve significant reductions in computational requirements: up to 90% fewer Floating point Operations and 78% fewer parameters compared to MSQNet. These optimizations not only make the model more efficient but also enable it to outperform Transformer-based counterparts on the Animal Kingdom dataset, achieving a mean Average Precision of 74.6, marking an improvement over previous architectures. This combination of enhanced efficiency and improved performance represents a significant advancement in the field of animal action recognition. The dramatic reduction in computational demands, coupled with a performance boost, opens new possibilities for real-time animal behavior monitoring in resource-constrained environments. This enhanced efficiency could revolutionize how we observe and analyze animal behavior, potentially leading to breakthroughs in animal welfare assessment, behavioral studies, and conservation efforts.

1. Introduction

Understanding animal actions is fundamental to ethological research and has far-reaching implications across numerous disciplines, attracting increasing research attention in recent years (Fazzari et al., 2024; Ghosh and Dasgupta, 2022a). Today, recognizing animal actions is particularly significant for several key applications: animal behavior monitoring, neurological studies, bio-inspired robotic engineering.

While these applications share some methodologies, they differ significantly in their objectives. The first area primarily focuses on livestock in agriculture, where multiple or single animals are observed to analyze abnormal behaviors and evaluate their well-being. Recent research in this domain includes analysis of pig contact and biting behaviors (Odo et al., 2023; Alameer et al., 2022), monitoring feeding habits (Kavlak et al., 2023; Ollagnier et al., 2023), and identification of illness-related movements (Mei et al., 2023), researches that are complementary to general advancement in computer vision for precision agriculture (Li et al., 2024a; Luo et al., 2021). Neurological studies typically target smaller animals, mainly mice, to investigate

diseases such as Alzheimer's (Sutoko et al., 2021). The popularity of this field has led researchers to develop numerous tools for analyzing neural, video, and tracking data to extract meaningful features correlating animal movements with behaviors (Mathis et al., 2018; Pereira et al., 2022; Luxem et al., 2022; Segalin et al., 2021). Bio-inspired robot engineering aims to transpose animal movements into physical robotic animal (Manduca et al., 2024; Peng et al., 2020). The common thread across all these applications is the need to discretize and recognize actions, with subsequent uses tailored to each specific domain. However, the majority of these operations occur post hoc, after data collection from the animals. The collected data is typically processed using machine learning algorithms, statistical methods, or a combination of both. This method can introduce latency between observation and insight. Alternatively, real-time analysis involves immediate processing, often facilitated by human operators making on-the-spot decisions. While this approach reduces latency, it may be limited by human cognitive capacity and prone to inconsistencies (Tjandrasuwita et al., 2021).

* Corresponding author at: The BioRobotics Institute, Sant'Anna School of Advanced Studies, Viale Rinaldo Piaggio, Pontedera, 56025, Tuscany, Italy.
E-mail addresses: edoardo.fazzari@santannapisa.it (E. Fazzari), donato.romano@santannapisa.it (D. Romano), fabrizio.falchi@cnr.it (F. Falchi), cesare.stefanini@santannapisa.it (C. Stefanini).

URL: <https://www.santannapisa.it/en/edoardo-fazzari> (E. Fazzari).

<https://doi.org/10.1016/j.ecoinf.2024.102955>

Received 7 November 2024; Received in revised form 7 December 2024; Accepted 12 December 2024

Available online 19 December 2024

1574-9541/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Machine Learning (ML), particularly Deep Learning (DL), has significantly advanced research in Animal Action Recognition (AAR) (Kleanthous et al., 2022). DL models have been applied to various modalities, including sensory data, images, videos, and multi-modal fusion approaches. For processing sensor data (e.g., GPS, accelerometers, and gyroscopes), Multi-Layer Perceptrons (Arablouei et al., 2023), Recurrent Neural Networks (Dang et al., 2022), and Convolutional Neural Networks (Pan et al., 2023) are commonly employed. Image-based approaches primarily utilize object detection techniques like YOLO (Manoharan, 2020; Riekert et al., 2020) or segmentation strategies (Xiao et al., 2023) for generating useful features, leading to the development of specialized tools such as EthoFlow (Bernardes et al., 2021). However, sensor data and static images have limitations in capturing comprehensive animal movement information. Sensor data lack skeletal details, while images fail to capture motion dynamics. Video, in contrast, provides both physiological characteristics and sequential movement information. To address the need for diverse animal datasets, several video collections have been proposed (Feng et al., 2021; Liu et al., 2020; Rahman et al., 2014). Recently, the Animal Kingdom dataset (Ng et al., 2022) was introduced as the largest publicly available multilabel action recognition dataset for animals. The current state-of-the-art model for this dataset, in terms of mean Average Precision (mAP), is the Multi-modal Semantic Query Network (MSQNet) (Mondal et al., 2023).

MSQNet is a multi-modal fusion model built on the Transformer architecture (Dosovitskiy et al., 2020). It comprises three key components: (1) a spatio-temporal transformer video encoder, (2) a multi-modal query encoder that integrates information from both video and action class-specific sources, and (3) a multi-modal decoder utilizing multi-headed self-attention and encoder–decoder attention mechanisms to process the video encoding. While this architecture demonstrates good capabilities, its complexity results in significant computational demands. Consequently, MSQNet’s resource-intensive nature may limit its practical application in real-world scenarios where computational resources are constrained due to the use of Transformers. Consequently, MSQNet’s resource-intensive nature may limit its practical application in real-world scenarios where computational resources are constrained due to the inherent limitations of Transformer architectures. Transformers fundamentally suffer from quadratic computational complexity, primarily arising from their self-attention mechanism, which requires pairwise comparisons between all sequence tokens (Vaswani, 2017). This architectural design creates significant computational bottlenecks: processing long sequences becomes exponentially more expensive, memory consumption increases dramatically, and inference speed becomes prohibitively slow, especially when dealing with high-resolution or extensive data inputs. These computational inefficiencies make Transformer-based models challenging to deploy in resource-constrained environments such as edge devices, wildlife monitoring systems, or mobile applications (Lu et al., 2024).

To reduce their computational demands while maintaining their performance and ability to model sequential data, Linear State-Space Layers (LSSLs), later known as State Space Models (SSMs), were introduced (Gu et al., 2021b).

LSSLs map a 1-dimensional function or sequence $u(t) \rightarrow y(t)$ through an implicit state $x(t)$ by simulating a linear continuous-time state-space representation in discrete-time. Theoretically, LSSLs combine the strengths of Recurrent Neural Networks, Convolutional Neural Networks, and Neural Differential Equations, being simultaneously recurrent, convolutional, and continuous-time. Efficient implementation requires careful selection of A . Structured State Spaces (S4) (Gu et al., 2021a) condition A with a low-rank correction, enabling stable diagonalization and reducing the SSM to a Cauchy kernel computation. Diagonal State Space (DSS) (Gupta et al., 2022) demonstrated that a fully diagonal state matrix can preserve S4’s performance. S4D (Gu et al., 2022) later introduced a diagonal SSM combining S4’s computation and parameterization with DSS’s initialization, resulting in

a simpler method. Despite these improvements, SSMs initially underperformed attention in language modeling and were slower than Transformers due to poor hardware utilization. H3 (Fu et al., 2022) addressed these issues, which was further improved by Mamba (Gu and Dao, 2023). Mamba incorporates a selection mechanism with parallel scan, enabling the architecture to surpass Transformers in processing long sequences.

Mamba’s rapid success in Natural Language Processing quickly expanded to Computer Vision, mirroring the trajectory of Transformers. Vision Mamba (Vim) (Zhu et al., 2024) employs bidirectional Mamba layers for data-dependent global visual context modeling and position embedding for location-aware visual understanding, achieving performance comparable to Vision Transformers. VideoMamba (Li et al., 2024b) extended this approach to video processing, handling sequential frame information for single- and multi-modality video tasks.

We propose integrating Mamba (Gu and Dao, 2023) into MSQNet (Mondal et al., 2023) to address critical computational limitations in animal action recognition. Our specific research objectives are threefold:

- Develop a more computationally efficient architecture for animal behavior analysis by replacing the existing spatio-temporal video encoder and multi-modal decoder with Mamba-based components.
- Demonstrate the potential of State Space Models in reducing model complexity without compromising performance in animal action recognition.
- Introduce **Mamba-MSQNet**, a *novel* family of architectures with configurable dimensions and Mamba block configurations that offer improved computational efficiency.

The proposed integration involves replacing the original encoder with a VideoMamba (Li et al., 2024b) encoder and implementing Mamba blocks for information processing. We have developed multiple configurations varying in video encoder dimensions (tiny, small, medium), pretraining strategies, and the number of Mamba Blocks used for information fusion. Critically, our approach not only reduces computational complexity in terms of model parameters and Floating point Operations (FLOPs), but also demonstrates performance comparable to or potentially exceeding the original MSQNet architecture. To our knowledge, this represents the first systematic application of Mamba architectures to Animal Action Recognition, presenting a significant step towards more efficient and rapid behavioral inference in ecological and animal welfare research.

2. Materials and methods

2.1. Dataset

Our models for recognizing actions in animal videos were evaluated using the Animal Kingdom dataset (Ng et al., 2022), which is currently the largest publicly available dataset for Animal Action Recognition (AAR). This dataset comprises 30,100 video clips, totaling 50 h of footage, divided into training and validation sets of 24,004 and 6096 clips, respectively. The complexity of the task is heightened by the multi-label nature of the videos, where multiple animals and actions can occur simultaneously. The dataset encompasses 140 distinct actions, categorized into sixteen groups: affection, aggression, communication, death, defense, feeding, general behavior, life cycle, maintenance, movement, predation, resting, sexual behavior, shelter-related actions, social interactions, and transport. The animal subjects represent a diverse range of 850 unique species, including mammals, reptiles, amphibians, birds, fish, and insects, providing a comprehensive spectrum of faunal behavior.

The dataset exhibits significant variability in the number of frames per clip, ranging from 2 to 2797 frames, with a mean of 146.5 and a

standard deviation of 168.8. This diversity is crucial to consider during training, as we select a subset of frames from each clip based on the employed video encoder. The frame selection process depends on the relationship between the clip length (l_c) and the video encoder's frame capacity (l_e). When $l_c \leq l_e$, we sample frames at equal intervals across the clip's duration. Conversely, when $l_c > l_e$, we first generate $l_e + 1$ equally spaced indexes (*linear_selection*), then adjust these indexes using the following equation:

$$\text{indexes} = \text{linear_selection}[1 :] - \frac{\text{linear_selection}[1 :] - \text{linear_selection}[: -1]}{2} \quad (1)$$

This adjustment positions the selected frames at the midpoint between the initial linear selections, ensuring a balanced representation of the clip's content. For each clip we selected 16 frames, which is the input for our networks.

2.2. Data augmentation

Data augmentation was applied to both training and validation sets. For the validation set, the process involved resizing video images while preserving aspect ratios, followed by center cropping to a fixed size. The images from each video were then stacked into a single tensor, converted to a PyTorch tensor, and normalized using predefined mean and standard deviation values. While this validation set augmentation primarily aimed to match the model's input requirements, a more extensive augmentation process was implemented for the training set.

For the training set, a more extensive augmentation process was applied. First, a multi-scale cropping was performed by cropping the images in a video using random scales selected from a list (100%, 87.5%, 75%, and 66% of the original size). Next, the images were horizontally flipped with a probability of 50%. Following this, a random color jitter operation was applied with an 80% probability, adjusting brightness by up to 40%, contrast by 40%, saturation by 20%, and hue by 10%. Additionally, a random grayscale effect was applied with a 20% probability. Subsequently, the images from the video were stacked into a single tensor. These images were then converted into a PyTorch tensor and normalized for pixel values. Finally, the images were normalized using predefined mean and standard deviation values.

2.3. Mamba-MSQNet architectures

Mamba-MSQNet represents a family of architectures sharing common structures but differing in their pretrained VideoMamba (Li et al., 2024b) models and the number of Mamba blocks at the network's end. This approach simplifies MSQNet (Mondal et al., 2023) by replacing Transformer-based components with Mamba-based alternatives. Two key substitutions are made: (1) the spatio-temporal transformer video encoder is replaced with a spatio-temporal Mamba video encoder, and (2) the multi-modal decoder's multi-head self-attention and encoder-decoder attention mechanisms are substituted with a combination of Multi-Layer Perceptron and stacked Mamba blocks.

For the spatio-temporal video encoder, we replaced MSQNet's TimeSformer model (pretrained on K400 Kay et al., 2017) (Bertasius et al., 2021b) with VideoMamba models pretrained on K400 and SthSthV2 (Goyal et al., 2017), following initial pretraining on ImageNet-1k (Deng et al., 2009) or using a mask strategy, as mentioned in Table 2. VideoMamba's three configurations (medium, small, and tiny) generate embeddings of 576, 384, and 192 dimensions, respectively. VideoMamba-S and -M models outperformed transformer-based architectures on K400 and SthSthV2 datasets, establishing themselves as compelling replacements for TimeSformer thanks to their superior ability to capture spatio-temporal pattern more efficiently by using Mamba blocks (Li et al., 2024b). Our experiments utilized all available VideoMamba configurations with 16 frames, consistent with MSQNet's optimal setup.

Table 1

Comparison in terms of floating point operations (FLOPs) and parameters between MSQNet (Mondal et al., 2023) and our Mamba implementations. M, S, Ti refers to the different VideoMamba backbones employed, middle, small, tiny, respectively.

Model	# Mamba Blocks	FLOPs (T)	PARAM (M)
MSQNet (Mondal et al., 2023)	–	0.677	252
Mamba-M-MSQNet	16	0.487	190
Mamba-M-MSQNet	8	0.485	176
Mamba-M-MSQNet	4	0.484	166
Mamba-S-MSQNet	16	0.353	142
Mamba-S-MSQNet	8	0.351	127
Mamba-S-MSQNet	4	0.350	120
Mamba-Ti-MSQNet	16	0.303	123
Mamba-Ti-MSQNet	8	0.301	108
Mamba-Ti-MSQNet	4	0.300	101

To integrate the query required for the transformer decoder with information from the global encoder and substitute this decoder, we implemented a two-stage fusion process. First, an MLP concatenates and fuses the query with the encoder output, providing an initial layer of information integration. This fused representation is then fed into a series of Mamba blocks for further processing. Following the recommendation of the Mamba authors (Gu and Dao, 2023), we experimented with doubling the number of Mamba blocks relative to the original transformer heads, which was eight. Specifically, we tested configurations with 16, 8, and 4 Mamba blocks to evaluate how performance varies with block count. The details of the architecture are shown in Fig. 1.

Our architecture demonstrates significant reductions in floating-point operations and model parameters. The 16-block versions show reductions of 29%, 48%, and 55% in FLOPs for VideoMamba-M-MSQNet, VideoMamba-S-MSQNet, and VideoMamba-Ti-MSQNet, respectively, compared to MSQNet. Excluding the unmodified CLIP image encoder (0.261T FLOPs), the reductions increase to 48%, 77%, and 90%. Similarly, parameter counts are substantially reduced: 25% (38% without CLIP) for VideoMamba-M-MSQNet, 43% (66% without CLIP) for VideoMamba-S-MSQNet, and 51% (78% without CLIP) for VideoMamba-Ti-MSQNet.

2.4. Training and validation

The training process was conducted in two phases: initially, the video encoder for each model kept frozen for 100 epochs, followed by 150 epochs with the video encoder weights trainable. This strategy was implemented to prevent the propagation of not-ideal gradients caused by the initial random initialization of the other layers, which can lead to feature distortion (Kumar et al., 2022). Through the training, model validation was performed every 5 epochs using the validation set, reporting the mean Average Precision (mAP). The model used Binary Cross Entropy with Logits (BCEWithLogitsLoss) as the loss function, which is particularly effective for multilabel recognition tasks. This is because it treats each label independently, enabling the model to estimate the probability of multiple positive labels simultaneously. For optimization, AdamW (Adam with weight decay regularization) (Loshchilov and Hutter, 2017) was chosen, as it addresses the limitations of standard Adam by decoupling weight decay from the learning rate, thereby improving regularization and generalization performance. AdamW was employed with a learning rate of 10^{-4} , β_1 of 0.9, controlling the decay rate for the first moment estimates, β_2 of 0.95, controlling the decay rate for the second moment estimates, and a weight decay of 0.1. Additionally, a scheduler was employed based on cosine annealing with warm restarts (Loshchilov and Hutter, 2016) with 10 interactions before a restart. The batch size was set to 8 for all models, except for Mamba-Ti-MSQNet implementations, which used a batch size of 16.

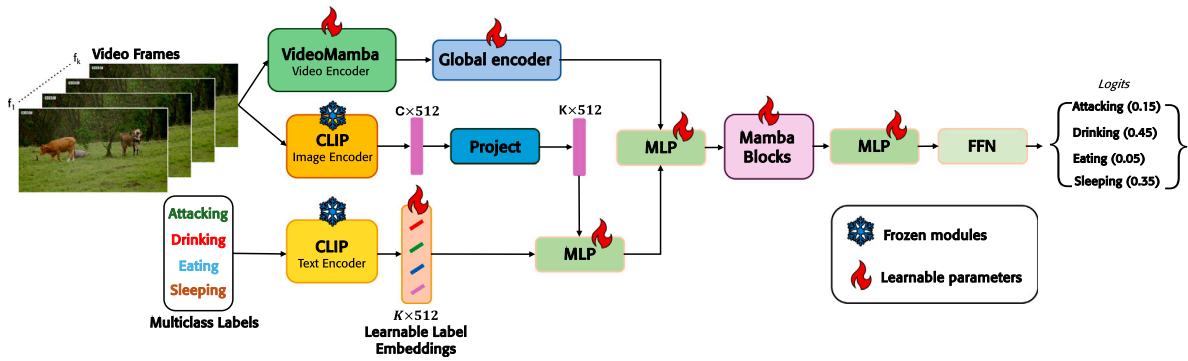


Fig. 1. Mamba-MSQNet architecture. FLOPs are considered for processing a single sample made of 16 frames.

2.5. Software and tools

All neural networks were trained on an NVIDIA A100 GPU using Python and PyTorch. The training environment was set up using the pytorch/pytorch:2.2.2-cuda12.1-cudnn8-devel Docker container, with additional PyTorch utilities for vision and augmentation installed (detailed dependencies are available in our GitHub repository). For employing VideoMamba (Li et al., 2024b), we installed a specific version of Mamba provided within the VideoMamba package, as the original Mamba version is incompatible with VideoMamba’s requirements.

Given that the NVIDIA A100 GPU used for training is better suited for high-performance server environments, rather than real-world, resource-constrained devices, we tested inference times on an NVIDIA GTX 1650 GPU. This allowed us to better assess the model’s potential for future in-field applications.

3. Results

We evaluated our models using mean Average Precision (mAP), a metric previously employed by Ng et al. (2022) and Mondal et al. (2023) for this dataset. Table 2 presents the performance of our Mamba-MSQNet implementations alongside CARE-X3D, the best model proposed with the Animal Kingdom dataset (Ng et al., 2022), and MSQNet. The results are categorized by backbone type, Mamba layers, and pretraining strategies.

Our top-performing model, Mamba-Ti-MSQNet (16 layers, ImageNet-1K pretrained, K400 training), achieves a mAP of 74.6, slightly surpassing MSQNet while significantly reducing parameter count and FLOPs as displayed in Table 1. Several insights emerge from the results: (1) all but three of our implementations matched or exceeded MSQNet’s mAP score, demonstrating the robustness and potential of our proposed approach; (2) smaller backbones generally yielded superior performance, possibly due to reduced network complexity and improved generalizability; (3) ImageNet-1K pre-training proved more effective than MASK pretraining, highlighting the importance of foundational visual representation learning; and (4) increasing the number of Mamba blocks systematically enhanced network performance by enabling deeper and more refined information processing.

The previous evaluations relied solely on pretrained VideoMamba backbones. In order to assess the model’s adaptability, we also examined the performance of Mamba-MSQNet without this pretraining. We trained the three configurations of Mamba-MSQNet with 16 Mamba blocks over 250 epochs. The results showed a 2–3 point decrease in mAP compared to the models with pretrained backbones. This underscores the benefits of utilizing a pretrained backbone to enhance the overall performance of the Mamba-MSQNet model, as was previously done by MSQNet.

To gain deeper insight into the similarities and differences between our best-performing architecture and MSQNet, we conducted a comparative analysis of their predictive capabilities. We examined the

Table 2

Results in terms of mean Average Precision (mAP). VidEnc stands for Video Encoder. The ‘Pretrained On’ columns refer to the initial pretrained before the actual trained of the network on the ‘Trained On’ task. The best result is the one in bold.

Model	# Mamba layers	VidEnc pretrained on	VidEnc trained on	mAP
CARE-X3D	–	–	–	25.2
MSQNet	–	–	K400	73.1
Mamba-M-MSQNet	16	–	–	
Mamba-S-MSQNet	16	–	–	71.3
Mamba-Ti-MSQNet	16	–	–	71.5
Mamba-M-MSQNet	16	MASK	K400	73.1
Mamba-M-MSQNet	8	MASK	K400	73.0
Mamba-M-MSQNet	4	MASK	K400	72.9
Mamba-M-MSQNet	16	ImageNet-1K	K400	74.2
Mamba-M-MSQNet	8	ImageNet-1K	K400	73.8
Mamba-M-MSQNet	4	ImageNet-1K	K400	73.2
Mamba-S-MSQNet	16	ImageNet-1K	K400	73.7
Mamba-S-MSQNet	8	ImageNet-1K	K400	73.4
Mamba-S-MSQNet	4	ImageNet-1K	K400	73.4
Mamba-Ti-MSQNet	16	ImageNet-1K	K400	74.6
Mamba-Ti-MSQNet	8	ImageNet-1K	K400	74.1
Mamba-Ti-MSQNet	4	ImageNet-1K	K400	73.1
Mamba-M-MSQNet	16	MASK	SthSthV2	73.7
Mamba-M-MSQNet	8	MASK	SthSthV2	73.6
Mamba-M-MSQNet	4	MASK	SthSthV2	73.0
Mamba-M-MSQNet	16	ImageNet-1K	SthSthV2	74.0
Mamba-M-MSQNet	8	ImageNet-1K	SthSthV2	73.6
Mamba-M-MSQNet	4	ImageNet-1K	SthSthV2	73.1
Mamba-S-MSQNet	16	ImageNet-1K	SthSthV2	73.6
Mamba-S-MSQNet	8	ImageNet-1K	SthSthV2	73.4
Mamba-S-MSQNet	4	ImageNet-1K	SthSthV2	73.0
Mamba-Ti-MSQNet	16	ImageNet-1K	SthSthV2	74.4
Mamba-Ti-MSQNet	8	ImageNet-1K	SthSthV2	74.3
Mamba-Ti-MSQNet	4	ImageNet-1K	SthSthV2	74.1

Multilabel Area Under the Curve (AUC) for both MSQNet (Fig. 2) and Mamba-Ti-MSQNet (our best implementation, Fig. 3), focusing on individual action categories. This analysis allowed us to identify specific strengths and weaknesses in each model’s performance across different types of actions. Numerical results for each action are specified in Table 3.

- **Affection:** Mamba-Ti-MSQNet demonstrated superior performance compared to MSQNet in the Affection category, achieving higher Area Under the Curve (AUC) scores for two specific actions: *Hugging* (ID 65) and *Showing Affection* (ID 106). The improvement was substantial, with an increase of approximately 0.2 in AUC for both actions. It is worth noting that the action *Holding Hands* (63) was not represented in the test set.
- **Aggressive:** The Aggressive category comprises 13 distinct actions. Mamba-Ti-MSQNet demonstrated performance comparable to MSQNet for several actions, including *Attacking* (ID 1), *Chasing* (14), *Coiling* (17), *Disturbing other animal* (27), *Fighting* (47),

and *Spitting venom* (112). Notably, both models failed to recognize the *Coiling* action, which had a single sample in the test set, resulting in an AUC of 0 for both. Mamba-Ti-MSQNet showed substantial improvements in certain actions. For *Competing for dominance* (18), the AUC increased from 0.069 to 0.533, while *Wrapping itself around prey* (137) saw an improvement from 0.064 to 0.708. A modest increase was observed for *Preying* (91), with the AUC rising from 0.132 to 0.304. However, MSQNet marginally outperformed Mamba-Ti-MSQNet in three actions: *Hissing* (62), where MSQNet achieved an AUC of 0.856 compared to Mamba-Ti-MSQNet's 0.601; *Rattling* (94), decreasing from 0.239 to 0.175; and *Wrapping prey*, declining from 0.176 to 0.039. The action *Pounding* (89) had no entries in the test set, precluding comparative analysis for this particular action.

- **Communication:** Mamba-Ti-MSQNet demonstrated superior performance across the majority of actions within this category. Notably, our model exhibited improved results for several actions, including *Barking* (3), *Calling* (10), *Giving off light* (56), and *Waving* (136). The latter two actions, each represented by a single sample in the test set, were particularly noteworthy. While MSQNet failed to detect these actions, resulting in an AUC of 0.0, Mamba-Ti-MSQNet achieved perfect detection with an AUC of 1.0. For the action *Chirping* (15), both MSQNet and our model demonstrated equivalent performance.
- **Death:** Regarding mortality-related actions, Mamba-Ti-MSQNet demonstrated varying degrees of improvement. For the action *Dying* (39), our model showed a marginal enhancement in performance. However, for the action *Dead* (21), Mamba-Ti-MSQNet achieved a substantial improvement, with an Area AUC of 0.851 compared to MSQNet's 0.505.
- **Defensive:** Our best architecture demonstrated performance comparable to MSQNet for the majority of actions, including *Camouflaging* (11), *Displaying Defensive Pose* (26), *Escaping* (42), *Fleeing* (51), *Retaliating* (96), and *Retreating* (97). However, Mamba-Ti-MSQNet exhibited superior performance in detecting *Defensive rearing* (23), which has the highest increase (from 0.068 to 0.246), *Standing in alert* (117), and *Struggling* (120). Conversely, it underperformed compared to MSQNet in identifying *Doing a back kick* (29).
- **Feeding:** The actions classified in this category (8, 38, 40, 80, and 105) exhibited comparable performance across both models. In particular, Mamba-Ti-MSQNet demonstrated superior performance in feeding-related actions, consistently outperforming MSQNet. These results underscore the model's enhanced capability to recognize and interpret complex interactions involving food and nutritional activities.
- **General:** Comparable results were observed for most actions, including "Flapping its ears" (49), "Keeping still" (68), "Panting" (79), "Startled" (118), and "Stinging" (119). A notable exception was "Lying on top" (75), where Mamba-Ti-MSQNet significantly improved the AUC from 0.02 to 0.19. It is worth noting that three actions in this category—"Gasping for air" (53), "Spitting" (111), and "Tail swishing" (126)—had no entries in the test set.
- **Life cycle:** Our model shows particularly strong performance in identifying *Exiting cocoon* (43), *Giving birth* (55), and *Laying eggs* (71), with significantly higher mAP scores than MSQNet. Additionally, it was able to identify the only *Undergoing chrysalis* (129) entry. No identification was possible for the simple entry for *Hatching* (60), as for MSQNet. Also in this category an action has no entries for the test set, the action is *Molting* (77),
- **Maintenance:** This category comprises 13 actions, for which the performance of MSQNet and Mamba-Ti-MSQNet varies. In several cases, the two networks yield very similar results, specifically for *Defecating* (22), *Doing a face dip* (32), *Grooming* (58), *Preening* (90), and *Shaking head* (104). Our model performs slightly worse

in some actions, including *Doing a chin dip* (31), *Performing allo-preening* (82), *Shaking* (103), and *Washing* (135). However, it demonstrates superior performance in *Performing allo-grooming* (81), *Rubbing its head* (99), *Licking* (73), and *Urinating* (132). Notably, for the latter two actions, Mamba-Ti-MSQNet significantly outperforms the standard version, which exhibits an AUC close to or equal to zero.

- **Movement:** This category, comprising 46 actions, is the largest in the dataset. For actions with substantial test entries, Mamba generally outperforms MSQNet. Notable improvements are observed in *Digging* (25), *Doing a backward tilt* (30), *Immobilized* (66), *Running on water* (101), *Standing* (116), and several others (33, 37, 44, 74, 98). Mamba-Ti-MSQNet also demonstrates significant improvements in actions with fewer entries, such as *Learning* (72), *Pulling* (93), *Sinking* (107), *Squatting* (115), and *Swaying* (122). However, MSQNet performs better in some cases, including *Falling* (46), *Flying* (52), *Sitting* (108), and others (48, 57, 64). It is noteworthy that five actions (*Detaching as a parasite* (24), *Doing a side tilt* (34), *Doing somersault* (36), *Lying down* (70), and *Spreading* (113)) had no test entries, limiting our ability to compare performance for these specific actions. The remaining actions in this category, such as *Jumping* (67), *Swimming* (123), and *Walking* (133), along with others (0, 16, 28, 41, 50, 59, 69, 76, 78, 100, 121, 125, 128, 130, 131, 134), show very similar AUC between our model and the non-Mamba implementation.
- **Prey:** In this category, Mamba-Ti-MSQNet consistently outperformed MSQNet, particularly for challenging actions. For instance, **Playing dead** (88) improved from 0.002 to 0.551, while **Trapped** (127) increased from 0.017 to 0.335. Though more modest, another improvement was also observed for **Being eaten** (7). These results suggest that the proposed model is more adept at recognizing subtle and complex prey-related behaviors.
- **Resting:** For most of the actions in this category the two models achieves the same results, specifically for *Resting* (95), *Sleeping* (109), *Sleeping in its nest* (110). However, Mamba-Ti-SQNet has an increment of 0.2 for *Yawning* (139).
- **Sensing:** Similar AUC results were obtained for *Attending* (2), *Exploring* (45), *Having a Flehmen response* (61), *Sensing* (102) actions.
- **Sexual:** Mamba-Ti-MSQNet demonstrated comparable or superior performance to MSQNet in the Sexual category. Actions such as *Dancing on water* (20), *Performing sexual display* (84), and *Performing sexual exploration* (85) showed similar results in terms of AUC. Interestingly, *Puffing its throat* (92), *Dancing* (19), *Doing push-ups* (35), and *Performing copulation mounting* (83) exhibited a significant increase in precision. However, *Performing sexual pursuit* (86) did not have any entries.
- **Shelter:** This category has only one action, *Building nest* (9), for which the two models have similar results, although ours performed slightly better.
- **Social:** Mamba-Ti-MSQNet achieves a 0.2 increase in accuracy for the *Playing* (87) action. Moreover, Mamba-Ti-MSQNet successfully identifies the sole instance of *Swimming in circles* (124) in the test set, a task in which MSQNet failed to detect.
- **Transport:** Comparable results were achieved from both the models in the Transport actions (5, 6, 12, 13). The action *Being carried* (4) has no entry for the test set (see Table 3).

4. Discussion

Our study introduces a novel approach utilizing Mamba blocks, a Selective State Spaces architecture, for action recognition in animal behavior analysis, comparing it to the state-of-the-art Transformer model, MSQNet. We developed a family of Mamba-MSQNet architectures with varying sizes based on the VideoMamba video encoder and

Table 3
Area Under the Curve (AUC) results for each action in the Animal Kingdom dataset using MSQNet and our best model (Mamba-Ti-MSQNet).

Action	ID	MSQNet	Ours	Action	ID	MSQNet	Ours
Affection				Movement (continue)			
Holding Hands	63	–	–	Digging	25	0.323	0.550
Hugging	65	0.572	0.725	Diving	28	0.193	0.198
Showing Affection	106	0.598	0.728	Doing a backward tilt	30	0.094	0.336
Aggressive				Doing a neck raise	33	0.681	0.788
Attacking	1	0.394	0.337	Doing a side tilt	34	–	–
Chasing	14	0.346	0.340	Doing somersault	36	–	–
Coiling	17	0.0	0.0	Drifting	37	0.320	0.452
Competing for dominance	18	0.069	0.533	Entering its nest	41	0.048	0.087
Disturbing other animal	27	0.868	0.866	Exiting nest	44	0.013	0.127
Fighting	47	0.548	0.612	Falling	46	0.706	0.544
Hissing	62	0.856	0.601	Flapping	48	0.635	0.567
Pounding	89	–	–	Flapping tail	50	0.0	0.026
Preying	91	0.132	0.304	Flying	52	0.720	0.654
Rattling	94	0.239	0.175	Gliding	57	0.499	0.286
Spitting venom	112	0.889	0.893	Hanging	59	0.700	0.764
Wrapping itself around prey	137	0.064	0.708	Hopping	64	0.617	0.328
Wrapping prey	138	0.176	0.039	Immobilized	66	0.295	0.464
Communication				Jumping	67	0.607	0.579
Barking	3	0.660	0.716	Landing	69	0.512	0.485
Calling	10	0.192	0.303	Lying down	70	–	–
Chirping	15	0.796	0.795	Learning	72	0.025	0.250
Giving off light	56	0.0	1.0	Lying on its side	74	0.506	0.667
Waving	136	0.0	1.0	Manipulating object	76	0.694	0.745
Death				Moving	78	0.574	0.587
Dead	21	0.505	0.851	Pulling	93	0.0	1.0
Dying	39	0.955	0.995	Rolling	98	0.415	0.505
Defensive				Running	100	0.515	0.574
Camouflaging	11	0.883	0.819	Running on water	101	0.433	0.696
Defensive rearing	23	0.068	0.246	Sinking	107	0.006	0.349
Displaying Defensive Pose	26	0.853	0.880	Sitting	108	0.423	0.319
Doing a back kick	29	0.291	0.194	Spreading	113	–	–
Escaping	42	0.014	0.047	Spreading wings	114	0.280	0.285
Fleeing	51	0.300	0.284	Squatting	115	0.002	0.334
Retaliating	96	0.540	0.513	Standing	116	0.122	0.302
Retreating	97	0.775	0.764	Surfacing	121	0.773	0.771
Standing in alert	117	0.219	0.380	Swaying	122	0.023	0.350
Struggling	120	0.442	0.539	Swimming	123	0.775	0.795
Feeding				Swinging	125	0.002	0.001
Biting	8	0.619	0.626	Turning around	128	0.465	0.531
Drinking	38	0.884	0.884	Unmounting	130	0.004	0.0
Eating	40	0.625	0.634	Unrolling	131	0.0	0.0
Pecking	80	0.835	0.851	Walking	133	0.635	0.625
Sharing food	105	0.386	0.455	Walking on water	134	0.0	0.0
General				Prey			
Flapping its ears	49	0.545	0.522	Being eaten	7	0.375	0.494
Gasping for air	53	–	–	Getting bullied	54	0.0	0.167
Keeping still	68	0.519	0.533	Playing dead	88	0.002	0.551
Lying on top	75	0.020	0.190	Trapped	127	0.017	0.335
Panting	79	0.001	0.002	Resting			
Spitting	111	–	–	Resting	95	0.013	0.002
Startled	118	0.685	0.639	Sleeping	109	0.006	0.006
Stinging	119	0.021	0.036	Sleeping in its nest	110	0.633	0.700
Tail swishing	126	–	–	Yawning	139	0.312	0.555

(continued on next page)

Table 3 (continued).

Lily cycle				Sensing			
Exiting cocoon	43	0.004	0.355	Attending	2	0.514	0.500
Giving birth	55	0.126	0.508	Exploring	45	0.367	0.399
Hatching	60	0.0	0.0	Having a Flehmen resp.	61	0.673	0.557
Laying eggs	71	0.006	0.415	Sensing	102	0.468	0.488
Molting	77	–	–	Sexual			
Undergoing chrysalis	129	0.0	1.0	Dancing	19	0.106	0.785
Maintenance				Dancing on water	20	1.0	1.0
Defecating	22	0.729	0.711	Doing push up	35	0.122	1.0
Doing a chin dip	31	1.0	0.633	Perf. copulation mounting	83	0.032	0.752
Doing a face dip	32	0.896	0.897	Perf. sexual display	84	0.867	0.885
Grooming	58	0.319	0.314	Perf. sexual exploration	85	0.002	0.042
Licking	73	0.004	0.501	Perf. sexual pursuit	86	–	–
Perf. allo-grooming	81	0.335	0.526	Puffing its throat	92	0.002	0.130
Perf. allo-preening	82	0.359	0.073	Shelter			
Preening	90	0.780	0.773	Building nest	9	0.223	0.272
Rubbing its head	99	0.811	0.919	Social			
Shaking	103	0.548	0.489	Playing	87	0.670	0.841
Shaking head	104	0.636	0.600	Swimming in circles	124	0.055	1.0
Urinating	132	0.0	0.501	Transport			
Washing	135	0.864	0.788	Being carried	4	–	–
Movement				Being carried in mouth	5	0.686	0.734
Abseiling	0	0.0	0.0	Being dragged	6	0.070	0.001
Climbing	16	0.233	0.275	Carry	12	0.020	0.007
Detaching as a parasite	24	–	–	Carrying in mouth	13	0.541	0.600

the number of Mamba blocks used. Our findings indicate that smaller configurations of VideoMamba performed better, while increasing the number of Mamba blocks consistently led to improved results within similar configurations.

One of the most notable advantages of our models is their computational efficiency compared to the Transformer-based model, demonstrating a substantial reduction in both FLOPs and model parameters. These efficiency gains are crucial for fast inference, making our models applicable to potentially constrained devices in real environments for on-the-fly analysis. However, our model still relies on CLIP for text and image encoding, which could be a limitation. Text encoding is performed only once before training, with Learnable Label Embeddings (resulting in a matrix) being the only part considered during inference and training. The image decoder is employed for every operation within the network. Currently, no Mamba-based models that can effectively perform like CLIP are completely available (Huang et al., 2024).

Furthermore, we tested our best model on a cost-effective GPU, the NVIDIA GTX 1650 (approximately \$200 USD at the time of submission), to assess its inference speed, which is crucial for real-world animal monitoring applications. Our model is able to predict actions in a video in just under 0.8 s on average, whereas MSQNet takes 1.3 s. Given that the average clip length in the Animal Kingdom dataset is about 5.5 s (with 16 frames per clip, meaning a frame every 0.35 s), our model is able to process and prepare for the next sequence after approximately 2.3 meaningful frames. In contrast, MSQNet requires 3.7 frames before it can process again. This difference represents a significant reduction in processing time, particularly for continual monitoring applications. For instance, after an hour of continuous monitoring, our Mamba-MSQNet model would be capable of making 4500 predictions, compared to MSQNet's 2769.

Analysis of the Multilabel AUC results yields interesting conclusions. The Mamba architectures employed in our best-performing model are designed to capture long-range dependencies more effectively than traditional Transformer architectures. This is a key strength, as many of the actions in the Animal Kingdom dataset, such as those in the Movement, Maintenance, and Life Cycle categories, involve complex, sequential patterns that span longer time horizons. For example, actions like *Exiting cocoon* (43), *Giving birth* (55), and *Laying eggs* (71) are

inherently sequential and require the model to understand the temporal evolution of an animal's behavior. The Mamba blocks, with their ability to selectively attend to relevant past states, are better equipped to learn these long-range dependencies compared to the fixed-size attention windows in Transformer-based models like MSQNet.

Similarly, the improved performance on low-sample actions can be attributed to the Mamba architecture's capacity to generalize better from limited data. By effectively modeling the temporal dynamics and contextual cues within the video sequences, Mamba-Ti-MSQNet is able to extract more robust and discriminative features, even for behaviors that are rarely observed in the training set. This advantage is particularly evident in actions like *Competing for dominance* (53), *Licking* (73), and others with sparse representations. The Mamba blocks' selective attention mechanism allows the model to focus on the most relevant information, mitigating the impact of limited training samples and enhancing its ability to recognize these behaviors accurately.

Furthermore, the performance boost in critical actions related to animal well-being, such as *Dead* (21), *Immobilized* (66), and *Urinating* (132), suggests that the Mamba-based architecture is better able to capture the subtle cues and contextual nuances that distinguish these important behaviors. This could be invaluable in real-world applications, where reliably identifying these actions is crucial for monitoring animal health and welfare (Wilkinson, 2011a).

In contrast, the few cases where MSQNet outperformed our model, such as *Hissing* (62) and *Rattling* (94), may indicate that the Transformer-based approach is better suited for capturing certain types of short-term, localized patterns in the data. However, the overall trends demonstrate the Mamba-Ti-MSQNet's superior capabilities in handling the complex, long-range dependencies and sparse representations prevalent in the Animal Kingdom dataset.

4.1. Robustness testing

We further demonstrated the robustness of our approach by evaluating it on another Animal Action Recognition dataset, the BaboonLand dataset (Duporge et al., 2024). This dataset consists of 20 h of video footage captured by Unmanned Aerial Vehicles (UAVs), showcasing ba-

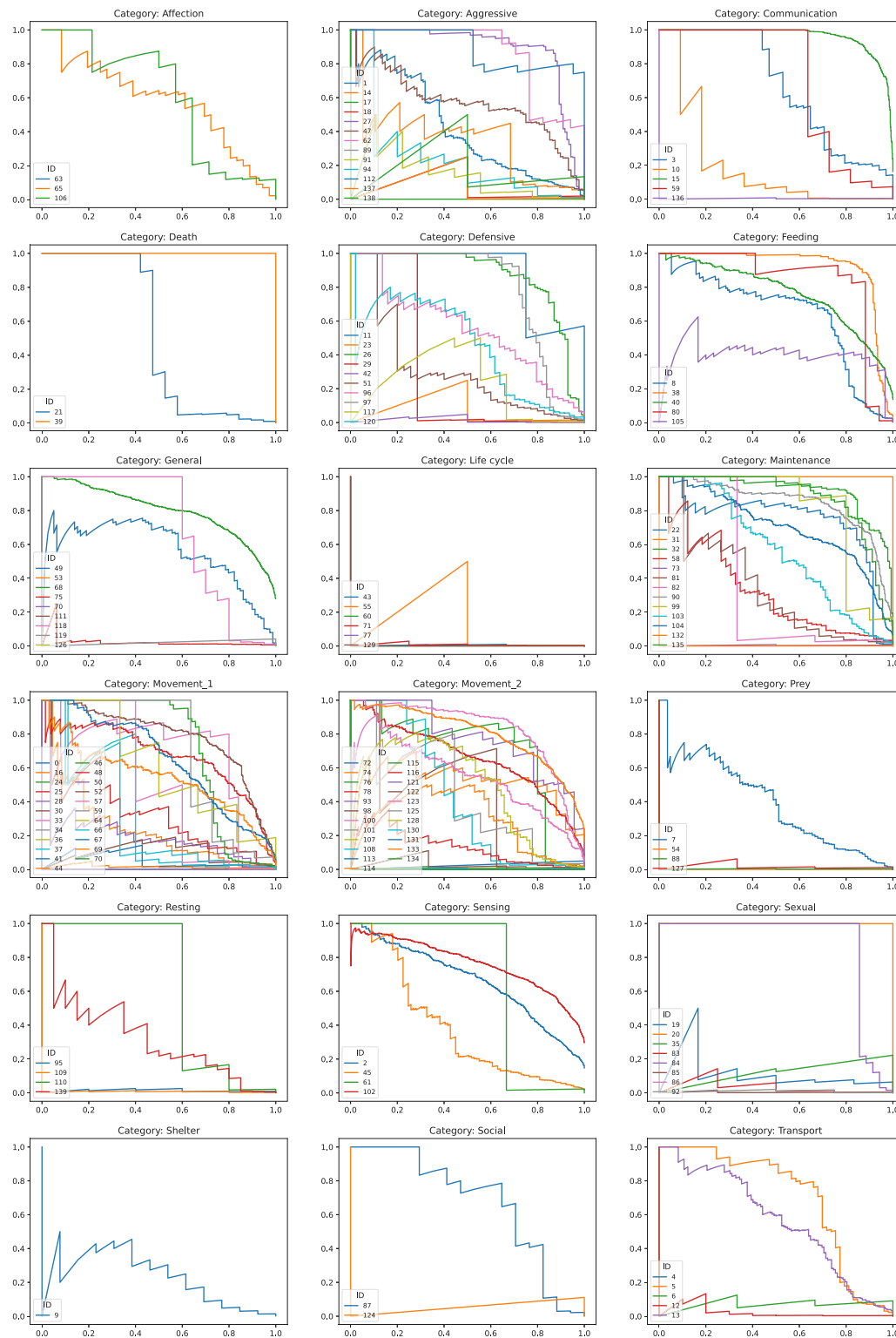


Fig. 2. Precision–Recall curve for MSQNet considering the multilabel nature of our task. The legend displays the action IDs.

boons performing 12 distinct actions (13 when including the ‘occluded’ action). BaboonLand presents a realistic and challenging real-world application scenario for testing our model. Although BaboonLand features fewer action categories than the Animal Kingdom dataset, it introduces unique challenges, such as fully occluded animals and varying camera angles. These characteristics allowed us to assess the model’s capacity to differentiate between frames where no action is visible (occluded) and those where actions are observable.

Table 4 reports the micro mAP results for our Mamba-MSQNet models, the X3D baseline presented in the original BaboonLand paper, and MSQNet, which we trained to ensure a fair comparison. For these experiments, we used VideoMamba backbones pretrained on ImageNet-1K followed by fine-tuning on K400, as this configuration yielded the best results on the Animal Kingdom dataset. The results clearly show that our Mamba-MSQNet outperforms both the baseline and MSQNet, achieving the highest scores across configurations.

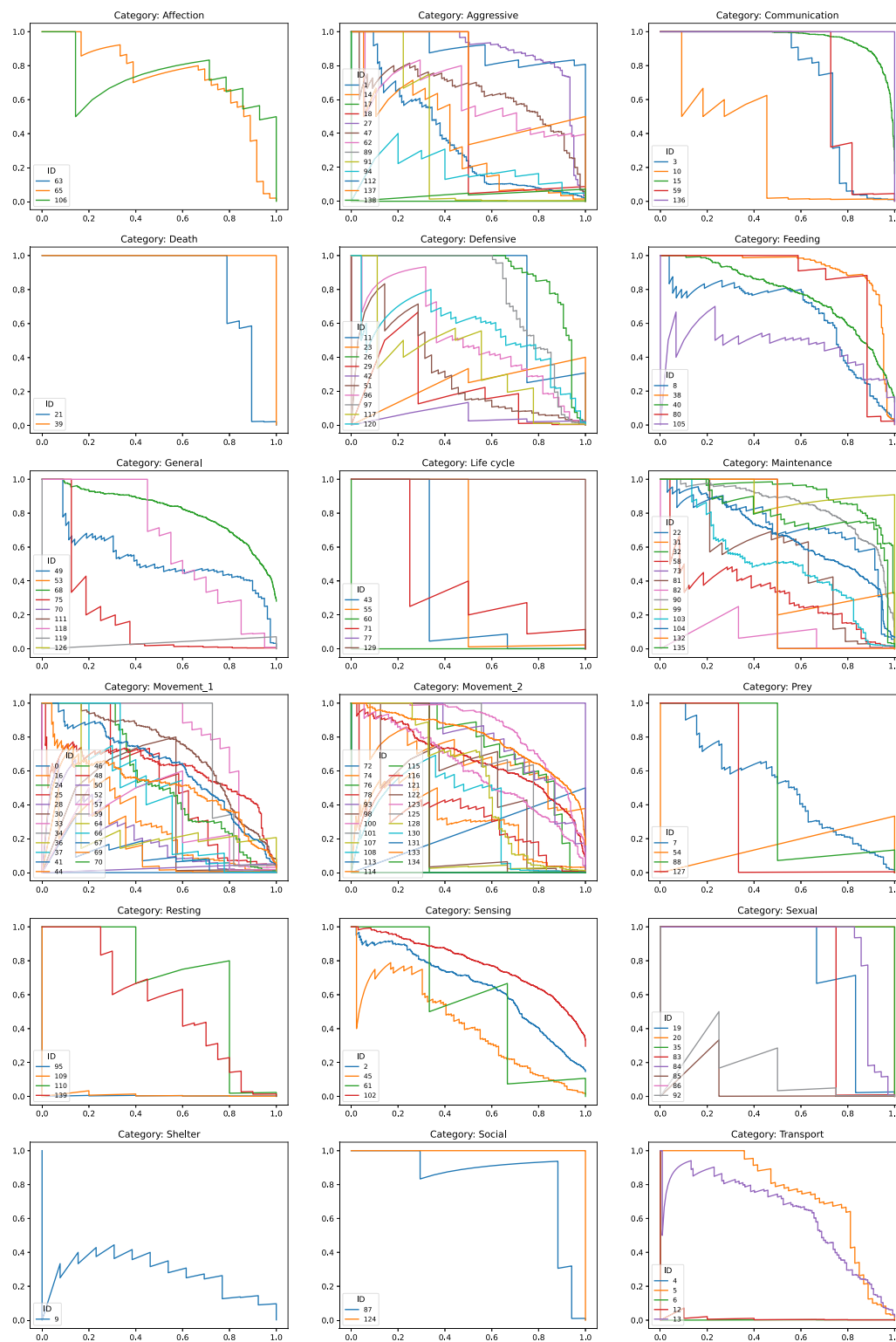


Fig. 3. Precision–Recall curve for our best model (Mamba-Ti-MSQNet using 16 Mamba blocks) considering the multilabel nature of our task. The legend displays the action IDs.

Once again, increasing the depth of the Mamba blocks led to general improvements in mAP performance. Additionally, smaller backbone configurations performed exceptionally well, demonstrating their ability to extract meaningful features for the multimodal model even with their reduced size. This finding underscores the effectiveness of shallow VideoMamba architectures in generating compact and informative representations compared to deeper configurations. However, the highest score was achieved by Mamba-M-MSQNet with 4 Mamba layers,

reaching a mAP of 78.9, followed by our previous best configuration for Animal Kingdom, Mamba-Ti-MSQNet with 16 Mamba layers, that achieved a comparable mAP of 78.3.

4.2. Possible challenges

Although our model outperforms prior efforts on both the Animal Kingdom and BaboonLand datasets, its mean Average Precision (mAP)

Table 4

Results in terms of mean Average Precision (mAP) for the Baboonland dataset. VidEnc stands for Video Encoder. The ‘Pretrained On’ columns refer to the initial pretrained before the actual trained of the network on the ‘Trained On’ task. The best result is the one in bold.

Model	# Mamba layers	VidEnc pretrained on	VidEnc trained on	mAP
X3D (baseline)	–	–	–	63.9
MSQNet	–	–	K400	76.5
Mamba-M-MSQNet	16	ImageNet-1K	K400	78.1
Mamba-M-MSQNet	8	ImageNet-1K	K400	78.2
Mamba-M-MSQNet	4	ImageNet-1K	K400	78.9
Mamba-S-MSQNet	16	ImageNet-1K	K400	78.3
Mamba-S-MSQNet	8	ImageNet-1K	K400	77.7
Mamba-S-MSQNet	4	ImageNet-1K	K400	77.5
Mamba-Ti-MSQNet	16	ImageNet-1K	K400	78.3
Mamba-Ti-MSQNet	8	ImageNet-1K	K400	76.7
Mamba-Ti-MSQNet	4	ImageNet-1K	K400	76.7

scores indicate that these tasks are far from being completely solved. Based on our analysis of the results, we have identified several challenges that complicate the recognition of animal behaviors. Specifically, we highlight four key challenges:

- **Occlusions or reduced visibility.** In some clips, animals are either partially visible or entirely occluded, as some in BaboonLand. Partial visibility can arise from various factors. For example, in the Animal Kingdom dataset, objects such as grass, trees, or walls often obscure parts of the animal. Additionally, environmental conditions like fog, underwater settings, or darkness further limit visibility and hinder the model’s ability to accurately discern animal actions.
- **Distance from the animal.** Besides occlusions, the distance between the animal and the camera significantly impacts action recognition. BaboonLand videos, captured using AUCs, consistently maintain a distance that allows animals to appear in their entirety. In contrast, the Animal Kingdom dataset varies widely in distance, presenting challenges for the model. For instance, close-up shots of animals often make it difficult to differentiate actions such as attending and staying still. While physiological features in an animal’s face could theoretically help deduce its actions, these nuances are challenging for the model to interpret.
- **Complexity of actions.** The complexity of actions also poses challenges. Some actions are intricate, involving multiple movements to complete. However, even simpler actions requiring minimal movement can be difficult to classify due to the need for contextual or physiological understanding that video alone cannot provide. For example, actions like resting, sleeping, or lying down could appear visually similar, but distinguishing between them often requires knowledge of the animal’s internal state or behavior patterns. This limitation likely explains the poor AUC scores for actions such as resting and sleeping.
- **Actions involving multiple animals.** Some actions, such as those in the Aggressive category, involve interactions between multiple animals, which complicates interpretation. For example, a chasing action could be mistaken for running if the other animal involved is not visible in the clip. This challenge highlights the importance of capturing the full context of an interaction. Analyzing a single animal in isolation may not provide sufficient information to accurately classify such actions.

5. Conclusions

Our work demonstrates the potential of Selective Space Models and Mamba blocks in revolutionizing animal action recognition, significantly improving inference speed and model suitability for constrained

devices. We show that it is possible to achieve performance comparable to large Transformer-based models like MSQNet with fewer parameters and FLOPs. The introduction of our Mamba-MSQNet family of architectures led to model size reductions of up to 78% (excluding the CLIP image encoder) and FLOPs reductions of up to 90% compared to MSQNet. Even with these reductions, our models achieved similar or slightly better results, with Mamba-Ti-MSQNet (featuring 16 Mamba blocks and a VideoMamba backbone pre-trained on ImageNet-1K and K400 datasets) reaching a mean Average Precision (mAP) of 74.6, compared to MSQNet’s 73.1.

To further validate the robustness and performance of our architectures, we tested them on the BaboonLand dataset. Mamba-Ti-MSQNet proved to be highly effective in recognizing baboon actions, achieving an mAP of 78.9, marking it as the top-performing model for that dataset.

While our Mamba-MSQNet family of architectures achieved impressive results, several limitations has been seen and documented as possible challenges, whose require future investigation. We addressed four of them, which we categorized as occlusions or reduced visibility, camera distance from the animals, complexity of actions (both in terms number of movements for performing the whole action and in physiological interpretations), actions requiring the presence of another animals, where focus on one animal could limit interpretability.

Our advancements in accelerating and enhancing animal action recognition represent a crucial step towards real-time animal monitoring, which is vital for timely interventions in diverse settings. This research has significant implications for applications in wildlife conservation, veterinary care, and automated monitoring in farming or ecological studies. By providing fast and accurate insights into animal behavior, our model can aid in the study of animal welfare, health monitoring, and conservation efforts, potentially transforming how we monitor animal populations and welfare on a large scale.

Future work will focus on the implementation of this system in real-world scenarios, further improving model performance through adaptation to specific animal species. Additionally, we will explore its application in longitudinal behavioral studies, automated welfare monitoring systems, and scaling the technology for use in diverse environments, such as remote wildlife monitoring or large-scale farm management.

CRedit authorship contribution statement

Edoardo Fazzari: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Donato Romano:** Writing – review & editing, Supervision, Funding acquisition. **Fabrizio Falchi:** Writing – review & editing, Supervision. **Cesare Stefanini:** Writing – review & editing, Supervision, Funding acquisition.

Code availability

The code used was uploaded on GitHub and it is available at <https://github.com/edofazza/mamba-msqnet>.

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.

Acknowledgment

This work was partially carried out in the framework of the H2020 FETOPEN Project “Robocoenosis-ROBOts in cooperation with a bio-COENOSIS” [899520]. The founder had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Data availability

Code and data shared on GitHub as indicated in the article.

References

- Alameer, A., Buijs, S., O'Connell, N., Dalton, L., Larsen, M., Pedersen, L., Kyriazakis, I., 2022. Automated detection and quantification of contact behaviour in pigs using deep learning. *Biosyst. Eng.* 224, 118–130.
- Arablouei, R., Wang, L., Currie, L., Yates, J., Alvarenga, F.A., Bishop-Hurley, G.J., 2023. Animal behavior classification via deep learning on embedded systems. *Comput. Electr. Agric.* 207, 107707.
- Bernardes, R.C., Lima, M.A.P., Guedes, R.N.C., da Silva, C.B., Martins, G.F., 2021. Ethoflow: computer vision and artificial intelligence-based software for automatic behavior analysis. *Sensors* 21 (3237).
- Bertasius, G., Wang, H., Torresani, L., 2021b. Is space-time attention all you need for video understanding? *ICML* 2 (4).
- Dang, T.H., Dang, N.H., Tran, V.T., Chung, W.Y., 2022. A lorawan-based smart sensor tag for cow behavior monitoring. In: 2022 IEEE Sensors. IEEE, pp. 1–4.
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
- Duporge, I., Kholiavchenko, M., Harel, R., Wolf, S., Rubenstein, D., Crofoot, M., Berger-Wolf, T., Lee, S., Barreau, J., Kline, J., 2024. BaboonLand dataset: Tracking primates in the wild and automating behaviour recognition from drone videos. arXiv preprint arXiv:2010.11929.
- Fazzari, E., Romano, D., Falchi, F., Stefanini, C., 2024. Animal behavior analysis methods using deep learning: A survey. arXiv preprint arXiv:2010.11929.
- Feng, L., Zhao, Y., Sun, Y., Zhao, W., Tang, J., 2021. Action recognition using a spatial-temporal network for wild felines. *Animals* 11 (485).
- Fu, D.Y., Dao, T., Saab, K.K., Thomas, A.W., Rudra, A., Ré, C., 2022. Hungry hungry hippos: Towards language modeling with state space models. arXiv preprint arXiv:2010.11929.
- Ghosh, S., Dasgupta, R., 2022a. Study of Animal Behavior and Machine Learning. Springer Nature Singapore, Singapore, pp. 231–237. http://dx.doi.org/10.1007/978-981-16-8881-2_27.
- Goyal, R., Kahou, S.Ebrahimi, Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al., 2017. The something something video database for learning and evaluating visual common sense. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5842–5850.
- Gu, A., Dao, T., 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2010.11929.
- Gu, A., Goel, K., Gupta, A., Ré, C., 2022. On the parameterization and initialization of diagonal state space models. *Adv. Neural Inf. Process. Syst.* 35, 35971–35983.
- Gu, A., Goel, K., Ré, C., 2021a. Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2010.11929.
- Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., Ré, C., 2021b. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Adv. Neural Inf. Process. Syst.* 34, 572–585.
- Gupta, A., Gu, A., Berant, J., 2022. Diagonal state spaces are as effective as structured state spaces. *Adv. Neural Inf. Process. Syst.* 35, 22982–22994.
- Huang, W., Shen, Y., Yang, Y., 2024. Clip-mamba: Clip pretrained mamba models with ood and hessian evaluation. arXiv preprint arXiv:2010.11929.
- Kavlak, A.T., Pastell, M., Uimari, P., 2023. Disease detection in pigs based on feeding behaviour traits using machine learning. *Biosyst. Eng.* 226, 132–143.
- Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al., 2017. The kinetics human action video dataset. arXiv preprint arXiv:2010.11929.
- Kleanthous, N., Hussain, A.J., Khan, W., Sneddon, J., Al-Shamma'a, A., Liatsis, P., 2022. A survey of machine learning approaches in animal behaviour. *Neurocomputing* 491, 442–463.
- Kumar, A., Raghunathan, A., Jones, R., Ma, T., Liang, P., 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. arXiv preprint arXiv:2010.11929.
- Li, H., Gu, Z., He, D., Wang, X., Huang, J., Mo, Y., Li, P., Huang, Z., Wu, F., 2024a. A lightweight improved YOLOv5s model and its deployment for detecting pitaya fruits in daytime and nighttime light-supplement environments. *Comput. Electron. Agric.* 220, 108914.
- Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., Qiao, Y., 2024b. Videomamba: State space model for efficient video understanding. arXiv preprint arXiv:2010.11929.
- Liu, D., Oczak, M., Maschat, K., Baumgartner, J., Pletzer, B., He, D., Norton, T., 2020. A computer vision-based method for spatial-temporal action recognition of tail-biting behaviour in group-housed pigs. *Biosyst. Eng.* 195, 27–41.
- Loshchilov, I., Hutter, F., 2016. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:2010.11929.
- Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:2010.11929.
- Lu, Z., Wang, F., Xu, Z., Yang, F., Li, T., 2024. On the performance and memory footprint of distributed training: An empirical study on transformers. arXiv preprint arXiv:2010.11929.
- Luo, L., Liu, W., Lu, Q., Wang, J., Wen, W., Yan, D., Tang, Y., 2021. Grape berry detection and size measurement based on edge image processing and geometric morphology. *Machines* 9 (233).
- Luxem, K., Mocellin, P., Fuhrmann, F., Kürsch, J., Miller, S.R., Palop, J.J., Remy, S., Bauer, P., 2022. Identifying behavioral structure from deep variational embeddings of animal motion. *Commun. Biol.* 5 (1267).
- Manduca, G., Santaera, G., Miraglia, M., Vuuren, G.Jansen.Van., Dario, P., Stefanini, C., Romano, D., 2024. A bioinspired control strategy ensures maneuverability and adaptability for dynamic environments in an underactuated robotic fish. *J. Intell. Robot. Syst.* 110 (69).
- Manoharan, S., 2020. Embedded imaging system based behavior analysis of dairy cow. *J. Electron.* 2, 148–154.
- Mathis, A., Mamidanna, P., Cury, K.M., Abe, T., Murthy, V.N., Mathis, M.W., Bethge, M., 2018. DeepLabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neurosci.* 21, 1281–1289.
- Mei, W., Yang, X., Zhao, Y., Wang, X., Dai, X., Wang, K., 2023. Identification of aflatoxin-poisoned broilers based on accelerometer and machine learning. *Biosyst. Eng.* 227, 107–116.
- Mondal, A., Nag, S., Prada, J.M., Zhu, X., Dutta, A., 2023. Actor-agnostic multi-label action recognition with multi-modal query. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops. pp. 784–794.
- Ng, X.L., Ong, K.E., Zheng, Q., Ni, Y., Yeo, S.Y., Liu, J., 2022. Animal kingdom: A large and diverse dataset for animal behavior understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19023–19034.
- Odo, A., Muns, R., Boyle, L., Kyriazakis, I., 2023. Video analysis using deep learning for automated quantification of ear biting in pigs. *Ieee Access* 11, 59744–59757.
- Ollagnier, C., Kasper, C., Wallenbeck, A., Keeling, L., Bee, G., Bigdeli, S.A., 2023. Machine learning algorithms can predict tail biting outbreaks in pigs using feeding behaviour records. *PLoS One* 18, e0252002.
- Pan, Z., Chen, H., Zhong, W., Wang, A., Zheng, C., 2023. A cnn-based animal behavior recognition algorithm for wearable devices. *IEEE Sens. J.* 23, 5156–5164.
- Peng, X.B., Coumans, E., Zhang, T., Lee, T.W., Tan, J., Levine, S., 2020. Learning agile robotic locomotion skills by imitating animals. arXiv preprint arXiv:2010.11929.
- Pereira, T.D., Tabris, N., Matsliah, A., Turner, D.M., Li, J., Ravindranath, S., Papadopyannis, E.S., Normand, E., Deutsch, D.S., Wang, Z.Y., et al., 2022. Sleep: A deep learning system for multi-animal pose tracking. *Nature Methods* 19, 486–495.
- Rahman, S.A., Song, I., Leung, M.K., Lee, I., Lee, K., 2014. Fast action recognition using negative space features. *Expert Syst. Appl.* 41, 574–587.
- Riekert, M., Klein, A., Adrion, F., Hoffmann, C., Gallmann, E., 2020. Automatically detecting pig position and posture by 2d camera imaging and deep learning. *Comput. Electron. Agric.* 174, 105391.
- Segalin, C., Williams, J., Karigo, T., Hui, M., Zelikowsky, M., Sun, J.J., Perona, P., Anderson, D.J., Kennedy, A., 2021. The mouse action recognition system (mars) software pipeline for automated analysis of social behaviors in mice. *Elife* 10, e63720.
- Sutoko, S., Masuda, A., Kandori, A., Sasaguri, H., Saito, T., Saido, T.C., Funane, T., 2021. Early identification of alzheimer's disease in mouse models: Application of deep neural network algorithm to cognitive behavioral parameters. *IScience* 24.
- Tjandrasuwita, M., Sun, J.J., Kennedy, A., Chaudhuri, S., Yue, Y., 2021. Interpreting expert annotation differences in animal behavior. arXiv preprint arXiv:2010.11929.
- Vaswani, A., 2017. Attention is all you need. *Adv. Neural Inf. Process. Syst.*
- Wilkinson, K.G., 2011a. On-farm composting of dead stock. In: Kumar, S. (Ed.), *Integrated Waste Management*. vol. 14, IntechOpen, Rijeka, <http://dx.doi.org/10.5772/20231>.
- Xiao, S., Wang, Y., Perkes, A., Pfrommer, B., Schmidt, M., Daniilidis, K., Badger, M., 2023. Multi-view tracking, re-id, and social network analysis of a flock of visually similar birds in an outdoor aviary. *Int. J. Comput. Vis.* 131, 1532–1549.
- Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X., 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint arXiv:2010.11929.