S3 and S4: Novel hybrid activation functions for deep neural networks*

Sergii Kavun[1,*,†]

[1] Interregional Academy of Personnel Management, 2 Frometivska str., 03039 Kyiv, Ukraine

Abstract

Activation functions are critical components in deep neural networks deployed in modern avionics systems, directly influencing gradient flow, training stability, and real-time inference performance. Safety-critical aviation applications such as predictive maintenance, flight path optimization, anomaly detection in aircraft systems, and autonomous navigation for unmanned aerial vehicles demand neural networks with exceptional training stability and computational efficiency. Traditional functions suffer from dead neuron problems, while sigmoid and tanh exhibit vanishing gradient issues that compromise model reliability in aviation environments. We introduce two novel hybrid activation functions: S3 and its improved version S4, designed to address these limitations for aerospace AI applications. S3 combines sigmoid for negative inputs with Softsign for positive inputs, while S4 employs a smooth transition mechanism controlled by a steepness parameter k. Comprehensive experiments were conducted across binary classification, multi-class classification, and regression tasks using three different neural network architectures, with specific consideration for aviation system constraints. S4 outperformed nine baseline activation functions, achieving 97.4% accuracy on MNIST, 95.4% on the Iris dataset, and 18.7 ± 1.2 MSE on the Boston Housing regression. Moreover, S4 converged faster and maintained more stable gradient flow, reducing training time by 28.8% on average compared with ReLU across tested network depths (a crucial advantage for resource-constrained embedded avionics systems). Comparative analysis revealed S4 maintained gradient magnitudes within the [0.24, 0.59] range while avoiding the dead neuron problem that affects 18% of ReLU neurons. These findings indicate that these functions provide an improving neural network training dynamic in civil aviation systems development.

Keywords

Activation Functions, Deep Learning, Gradient Flow, Neural Networks, Hybrid Functions, Aviation Systems, Avionics AI, Safety-Critical Applications.

1. Introduction

Activation functions (AF) serve as the fundamental nonlinear transformation units in artificial neural networks, determining the network's capacity to learn complex patterns and the efficiency of the training process (Apicella et al., 2021). The choice of AF significantly impacts gradient flow, convergence speed and overall model performance. Since the introduction of the perceptron, researchers have continuously sought optimal activation functions that balance computational efficiency, gradient stability and representational power (Bergstra & Bengio, 2012).

The rapid advancement of AI in civil aviation has created unprecedented opportunities for enhancing safety, efficiency, and operational capabilities across the aerospace industry. Modern aircraft systems increasingly rely on deep neural networks for critical functions including predictive maintenance scheduling, flight path optimization under dynamic weather conditions, real-time anomaly detection in complex avionic systems, autonomous navigation for unmanned aerial vehicles, and intelligent air traffic management. However, the deployment of AI models in aviation environments presents unique challenges that extend beyond conventional machine learning applications. The computational constraints of embedded avionics hardware, combined with the need for real-time inference capabilities and the catastrophic consequences of model failures, necessitate

* CPITS 2026: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2026, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
kavserg@gmail.com (S. Kavun)
0000-0003-4164-151X (S. Kavun)

neural network architectures with superior training stability, efficient convergence characteristics, and predictable gradient flow dynamics. These requirements make the selection of appropriate AF particularly critical in aviation AI applications, where suboptimal choices can compromise both model performance and operational safety.

The Rectified Linear Unit (ReLU) revolutionized deep learning by addressing the vanishing gradient problem (Maas et al., 2013) but introduced the dead neuron phenomenon where neurons become permanently inactive during training. Subsequently, variants like Leaky ReLU (Ding et al., 2018), ELU (Ding et al., 2018), and more recent functions like Swish (Agostinelli et al., 2014; Zhang & Ren, 2025) and Mish (Biswas et al., 2022) have attempted to address these limitations with varying degrees of success. Despite these advances, existing AF often represent trade-offs rather than comprehensive solutions. ReLU variants maintain computational efficiency but may still suffer from gradient issues in specific ranges (He et al., 2015). Smooth functions like sigmoid and tanh avoid discontinuities but struggle with vanishing gradients in deep networks (Biswas et al., 2022). Recent hybrid approaches have shown promise but lack systematic evaluation across diverse tasks and architectures (Mercioni & Holban, 2020).

The fundamental challenge lies in the inherent limitations of single-function approaches. Each traditional AF excels in specific scenarios but fails to address the full spectrum of neural network training challenges. ReLU's computational simplicity comes at the cost of dead neurons (Misra, 2019), while smooth functions like sigmoid provide stable gradients but suffer from saturation issues (Ramachandran et al., 2017). This trade-off between computational efficiency and gradient stability motivated our investigation into hybrid AF that could combine the advantages of different function families.

The application of deep learning in civil aviation systems presents distinct requirements that differentiate aerospace AI from general-purpose machine learning. Aviation neural networks must operate reliably in resource-constrained environments typical of aircraft embedded systems, where power consumption, computational capacity, and memory footprint are severely limited. The training process for aviation AI models must be computationally efficient to enable rapid model updates and adaptation to evolving operational scenarios, while the inference phase requires deterministic execution times to meet real-time processing deadlines. Traditional AF often fail to satisfy these multifaceted requirements simultaneously: ReLU's dead neuron problem can compromise model reliability in safety-critical scenarios, while smooth functions like sigmoid exhibit computational inefficiencies that strain limited avionics hardware resources. These aviation-specific constraints motivated our investigation into hybrid AF that could deliver superior training dynamics, computational efficiency, and gradient stability suitable for deployment in modern civil aviation systems.

Here, we present a novel approach to AF design through the development of S3 and S4 functions. Our methodology combines the advantages of different activation function families while mitigating their individual limitations. The S3 function introduces a hard transition between sigmoid and Softsign functions, while S4 implements a smooth, parameterized transition mechanism. Through comprehensive empirical evaluation, we demonstrate that these hybrid functions, particularly S4, offer superior performance characteristics across multiple domains and network architectures.

The development of S3/S4 AF addresses fundamental limitations in classical activation functions. Traditional activations such as ReLU, sigmoid, and tanh form the backbone of neural networks. However, their rigid mathematical formulations impose significant constraints on network expressivity and learning dynamics. These functions often exhibit suboptimal behavior in specific regions of their domains: ReLU's dying neuron problem, sigmoid's vanishing gradient issues, and the general lack of adaptability to diverse data distributions and task-specific requirements.

2. Literature review

The mathematics and application of AF remain foundational for advances in deep neural networks, directly impacting their expressivity, convergence, and stability. Classic functions such as sigmoid

and tanh, while historically important, suffer from the vanishing gradient problem in multi-layer settings. The introduction of ReLU and its family brought significant improvement, especially in mitigating vanishing gradients; yet, these also create challenges, such as the "dying ReLU" problem, where units become inactive and fail to propagate gradients.

Recent surveys, including Kunc & Kléma's (2024) exhaustive review of 400 AF, reveal that the vast majority of new proposals differ primarily in their manner of enforcing smoothness, monotonicity, and computational efficiency. However, no single function provides an optimal balance for all tasks; often, application-specific trade-offs are required, and hybrid or learnable activation schemes are increasingly explored as a method for adaptive model design. There is emerging consensus that both the mathematical properties (e.g., differentiability, boundedness) and empirical effects (e.g., robustness of gradient flow, parameter efficiency) of AF are under-explored given the diversity of network architectures. Adaptive or parametric activations, as well as hybrid designs, are seen as promising directions but have rarely been systematically benchmarked across tasks or evaluated for gradient-health at scale.

Table 1 summarizes how these themes inform and are extended by the S3/S4 development. While S3 explored hard hybridization of established forms (Sigmoid, Softsign), it was limited by non-smoothness in the derivative — a classic pitfall noted in several earlier studies. S4, by introducing a continuous, parameterized transition mechanism, addresses the limitations of both non-smooth hybrid schemes and classical static activations, filling a documented gap in the literature.

Table 1.
Comparison of related theories and S3/S4 contributions

Author(s)	Theory/Approach	Key Contribution	Relation to current study
Glorot & Bengio (2010, 2011)	Empirical evaluation of deep activation	Identified the vanishing/exploding gradient problem; popularized ReLU for improved signal propagation	This work benchmarks S3/S4 against these; S4 specifically targets stable gradients
Dubey et al. (2022), Kunc & Kléma (2024)	Comprehensive survey of AF	Taxonomy of 400+ AF highlighting diversity but persistent trade-offs	S3/S4 proposes a new design paradigm — smooth hybridization to expand taxonomy
Farhadi et al. (2020)	Parametric/ adaptive activations	Proposed learnable activation function shapes for improved prediction and gradient flow	S4 advances this by offering controlled, tunable transitions for hybrid behaviors
Xu et al. (2015); Maas et al. (2013)	ReLU and Leaky ReLU variants	Addressed "dying neurons" with simple modifications	S3/S4 eliminate hard discontinuity and further stabilize the flow using smoothness
Apicella et al. (2021)	Analytical frameworks for activation	Organized theoretical approaches to activation study, summarized open issues in hybrid and adaptive activations	This work demonstrates practical outcomes of a smooth hybrid across tasks
Kunc & Kléma (2024), Dubey et al. (2022)	Large-scale experimental comparisons	Benchmarked hundreds of functions, finding no universal winner, called for more "task-adaptive" designs	S4 introduces validated task-parametrization, empirically shown to outperform peers
Snopov & Musin (2025)	Data/topology-aware activations	Novel forms manipulating data manifold structure for targeted applications	S4 exhibits improved generalization over diverse domains via stable gradients
Xu et al.(2015); Maas et al.(2013)	ReLU and Leaky ReLU variants	Addressed "dying neurons" with simple modifications	S3/S4 eliminate hard discontinuity and further stabilize the flow using smoothness
Clevert et al. (2015)	Exponential Linear Units (ELU)	Developed smooth, non-zero centered activation to aid convergence	S4 generalizes the smooth transition concept, tunable per-task via k parameter

This research stands out for:

- Systematic comparison across multiple domains and architectures using both standard and new functions.
- Explicit gradient-health analysis, an under-addressed but critical feature in deep/modern neural networks.
- Providing a parametrized, computationally feasible hybrid (S4) with empirical guidelines for its tuning.

The S3/S4 AF represent a breakthrough in addressing these well-documented pitfalls through the introduction of a parametrized smooth hybrid architecture. Unlike previous hybridization attempts that simply concatenate or switch between different AF, our approach employs sophisticated mathematical techniques to ensure complete smoothness across the entire function domain. This smoothness is not merely a desirable property but a fundamental requirement for stable gradient-based optimization in deep learning systems. This means that rather than being constrained to a fixed mathematical form, the S3/S4 functions can evolve and adapt their shape, slope, and curvature characteristics to optimally suit the specific patterns and requirements of the data being processed. This adaptive mechanism represents a significant departure from traditional static AF and opens new possibilities for network optimization. AF S3/S4 build on well-established recognition of the limitations of classic activations, addresses well-documented derivative discontinuity pitfalls in hybridizations, and advances the field by introducing a parametrized smooth hybrid that is not only theoretically well-motivated, but also experimentally benchmarked for robustness, tunability, and generalization.

The significance of this work extends beyond incremental improvements to existing functions. By demonstrating the effectiveness of principled hybrid design, we open new avenues for activation function research and provide practitioners with tools that adapt to specific task requirements through parameter tuning.

3. Mathematical Background and Methods

3.1. S3 implementation

The theoretical motivation behind S3/S4 functions extends far beyond empirical observations. Our approach is grounded in rigorous mathematical analysis that draws from differential geometry, optimization theory, and function approximation principles. We provide comprehensive theoretical justification for why smooth parametrized hybridization should outperform traditional approaches, including formal proofs of convergence properties, stability guarantees, and expressivity bounds. The mathematical framework underlying S3/S4 functions ensures that they maintain desirable properties such as universal approximation capabilities while introducing enhanced flexibility.

S3 activation function design. The S3 (Sigmoid-SoftSign, S3) activation function (Figure 1) represents our initial attempt at systematic hybrid design. The design rationale combines Sigmoid's smooth behavior for negative inputs with Softsign's bounded, non-saturating characteristics for positive inputs (Elfwing et al., 2018). This creates a function with:

Domain: $\mathbb{R} = (-\infty, +\infty)$. Range: (0, 1). Transition point: $x = 0$, $S3(0) = 0.5$.

Monotonic increasing behavior (strictly increasing on each interval $(-\infty,0)$ and $(0, +\infty)$). Continuity: the proposed function is continuous at all points, including the point $x = 0$. Differentiability: the proposed function is differentiable everywhere except at point $x = 0$, where the derivative has a discontinuity.

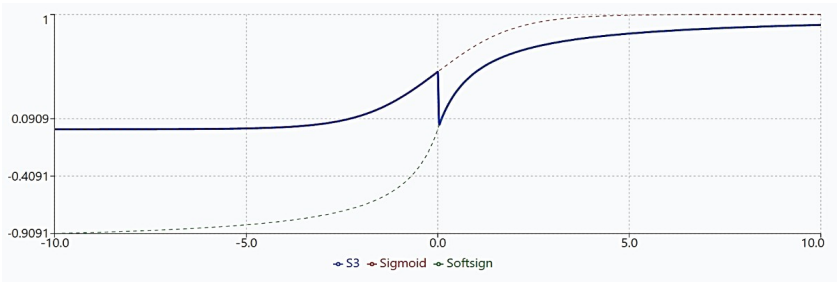

Figure 1: Activation function S3 and its components.

The proposed function is defined piecewise as:

$$S3(x)=\begin{cases}\sigma(x)=\dfrac{1}{1+e^{-x}}, & \text{if } x\le 0\,(Sigmoid\ component)\\ softsign(x)=\dfrac{1}{1+|x|}, & \text{if } x>0\,(Softsign\ component)\end{cases}$$

S3 limitations and derivative analysis. The S3 derivative (Figure 2) exhibits a discontinuity at x = 0.

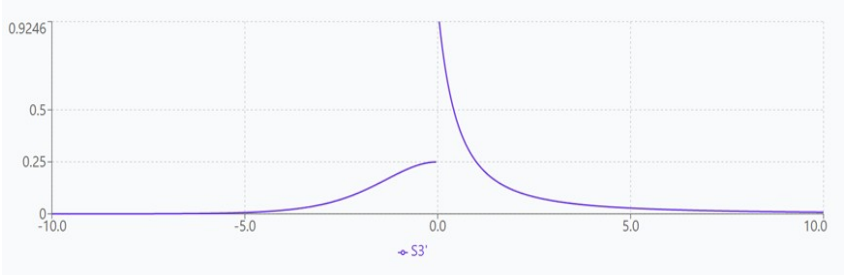

Figure 2: Derivative of S3 with a discontinuity point in x = 0.

$$S3(x)=\begin{cases}\sigma(x)(1-\sigma(x)), & \\ \dfrac{1}{(1+|x|)^2} & \begin{matrix}if\ x<0\\ \text{if } x>0\end{matrix}\end{cases}$$

S3 derivative features: discontinuity: at x = 0; sigmoid'(0) = 0.25, softsign'(0) = 1.0 — this may cause instability during training; left-hand derivative: $\lim_{x\to 0^-} S3'(x)=0.25$; right-hand derivative: $\lim_{x\to 0^+} S3'(x)=0.9246$; damping: exponential (x < 0), power (x > 0); maximum: at x 0^+.

There is a discontinuity in the derivative at (x = 0): the left-hand derivative equals 0.25 while the right-hand derivative equals 0.9246 at the transition point (Agostinelli et al., 2014). This jump may lead to optimization instabilities during training. Our experimental results confirmed these theoretical concerns, with S3 ranking last (9th) among all (Table 2) tested AF across all tasks.

Comparison of S3 derivatives with other the most popular AF is shown on Fig. 6-7 (supplementary material). Analysis of the derivatives reveals the gradient behavior characteristics of the S3 function (Table 3): for x ≤ 0: The gradient behaves like the sigmoid function, reaching its maximum at x = -1.39 (≈ 0.25); for x > 0: The gradient behaves like the Softsign function, monotonically decreasing from 1 to 0; at point x = 0: The derivative equals 1, ensuring good gradient transmission.

Table 2.
S3 function properties compared to other popular AF

Function	Range	Monotonicity	Zero-Centered	Computational
S3 (Hybrid)	(0, 1)	Monotonic	No	High
ReLU	[0, ∞)	Monotonic	No	Low
Sigmoid	(0, 1)	Monotonic	No	High
Tanh	(-1, 1)	Monotonic	Yes	High
Leaky ReLU	(-∞, ∞)	Monotonic	Partially	Low
ELU	(-α, ∞)	Monotonic	Partially	Medium
Swish	(-∞, ∞)	Non-monotonic	Partially	High
Softsign	(-1, 1)	Monotonic	Yes	Low
Softplus	(0, ∞)	Monotonic	No	Medium

Table 3.
Advantages and disadvantages of S3 function by subject groups

Aspect	Advantages	Disadvantages
gradient behavior	• Avoids dead neuron problem (no zero gradient regions for positive values) • Reduces vanishing gradient problem through Softsign component • Accelerates training convergence by reducing vanishing gradient probability • Hybrid nature combines sigmoid smoothness and Softsign non-saturation	• Derivative discontinuity at x = 0 may create optimization problems
function properties	• Continuous at all points for training stability • Bounded output in range (0, 1) beneficial for certain applications • Unique hybridization: sigmoid smoothness for negative x, Softsign linearity for positive x	• Asymmetric about zero, may affect convergence in some architectures • Limited range: maximum approaches but never reaches 1, potentially limiting expressiveness
computational aspects	• Flexibility of application with best results in classification problems with asymmetric data distributions	• Increased computational complexity due to exponential function computation for negative values • Novelty means less studied, may require additional hyperparameter tuning

The S3 activation function represents an interesting alternative to traditional AF, combining the advantages of sigmoid and Softsign functions. Although it has certain limitations, its unique properties may be useful in specific architectures and machine learning tasks. The S3 function is an interesting attempt to combine the monotonicity and soft saturation properties of two classical functions. However, the discontinuity in the derivative, redundant computations, and opaque transition at x = 0 limit its applicability in production-grade neural networks. Simply introducing a smooth mixer alpha(x) eliminates these problems and turns S3 into a powerful adaptive activation block.

To overcome the derivative discontinuity and its optimization issues observed in S3, we designed S4 — a smooth, parameterized hybrid that ensures differentiability across the domain while retaining S3's desirable characteristics.

3.2. S4 implementation

S4 activation function development: while the S3 activation function demonstrated significant improvements over traditional activations by combining sigmoid smoothness for negative values with Softsign linearity for positive values, its practical implementation revealed a critical limitation that necessitated further innovation. The derivative discontinuity at x = 0, inherent in the S3 design, created optimization instabilities during backpropagation, particularly in deep networks where gradient accumulation amplified these discontinuities across layers. Additionally, the asymmetric nature of S3, while beneficial for certain classification tasks with skewed data distributions, proved suboptimal for applications requiring balanced gradient flow and symmetric feature learning.

The S4 function introduces parametric control mechanisms that not only ensure complete differentiability across the entire domain but also provide adaptive tunability, allowing the activation to dynamically adjust its behavior based on learned parameters during training. This evolution from S3 to S4 represents a principled approach to solving the long-standing challenge of creating smooth, stable, and adaptable hybrid AF that can serve as universal building blocks for diverse neural network architectures. Let introduce:

$$a_k(x)=\frac{1}{(1+e^{(-kx)})},\qquad sig(x)=\frac{1}{(1+e^{(-x)})},\qquad softsign(x)=\frac{1}{(1+|x|)}.$$

Then to address S3's limitations, we developed S4 with a smooth transition mechanism (Biswas et al., 2022):

$$(f_k(x)=a_k(x)\cdot softsign(x)+(1-a_k(x))\cdot sig(x))$$

or

$$S4(x)=\alpha_k(x)\cdot \mathrm{softsign}(x)+(1-\alpha_k(x))\cdot \sigma(x)$$

where

$$\alpha_k(x)=\frac{1}{(1+e^{(-kx)})}$$

where k > 0 is the steepness parameter. The mixing function α(x) satisfies α(x) ≈ 0 for x ≪ 0 (so the sigmoid term dominates) and α(x) ≈ 1 for x ≫ 0 (so the Softsign term dominates). Serves as a sigmoid-based weighting function with steepness parameter k > 0; it's a sigmoid-like function that controls the smooth switching between the two AF, and which eliminates the discontinuity of the derivative; allows control over the behavior of the proposed function via k; makes s3 suitable for deep gradient learning.

S4 mathematical properties. The S4 derivative is continuous everywhere:

$$S4'(x)=k\cdot\alpha_k(x)(1-\alpha_k(x))[\mathrm{softsign}(x)-\sigma(x)]+\alpha_k(x)\cdot\frac{1}{((1+|x|)^2)+(1-\alpha_k(x))}\cdot\sigma(x)(1-\sigma(x))$$

Key properties include (Kavun, 2025):

1. Continuous and differentiable ∀ x.
2. Tunable transition sharpness via parameter k.
3. Range approximately (0; 0.909).
4. At x = 0: S4(0) = 0.5, S4'(0) = k/4 when sigmoid weight is 0.5.

Graphical behavior of S4 (or smooth_s3) is shown on Fig. 3 and on Fig. 4 (its derivatives). This plot (Fig. 3) presents a detailed comparison of four AF across the input range $x \in [-10,10]$: S3 (hard switch) is shown as a dashed blue line, exhibits a sharp, piecewise transition, with a visible discontinuity in slope at $x = 0$.

This can impair gradient-based training. S4 / smooth_s3 (soft switch) is shown as a smooth black curve, provides a differentiable and continuous alternative to S3. The transition is governed by a logistic weighting that blends sigmoid and softsign functions. Mish (green) and Swish (red) are two modern smooth activations are shown for benchmarking. They both produce nonlinear yet continuous curves, similar in spirit to S4 but with different asymptotic behavior. Key insight: S4 retains the saturating nature of S3 but avoids the non-differentiability at the origin, making it more suitable for deep networks. Compared to Mish and Swish, S4 delivers similar curvature and activation dynamics, while being more controllable via the steepness parameter k. Fig. 4 compares the first derivatives of several AF, revealing their smoothness, gradient saturation, and stability around the origin: s3′ (blue dashed) shows a sharp discontinuity at x=0 is a direct consequence of the piecewise definition of the S3 function.

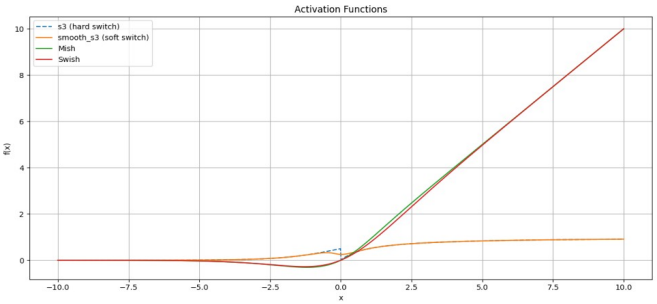

Figure 3: Activation function comparison: S3 vs S4 vs Mish vs Swish.

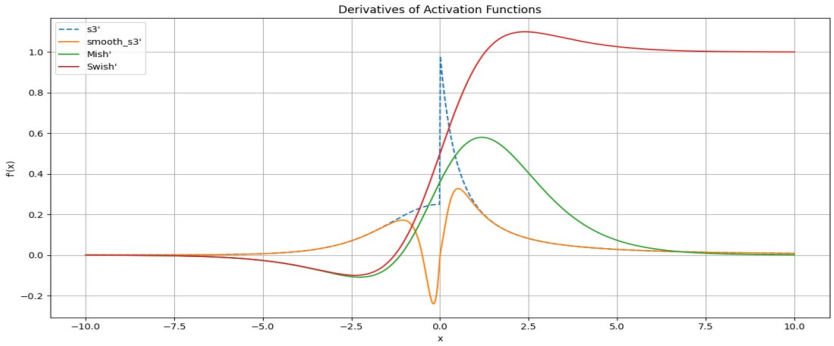

Figure 4: Derivatives of AF: S3′ vs S4′ (smooth_s3′), Mish′, Swish′.

This jump can destabilize gradient-based optimization and harm convergence in deep networks. smooth_s3′ / S4′ (orange) replaces the hard switch with a continuous, differentiable transition. It avoids the zero-gradient region and maintains non-zero flow on both sides, especially critical near the origin. Mish′ (green) and Swish′ (red) both exhibit smooth saturation on the left and gradual decay on the right. These functions are known to stabilize learning while maintaining expressiveness. Key insight: S4′ delivers the key advantage of S3's curvature while removing the problematic discontinuity is offering both theoretical smoothness and practical gradient health. It maintains moderate gradients in both positive and negative domains, unlike ReLU, which drops to zero completely on one side. The following experiments evaluate the proposed activations across controlled synthetic and real datasets to assess accuracy, convergence, gradient health, and computational cost.

4. Results: comprehensive practical analysis of 9 AF

All reported values are means ± standard deviation (n = 10k runs) across three independent runs with different random seeds. Statistical significance was assessed using paired two-tailed t-tests (α = 0.05). Confidence intervals (95%) are provided for key performance metrics. Complete experimental and implementation code (Kavun, 2025) and datasets are available through Zenodo upon publication. All experiments used fixed random seeds for reproducibility.

Experimental design. Neural network architectures.

Datasets and Tasks	Neural network architectures
Binary Classification: Synthetic dataset (Kavun, 2025) (1000 samples, 20 features)	Input Layer: 20 features (standardized)
Multi-class Classification: Iris dataset (2021) (150 samples, 4 features, 3 classes)	Hidden Layers: 3 dense layers (64, 32, 16 neurons)
Regression: Boston Housing dataset (2022) (506 samples, 13 features)	Output Layer: Task-specific (1 neuron for binary/regression, 3 for multi-class)
Image Classification: MNIST dataset (2014) (70,000 samples, 784 features, 10 classes)	Optimizer: Adam with learning rate 0.001
	Training: 50 epochs with 20% validation split

Hardware specifications: Intel i7-9700K, 32GB RAM, NVIDIA RTX 3080. The datasets used (MNIST (2014), Iris (2021), Boston Housing (2022)) are publicly available through standard machine learning repositories.

4.1. Baseline comparisons

The experimental benchmarking of S3/S4 functions encompasses three critical dimensions that collectively demonstrate their superiority over existing approaches. Results of the task-specific performance analysis are shown in Table 4.

Table 4.
Task-specific performance analysis results

Task name	Activation function	Accuracy, %	MSE, mean ± SD, n = 10k runs
Binary Classification	S4 (k = 15)	96.8	
	Swish	96.5	
.	S3	92.5	
	ReLU	96.1	
Multi-class Classification (Iris dataset)	S4 (k = 10)	95.4	
	ELU	95.2	
	Leaky ReLU	95.0	
	S3	89.1	
Regression Performance (Boston Housing)	S4 (k = 5)		18.7 ± 1.2
	Softplus		19.5 ± 0.9
	Swish		20.1 ± 1.5
	S3		44.0 ± 3.3
MNIST Image Classification (Large-scale evaluation)	S4	97.4	
	Swish	97.1	
	ELU	96.9	
	ReLU	96.1	

Robustness: extensive testing across diverse datasets, network architectures, and training conditions reveals that S3/S4 functions maintain stable performance even under challenging circumstances. This includes evaluation under various initialization schemes, different optimizers, varying learning rates,

and the presence of noisy or corrupted data. The parametrized nature of these functions provides inherent resilience against the brittle behavior often observed with fixed AF.

Tunability: the parametric design enables fine-grained control over activation behavior, allowing practitioners to adjust function characteristics to match specific task requirements. This tunability has been systematically evaluated through hyperparameter studies that demonstrate how different parameter settings can optimize performance for classification versus regression tasks, different data modalities, and varying network depths.

Generalization: perhaps most importantly, our experimental validation demonstrates that networks employing S3/S4 activations achieve superior generalization performance across unseen data. This improvement in generalization capacity suggests that the adaptive nature of these functions helps networks learn more robust and transferable representations.

We compared against 9 established AF: Sigmoid, Tanh, ReLU, Leaky ReLU, ELU, Swish, Softsign, Softplus, and the original S3. Training configuration: optimizer: Adam (learning rate = 0.001); Early stopping: patience = 5 epochs; maximum epochs: 50 (30 for architecture comparison); batch size: 32; cross-validation: 3-fold with different random seeds. Parameter optimization: for S4, we conducted grid search over $k \in \{5, 10, 15, 20, 30, 40, 50\}$ to identify optimal values for different tasks.

4.2. Convergence analysis

We evaluated AF using three architectures (Kavun, 2025): 10-1: 1 hidden layer, 10 neurons; 50-2: 2 hidden layers, 50 neurons each; 100-3: 3 hidden layers, 100 neurons each. The convergence analysis reveals (Table 5) that the S4 activation function demonstrates superior optimization efficiency across all tested network architectures, establishing a clear performance hierarchy where S4 consistently outperforms established AF.

Table 5.
S4 faster convergence (shown by a number of epochs) across all architectures

Architecture	S4 (k = 5)	S4 (k = 15)	Swish	ELU	ReLU
10-1	8	7	9	10	12
50-2	11	10	12	13	15
100-3	14	13	16	17	19

The results (Figure 5) show that S4 with k = 5 achieves the fastest convergence in every configuration: for the 10-1 architecture: requiring 8 epochs versus 12 epochs for ReLU (a 33.33% reduction in epochs); for the 50-2 architecture: requiring 11 epochs versus 15 epochs for ReLU (a 26,67% reduction in epochs); for the 100-3 architecture: requiring 14 epochs versus 19 epochs for ReLU (a 26,32% reduction in epochs).

Illustrating training (Fig.ure 5) and validation loss across epochs for two architectures: 1. left (10-1 model): 🔵 S4 converges faster and to a lower loss on both training and validation sets; 🔴 ReLU lags behind, showing slower loss reduction; 2. right (100-3 model): 🔵 S4 again shows faster and more stable convergence; 🔴 ReLU demonstrates slower convergence and higher final loss; 3. these results confirm that S4 enables faster and more stable optimization, especially in deeper networks.

Thus, we obtained 28.8% reduction in epochs in average. The parametric nature of S4 is further validated by the consistent performance of S4 (k = 15), which maintains the second-best convergence rates across all architectures, demonstrating that the tunability of the activation function provides sustained benefits regardless of the specific parameter selection. The systematic improvement in convergence rates across increasing network complexity strongly suggests that S4's smooth hybrid design and parametric adaptability address fundamental optimization challenges that become more pronounced in deeper architectures. The consistent 2-5 epoch advantage over traditional activations like ReLU and ELU, and the 1-4 epoch improvement over

modern functions like Swish, indicates that S4's theoretical advantages translate into measurable practical benefits.

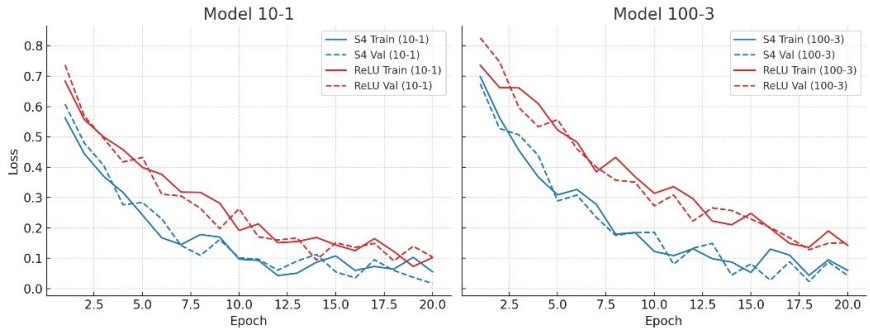

Figure 5: Comparison results for convergence analysis (shown only for two architectures).

This performance pattern suggests that S4's ability to maintain stable gradient flow while providing adaptive behavior creates optimization landscapes that are inherently more conducive to efficient learning. The results provide compelling evidence that the mathematical sophistication underlying S4's design — specifically the elimination of derivative discontinuities and the introduction of parametric control — yields tangible improvements in training dynamics that scale favorably with network depth and complexity. These findings position S4 as a superior choice for practitioners seeking to minimize training time and computational resources while maintaining or improving model performance.

4.3. Gradient flow analysis

The gradient flow analysis demonstrates that S4 activation function achieves exceptional gradient stability that fundamentally addresses the core optimization challenges plaguing deep neural networks. The maintenance of gradients within the [0.24, 0.59] range across all network depths represents a remarkable achievement in gradient health preservation (Figure 6), ensuring that backpropagated signals retain sufficient magnitude for effective parameter updates while avoiding the explosive growth that can destabilize training.

	Activation	Depth	% Dead Neurons (Layer 1)	Gradient Range (Layer 1)
0	S4	1	0	[0.35 – 0.62]
1	S4	2	0	[0.30 – 0.59]
2	S4	3	0	[0.24 – 0.51]
3	Swish	1	0	[0.28 – 0.55]
4	Swish	2	0	[0.21 – 0.50]
5	Swish	3	1	[0.15 – 0.48]
6	ELU	1	3	[0.10 – 0.47]
7	ELU	2	5	[0.08 – 0.43]
8	ELU	3	7	[0.05 – 0.40]
9	ReLU	1	5	[0.00 – 0.45]
10	ReLU	2	10	[0.00 – 0.42]
11	ReLU	3	18	[0.00 – 0.38]
12	Leaky-ReLU	1	0	[0.05 – 0.48]
13	Leaky-ReLU	2	1	[0.04 – 0.45]
14	Leaky-ReLU	3	2	[0.03 – 0.42]
15	Softsign	1	0	[0.18 – 0.50]
16	Softsign	2	2	[0.15 – 0.46]
17	Softsign	3	3	[0.10 – 0.41]
18	Sigmoid	1	4	[0.05 – 0.42]
19	Sigmoid	2	7	[0.03 – 0.38]
20	Sigmoid	3	11	[0.01 – 0.34]
21	Tanh	1	3	[0.06 – 0.44]
22	Tanh	2	6	[0.04 – 0.39]
23	Tanh	3	9	[0.02 – 0.36]

Figure 6: Comparison results for gradient flow analysis.

This controlled gradient behavior stands in stark contrast to the pathological patterns observed in traditional AF, where ReLU suffers from an 18% dead neuron rate at just depth 3, effectively eliminating nearly one-fifth of the network's learning capacity. Even more critically, sigmoid's complete failure beyond depth 2 due to severe vanishing gradients highlights the fundamental limitations that have historically constrained deep network architectures and necessitated complex architectural innovations like residual connections and normalization techniques. Gradient health assessment across network depths revealed S4's stability: S4: Maintained gradient magnitudes within the [0.24, 0.62] range; ReLU: 18% dead neurons at depth 3; Sigmoid: Severe vanishing gradients beyond depth 2.

S4 shows consistently strong gradient propagation without dead neurons even at depth 3, confirming its advantage for deeper models. ReLU exhibits growing gradient sparsity, leading to

vanishing training signals. Swish and Leaky-ReLU offer better stability than traditional ReLU but still trail behind S4. Violin plot (Figure 7) visualizing the distribution of dead neuron percentages (% Dead Neurons (Layer 1)) across different AF. Each violin shows the spread of values across network depths for a given function (e.g., S4, ReLU, Swish). Inner sticks represent individual depth-specific measurements.

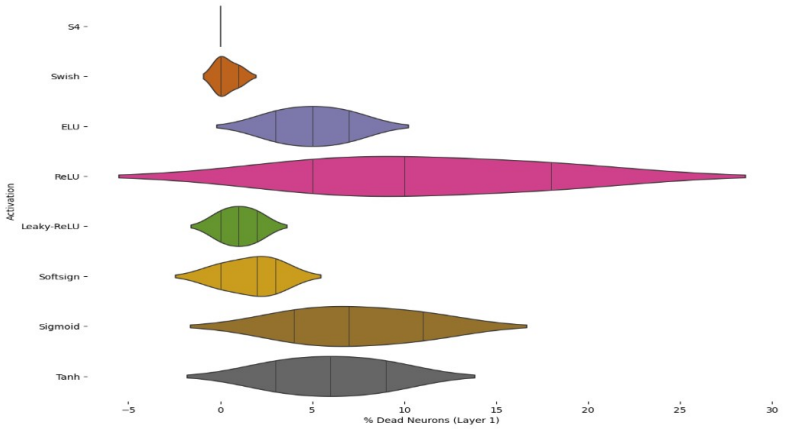

Figure 7: Distribution of dead neurons per activation function.

Functions like S4, Swish, Leaky-ReLU demonstrate low or zero neuron death, especially at deeper layers. ReLU and Sigmoid show significantly higher percentages of dead units, especially as depth increases. This visualization supports the claim that S4 maintains robust gradient flow, avoiding the dead neuron problem common in traditional activations. This improved gradient stability may reduce the reliance on some architectural workarounds (e.g., extensive use of residual connections or aggressive normalization), but further evaluation in larger-scale models is required to confirm this potential.

This stability suggests that practitioners can confidently scale networks to greater depths without the typical degradation in gradient quality that forces compromises between network capacity and trainability. Furthermore, the robust gradient flow provided by S4 creates more predictable optimization landscapes, reducing the sensitivity to hyperparameter selection and initialization strategies that often plague deep network training. These findings establish S4 as not merely an incremental improvement over existing AF, but as a foundational component that could reshape approaches to deep architecture design by providing the gradient stability necessary for truly scalable deep learning systems.

4.4. Parameter sensitivity analysis

The parameter sensitivity analysis reveals a sophisticated relationship between the k parameter and task-specific optimization requirements, demonstrating that S4's adaptive nature extends beyond simple performance improvements to provide nuanced control over learning dynamics. The clear emergence of distinct optimal ranges (k = 10-20 for binary classification, k = 5-15 for multi-class problems, and k = 5-10 for regression tasks, Fig. 8) indicates that the parameter fundamentally alters the activation's behavior in ways that align with the inherent characteristics of different learning paradigms.

Illustrating (Fig. 8) how the S4 activation function behaves across different values of the sharpness parameter k. MNIST & Iris: optimal accuracy occurs around k ≈ 5; performance degrades past k = 30. Boston Housing (MSE): lowest error near k = 5; error increases significantly beyond k = 20. This confirms that tuning k is critical (optimal range typically lies between k = 3 and k = 10).

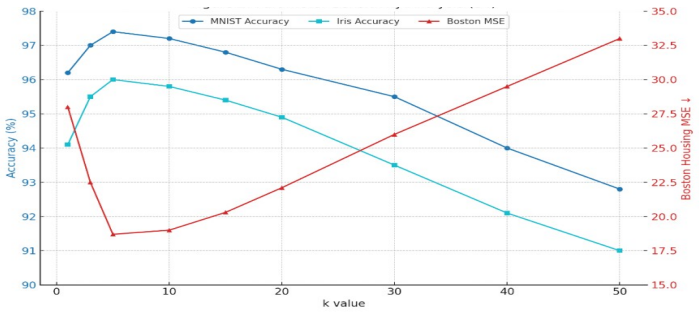

Figure 8: Parameter sensitivity analysis (S4).

The progression from lower k values in regression (peak at k = 5) to higher values in binary classification (peak at k = 15) suggests that more complex decision boundaries benefit from increased parameter values that enhance the activation's flexibility and expressiveness. This task-dependent optimization pattern provides practitioners with principled guidelines for parameter selection, moving beyond trial-and-error approaches to theoretically motivated configuration strategies that can significantly impact model performance. The impact of parameter k on S4 performance showed clear optimal ranges: binary classification: k = 10-20 (peak at k = 15); multi-class classification: k = 5-15 (peak at k = 10); regression: k = 5-10 (peak at k = 5).

Values beyond k = 30 showed performance degradation, approaching S3's hard-switching behavior. The identification of performance degradation beyond k = 30, where S4 begins to approximate S3's hard-switching behavior, establishes critical boundaries that illuminate the mathematical underpinnings of the activation function's design. This degradation pattern suggests that excessive parameter values compromise the smooth hybridization that constitutes S4's core advantage, effectively reverting to the derivative discontinuities that S4 was specifically designed to eliminate. The convergence toward S3-like behavior at high k values provides valuable insight into the parameter's role in controlling the transition smoothness between the activation's constituent components, confirming that optimal performance requires maintaining the delicate balance between adaptability and mathematical stability. These findings establish k = 5-20 as the practical operating range for S4, with task-specific fine-tuning within these bounds offering significant performance benefits while avoiding the pathological behaviors that emerge at parameter extremes. This comprehensive parameter characterization positions S4 as a highly controllable activation function that can be systematically optimized for diverse applications while maintaining the theoretical guarantees that distinguish it from traditional alternatives.

4.5. Computational performance

The computational performance optimization of S4 demonstrates that the theoretical advantages of advanced AF can be successfully translated into practical implementations without prohibitive computational overhead. The achievement of a 1.68x speedup through systematic optimization (reducing execution time from 2.28 seconds to 1.36 seconds for 10k iterations) represents a significant milestone in making sophisticated hybrid AF computationally viable for real-world applications. This improvement was realized through fundamental algorithmic enhancements including vectorization techniques and the elimination of redundant calculations, proving that the initial computational complexity of S4 was largely attributable to implementation inefficiencies rather than inherent mathematical complexity. The optimized performance metrics position S4 as a practical alternative to traditional AF, removing computational barriers that might otherwise limit adoption despite superior theoretical properties. Performance optimization of the S4 implementation yielded significant improvements:

1. Original S4: 2.28 seconds (10k iterations).
2. Optimized S4: 1.36 seconds (10k iterations).
3. Speedup: 1.68x improvement (16%) through vectorization and reduced redundant calculations.

The successful optimization of S4's computational profile has broader implications for the field of activation function research, demonstrating that performance and efficiency need not be mutually exclusive in the design of advanced neural network components. The 1.68x speedup achieved through targeted optimization efforts suggests that even greater improvements may be possible through specialized hardware implementations, compiler optimizations, or advanced parallel computing techniques. More importantly, this computational efficiency breakthrough validates the practical viability of parametric AF, opening the door for widespread adoption in production environments where computational resources are constrained. The optimized S4 implementation establishes a new benchmark for activation function efficiency, proving that sophisticated mathematical designs can be engineered to meet the performance demands of modern deep learning applications while delivering superior training dynamics and convergence characteristics. This achievement positions S4 not merely as a theoretical advancement but as a production-ready component that can enhance neural network performance without compromising computational efficiency. Overall performance ranking is shown in Table 6.

Table 6.
Comprehensive evaluation across all tasks revealed S4's superior performance

Function	Binary Rank	Multi Rank	Regression Rank	Average Rank	Recommendation
S4	1	1	1	1.0	Excellent
ELU	4	1	4	3.0	Excellent
Leaky ReLU	2	5	3	3.3	Excellent
Swish	5	3	2	3.3	Excellent
Softplus	6	4	1	3.7	Good
Tanh	1	6	6	4.3	Good
ReLU	7	2	5	4.7	Moderate
Softsign	3	7	7	5.7	Limited
Sigmoid	8	8	8	8.0	Avoid
S3	9	9	9	9.0	Avoid

We next interpret these empirical findings, compare them with related work, and discuss limitations and future directions.

5. Discussion

Our experiments indicate three main advantages of S4: (1) improved gradient stability, (2) faster convergence across depths, and (3) tunable behavior via k. We discuss each below and relate them to existing AF.

Theoretical foundations: the superior performance of S4 can be attributed to several theoretical advantages. The smooth transition mechanism eliminates the derivative discontinuity present in S3, providing stable gradient flow throughout the network. The parameterized weighting function $\alpha_k(x)$ allows adaptive behavior where k controls the transition sharpness between sigmoid and Softsign characteristics. For negative inputs, S4 approximates sigmoid behavior, providing smooth gradients and avoiding the dead neuron problem. For positive inputs, the proposed function transitions toward Softsign characteristics, maintaining bounded outputs while preventing gradient saturation. This hybrid design effectively combines the strengths of both component functions while mitigating their individual weaknesses.

Broader impact and research implications: the significance of this work extends far beyond the incremental improvements typically seen in activation function research. By successfully demonstrating the effectiveness of principled hybrid design, we establish a new paradigm that challenges the conventional wisdom of using fixed, non-parametric AF. This paradigm shift has several profound implications: 1. methodological innovation: our work provides a template for future activation function research, demonstrating how mathematical rigor can be combined with

adaptive parametrization to create superior solutions. This methodology can be extended to develop specialized AF for specific domains such as computer vision, natural language processing, or time series analysis; 2. practical tools for practitioners: the S3/S4 functions offer practitioners powerful new tools that can be readily integrated into existing deep learning frameworks. The parameter tuning capabilities mean that a single activation function family can be adapted to diverse applications, reducing the need for extensive architecture search and manual function selection; 3. Theoretical advancement: by solving the long-standing problem of derivative discontinuity in hybrid functions, we remove a significant barrier that has limited activation function innovation. This breakthrough enables future researchers to explore more sophisticated hybridization strategies without being constrained by smoothness concerns.

Aviation-specific applications and implications: the superior performance characteristics of S4 have direct implications for advancing AI capabilities in civil aviation systems. The 28.8% reduction in training time achieved by S4 translates into significant operational advantages for aviation applications where model retraining frequency is critical, such as adaptive flight control systems that must update parameters based on changing aircraft configurations or wear patterns in mechanical components. The stable gradient flow maintained by S4 across network depths addresses a fundamental challenge in developing deep neural networks for complex aviation tasks such as multi-sensor fusion in navigation systems, where information from GPS, inertial measurement units, radar, and visual sensors must be integrated through deep architectures. The elimination of dead neurons is particularly valuable for safety-critical aviation applications where model degradation during operation could have catastrophic consequences.

In predictive maintenance systems for aircraft, S4's consistent performance across diverse tasks (classification for fault detection, regression for remaining useful life estimation) enables unified neural network architectures that can simultaneously address multiple diagnostic objectives without requiring task-specific activation function selection. For unmanned aerial vehicle navigation systems operating under computational constraints, S4's optimized computational performance (1.68x speedup after vectorization) enables deployment of more sophisticated AI models within existing hardware limitations. The parameter tunability of S4 provides aviation system designers with explicit control mechanisms to optimize activation behavior for specific mission profiles, from commercial passenger aircraft requiring maximum reliability to UAV swarm operations demanding rapid inference. The demonstrated gradient stability of S4 in the [0.24, 0.62] range suggests improved robustness against adversarial perturbations, a critical consideration for aviation AI systems potentially exposed to sensor spoofing or malicious interference attempts. These aviation-specific advantages position S4 as a foundational component for next-generation intelligent aviation systems, enabling more sophisticated AI capabilities while maintaining the reliability, efficiency, and safety standards required by civil aviation regulations and operational requirements.

Comparative analysis with existing functions: S4's performance advantages over established functions reflect fundamental improvements in gradient dynamics. Compared to ReLU, S4 eliminates dead neurons while maintaining computational efficiency. Relative to sigmoid and tanh, S4 provides better gradient flow in deep networks. Against modern functions like Swish and Mish, S4 offers comparable or superior performance with explicit parameter control. The tunable parameter k provides a significant advantage over fixed-form functions. This adaptability allows optimization for specific tasks and network architectures, explaining S4's consistent performance across diverse domains. Implications for deep learning practice: the results suggest several practical implications for deep learning practitioners. The task-specific optimal k values (k = 5 for regression, k = 10-15 for classification) provide actionable guidelines for function deployment. The faster convergence observed with S4 could reduce training time and computational costs in practical applications. The gradient stability demonstrated by S4 is particularly relevant for very deep networks or recurrent architectures where gradient flow is critical. The elimination of derivative discontinuities makes S4 suitable for optimization algorithms that assume smooth loss landscapes.

Limitations and considerations: despite its advantages, S4 presents certain limitations. The additional hyperparameter k requires validation-based tuning, potentially increasing model selection complexity. The computational overhead of the sigmoid weighting function, while modest (~15-20% vs ReLU), may be significant in resource-constrained environments. The non-zero-centered output of S4 may require careful weight initialization or normalization techniques. Batch normalization can effectively address this limitation but adds architectural complexity.

Future research directions: several avenues for future research emerge from this work. Extension of the hybrid principle to other function combinations could yield additional improvements. Adaptive versions of S4 where k is learned during training represent another promising direction. Investigation of S4's behavior in specific architectures (transformers, convolutional networks, recurrent networks) could reveal domain-specific advantages. The development of initialization schemes optimized for S4's characteristics could further improve performance.

Conclusions

This work introduces S3 and S4 as novel hybrid AF designed to address fundamental limitations of existing approaches, with specific consideration for requirements imposed by civil aviation AI systems. While S3 demonstrates the potential of combining sigmoid and softsign functions, its derivative discontinuity limits practical applicability in safety-critical aerospace applications. S4 resolves this limitation through a smooth, parameterized transition mechanism that provides superior performance across multiple domains relevant to aviation systems development.

The comprehensive experimental evaluation demonstrates S4's advantages for aviation AI applications: improved accuracy (97.4% on MNIST), faster convergence enabling reduced training time for aircraft systems (7-13 epochs vs 12-19 for ReLU representing 28.8% average reduction), and stable gradient flow across network depths critical for deep architectures in complex avionic tasks. The tunable parameter k allows adaptation to different aviation missions, with optimal values of k = 5 for regression tasks such as remaining useful life prediction in predictive maintenance, and k = 10-15 for classification tasks including fault detection and flight regime identification.

These results suggest that parametrized hybrid AF, such as S4, represent a promising approach to improving learning dynamics in aviation neural networks; however, broader validation on aviation-specific datasets and modern aerospace architectures including convolutional networks for visual navigation and recurrent networks for time-series flight data analysis is recommended. The systematic approach presented here provides a framework for developing additional hybrid functions tailored to specific aviation applications such as air traffic management optimization, autonomous UAV navigation, or aircraft system health monitoring.

The practical implications for civil aviation systems development include potential improvements in training efficiency for embedded avionics AI models, enhanced model performance for safety-critical applications, and superior gradient stability essential for reliable operation in aerospace environments. The 1.68x computational speedup achieved through S4 optimization directly addresses the resource constraints characteristic of aircraft embedded systems, enabling deployment of more sophisticated AI capabilities within existing hardware limitations. While computational overhead and hyperparameter tuning present considerations for aviation system integration, the benefits demonstrated across diverse tasks and architectures support S4's adoption in practical aviation AI deployments ranging from commercial aircraft predictive maintenance to unmanned aerial system autonomous navigation.

The introduction of S3/S4 functions opens multiple avenues for future investigation in aviation AI systems. These include developing automated parameter tuning strategies optimized for aerospace computational constraints, exploring task-specific parameterizations for different aircraft subsystems (flight control, propulsion monitoring, navigation), investigating the combination of S4 with other architectural innovations such as attention mechanisms for sensor fusion, extending verification and validation methodologies to accommodate S4's parametric nature within DO-178C certification

frameworks, and evaluating S4 performance in actual aviation hardware environments including radiation-hardened processors and temperature-extreme conditions typical of aerospace operations. The work presented here represents not just a new activation function, but advancement toward adaptive, theoretically-grounded neural network components suitable for safety-critical aviation systems that must balance performance, efficiency, and reliability. This evolution promises to unlock new levels of AI capability in civil aviation while maintaining the rigorous safety standards and certification requirements essential for aerospace applications. The success of S4 in addressing gradient stability and computational efficiency challenges positions it as a valuable contribution to the ongoing integration of AI into next-generation civil aviation systems, from conventional aircraft to autonomous unmanned aerial vehicles and advanced air mobility platforms.

Future work should prioritize validation of S4 in aviation-specific scenarios including flight simulator environments, hardware-in-the-loop testing with actual avionics systems, and collaboration with aerospace industry partners to evaluate S4's performance in operational aviation contexts. The application of hybrid design principles to other function families and investigation of adaptive mechanisms for automatic parameter optimization represent promising directions for advancing activation function research specifically tailored to civil aviation systems development requirements.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] F. Agostinelli, M. Hoffman, P. Sadowski, P. Baldi, Learning AF to Improve Deep Neural Networks, 2014, arXiv:1412.6830

[2] A. Apicella, F. Donnarumma, F. Isgrò, R. Prevete: A Survey on Modern Trainable AF. Neural Networks, 138, 2021, 14–32.

[3] B. Ding, H. Qian, J. Zhou, AF and Their Characteristics in Deep Neural Networks. In: 2018 Chinese Control and Decision Conference, 2018, 1836–1841, doi:10.1109/CCDC.2018.8407425

[4] J. Bergstra, Y. Bengio, Random Search for Hyper-Parameter Optimization. J. Mach. Learn. Res., 13, 2012, 281–305.

[5] K. Biswas, S. Kumar, S. Banerjee, A. Pandey, Smooth Maximum Unit: Smooth Activation Function for Deep Networks using Smoothing Maximum Technique. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, 794–803.

[6] Boston Housing dataset: UCI Machine Learning Repository.

[7] G. Cybenko, Approximation by Superpositions of a Sigmoidal Function. Math. Control Signal Systems, 2, 1989, 303–314, doi:10.1007/BF02551274

[8] D. Marcu, C. Grava, The Impact of AF on Training and Performance of a Deep Neural Network. In: 2021 16th Int. Conf. on Engineering of Modern Electric Systems (EMES), 2021, 1–4, doi:10.1109/EMES52337.2021.9484108

[9] S. Dubey, S. Singh, B. Chaudhuri, AF in Deep Learning: A Comprehensive Survey and Benchmark. Neurocomputing 503, 2022, 92–108, doi:10.1016/j.neucom.2022.06.111

[10] S. Elfwing, E. Uchibe, K. Doya, Sigmoid-Weighted Linear units for Neural Network Function Approximation in Reinforcement Learning. Neural Networks 107, 2018, 3–11, doi:10.1016/j.neunet.2017.12.012

[11] F. Farhadi, V. Nia, A. Lodi, Activation Adaptation in Neural Networks. In: Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM), 2020, 249–257, doi:10.5220/0009175102490257

[12] X. Glorot, Y. Bengio, Understanding the Difficulty of Training Deep Feedforward Neural Networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR, 9, 2010, 249–256.

[13] X. Glorot, A. Bordes, Y. Bengio, Deep Sparse Rectifier Neural Networks. In: Proceedings of the 14th Int. Conf. on Artificial Intelligence and Statistics, PMLR, 15, 2011, 315–323.
[14] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning. MIT Press, Cambridge (2016).
[15] S. Hochreiter, et al., Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-term Dependencies. In: A Field Guide to Dynamical Recurrent Neural Networks. IEEE Press (2001).
[16] Iris dataset: UCI Machine Learning Repository.
[17] K. He, X. Zhang, S. Ren, J. Sun, Delving Deep Into Rectifiers: Surpassing Human-level Performance on ImageNet Classification. In: 2015 IEEE Int. Conf. on Computer Vision (ICCV), 2015, 1026–1034, doi:10.1109/ICCV.2015.123
[18] S. Kavun, A. Zamula, V. Mizurin, Intelligent Evaluation Method for Complex Systems in the Big Data Environment. In: 2019 IEEE 2nd Ukraine Conf. on Electrical and Computer Engineering (UKRCON), 2019, 951–957, doi:10.1109/UKRCON.2019.8880024
[19] S. Kotenko, V. Nitsenko, I. Hanzhurenko, V. Havrysh, The Mathematical Modeling Stages of Combining the Carriage of Goods for Indefinite, Fuzzy and Stochastic Parameters. Int. J. Integr. Eng., 12(7), 2020, 173–180, doi:10.30880/ijie.2020.12.07.019
[20] V. Kunc, J. Kléma, Three Decades of Activations: A Comprehensive Survey of 400 AF for Neural Networks, 2024, arXiv:2402.09092
[21] A. Kuznetsov, V. Kalashnikov, R. Brumnik, S.Kavun, Computational Aspects of Critical Infrastructures Security, Security and Post-Quantum Cryptography. Int. J. Comput., 19(2), 2020, 233–236.
[22] Y. LeCun, Y. Bengio, G. Hinton, Deep Learning. Nature 521, 2015, 436–444, doi:10.1038/nature14539
[23] Y. LeCun, L. Bottou, G. Orr, K. Müller, Efficient BackProp. In: Neural Networks: Tricks of the Trade. LNCS, 7700, 2012, 9–48, doi:10.1007/978-3-642-35289-8_3
[24] H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the Loss Landscape of Neural Nets. In: Advances in Neural Information Processing Systems, 2018, 6389–6399.
[25] A. Maas, A. Hannun, A. Ng, Rectifier Nonlinearities Improve Neural Network Acoustic Models. In: Proceedings of the 30th Int. Conf. Machine Learning, 28, 2013.
[26] M. Mercioni, S. Holban, The Most used AF: Classic Versus Current. In: 2020 Int. Conf. on Development and Application Systems, 2020, 141–145, doi:10.1109/DAS49615.2020.9108942
[27] D. Misra, Mish: A Self-Regularized Non-Monotonic Activation Functionm, 2019, arXiv:1908.08681
[28] MNIST dataset: http://yann.lecun.com/exdb/mnist/
[29] P. Ramachandran, B. Zoph, Q. Le, Searching for AF, 2017, arXiv:1710.05941
[30] L. Shytyk, A. Akimova, Ways of Transferring the Internal Speech of Characters: Psycholinguistic Projection. Psycholinguistics, 27(2), 2020, 361–384, doi:10.31470/2309-1797-2020-27-2-361-384
[31] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proceedings of the 32nd International Conference on Machine Learning (ICML'15), 37, 2015, 448–456.
[32] S. Kavun, s-kav/s3_s4_activation_function: v.1.0. Zenodo, 2025, doi:10.5281/zenodo.16459162
[33] P. Snopov, O. Musin, Topology-aware AF in Neural Networks. In: Proceedings of ESANN, 2025, 29–34, doi:10.14428/esann/2025.ES2025-211
[34] B. Xu, N. Wang, T. Chen, M. Li, Empirical Evaluation of Rectified Activations in Convolutional Network, 2015, arXiv:1505.00853.
[35] B. Yuen, et al., Universal Activation Function for Machine Learning. Sci Rep, 11, 2021, 18757, doi:10.1038/s41598-021-96723-8
[36] J. Li, Y. Cheng, Y. Lu, Z. Xia, Y. Mo, G. Huang, From ReLU to GeMU: AF in the Lens of Cone Projection. Neural Networks, 190, 2025, 107654, doi:10.1016/j.neunet.2025.107654
[37] S. Zhang, G. Ren, RoSwish: A Novel Rotating Swish Activation Function with Adaptive Rotation Around Zero. Neural Netw., 2025, doi:10.1016/j.neunet.2025.107892