Contrast-Enhanced Gating in GRUs for Robust Low-Data Sequence Learning

Barathi Subramanian[1], Rathinaraja Jeyaraj[1], Anand Paul[2]
[1]Stanford University, USA, [2]LSU Health Sciences Center New Orleans, USA
{barathi1, rajaj}@stanford.edu, apaul4@lsuhsc.edu

Abstract

Activation functions govern how recurrent networks regulate and transmit information across temporal dependencies. Despite advances in sequence modelling, gated recurrent units (GRUs) still depend on the standard sigmoid and tanh nonlinearities, which can produce weak gate separation and unstable learning, particularly when training data are limited. We introduce squared sigmoid-tanh (SST), a parameter-free activation that squares the gate nonlinearity to increase contrast between near-zero- and high-activations, thereby promoting sharper information filtering during GRU updates. We incorporate SST into GRU gating and evaluate it across low-data settings spanning sign language recognition, human activity recognition, and time-series forecasting and classification. Across tasks, SST-GRU consistently surpasses standard sigmoid/tanh GRU, with the largest improvements observed in the smallest-data domains, while adding negligible computational cost. We further examine gate activation statistics and training dynamics, showing that SST improves training stability, which aligns with its performance gains in data-scarce settings. SST is a parameter-free modification that complements more complex architectural advances by improving gating selectivity in low-data sequence learning.

1. Introduction

Activation functions determine how neural networks transform intermediate signals and are central to representational capacity and optimization stability in deep learning [1], [2]. In feedforward networks, piecewise-linear activation functions such as ReLU [3] and its variants [4], [5] are widely used due to their simple piecewise behaviour and favourable gradient propagation, enabling deep models to train reliably at scale. In contrast, recurrent neural networks (RNNs) must propagate information through time, where training is notoriously sensitive to vanishing and exploding gradients [6]. Early analyses showed that gradient-based learning becomes increasingly difficult as temporal dependencies grow, producing brittle optimization when long-range credit assignment is required. Later work further formalized these failure modes and proposed practical stabilizers (for example, gradient clipping and regularization strategies) to mitigate exploding or vanishing gradients in recurrent model training [7]. Gated architectures such as long short-term memory (LSTM) [8] was introduced to address these issues by improving error flow through time, and remains foundational sequence models. To handle low-data availability and reduce computational complexity, gated recurrent unit (GRU)-style architectures[9], [10] were introduced to learn sparse patterns in the dataset. GRU relies primarily on sigmoid and tanh nonlinearities [11], [12] to learn temporal dependencies effectively. However, in low-data regimes, these saturating nonlinearities may reduce separability between "open" and "closed" gate states, limiting the model's ability to sharply filter or preserve information, resulting in reduced downstream performance [6], [13]. This limitation becomes particularly relevant in compact sequential models deployed in resource-constrained environments such as gesture recognition pipelines [14] and sensor-based time-series modelling. Rather than increasing model capacity or introducing new architectural components, this work investigates whether improving gate selectivity alone can enhance sequence learning under limited data conditions.

Modern transformer-family models [15], [16] have become strong general-purpose sequence learners, but they introduce practical trade-offs that matter in the target applications. First, self-attention can be computationally expensive for long sequences [9], and many time-series variants focus on architectural changes to reduce parameterization and improve efficiency (e.g., factorized or lightweight designs), which implicitly acknowledges that vanilla transformers can be over-parameterized or inefficient for typical time-series inputs. Second, transformers [16], [17] and hybrid alternatives (including state-space inspired models [18], [21]) show that the field is actively searching for architectures that retain accuracy while reducing compute and improving robustness for time-series applications. Third, evidence is mixed on whether scaling or pretraining alone guarantees superior performance over smaller alternatives in time-series settings, motivating careful choices under limited data and constrained training budgets [19], [27]. Lightweight encoder-

only designs aim to retain accuracy while reducing compute for forecasting at scale. Broader evaluations also report mixed outcomes on whether larger or pretrained transformers consistently dominate smaller alternatives in time-series settings, motivating careful architecture choices when data and training data are limited. In addition, hybrid state-space models [18] highlight the continued push toward models that remain robust and efficient on long sequences.

In applications such as sign language recognition [14], human activity recognition [30], [31], speech recognition [22], [23], and time-series classification/forecasting [33], where datasets are limited, GRU remains attractive because they are compact, stable to train, and fast to deploy. However, data complexity makes GRU underfit the sequential data, which requires enhancing the model's ability to learn complex patterns. This motivates a complementary direction to improve the gating nonlinearity itself while keeping the GRU structure unchanged to improve the downstream task performance. To this end, we propose squared sigmoid-tanh (SST), a simple drop-in activation strategy that squares the outputs of sigmoid and tanh within the GRU gating pathway. Squaring increases the contrast between weak and strong activations (e.g., values near 0 are suppressed further while larger activations are emphasized), which can encourage sharper information filtering through the update dynamics to improve the model generalizability and downstream performance. Importantly, SST does not introduce additional parameters and adds negligible computational overhead, making it compatible with resource-constrained training and deployment. We empirically evaluate SST-powered GRU across multiple low-data tasks and show consistent gains over standard GRU nonlinearities. In summary, the following contributions are presented in this manuscript. We

- introduce, SST, a drop-in nonlinear modification for GRU gating designed for data-constrained sequence learning.
- provide an analysis of how SST changes gate activation behaviour (e.g., gate contrast/distribution) and discuss practical implications for stability and convergence.
- benchmark SST-powered GRU across multiple applications (time-series analysis, sign language and human activity recognition) under controlled low-data protocols.

2. Related Work

Activation functions and optimization: Activation functions strongly influence gradient flow and training stability [6]. Rectifier-based activations became prevalent in deep feedforward networks because they reduce the optimization difficulties caused by saturating nonlinearities, particularly vanishing gradients, while encouraging sparse activations and improved gradient flow [19]. This line of work motivated numerous rectifier variants, including learnable negative-slope units (PReLU) [5] and smooth alternatives such as ELU [26], which explicitly contrasts the contractive behaviour of tanh/sigmoid with rectifier-like units. Beyond hand-designed functions, automated search has also produced drop-in nonlinearities such as Swish [27], which can outperform ReLU in several deep-network settings. These advances, however, largely target feedforward optimization and do not directly resolve the temporal credit-assignment challenges characteristic of recurrent learning.

Recurrent learning and gating under limited data: Training RNNs on long-term dependencies remains difficult due to vanishing and exploding gradients [12], [25]. Practical analyses further show that recurrent optimization can be brittle and often requires stabilizers such as careful initialization and gradient control [2]. Gated models, notably LSTMs, were introduced to improve error flow through time and remain widely used for sequential data [8]. Yet, common gated RNNs still use sigmoid and tanh to implement bounded gating and state updates, meaning saturation effects can still constrain gate contrast and gradient propagation that are amplified when supervision is scarce (small datasets) or sequences are short/noisy [7].

Learned and parameterized activations: Adaptive piecewise linear (APL) units parameterize a flexible piecewise nonlinearity [28]. While such methods increase representational flexibility, they introduce additional learned parameters and typically validated in feedforward settings, leaving open the question of whether simple nonlinear modifications can improve recurrent gating in low-data settings.

Efficient recurrent cells and alternatives to standard RNNs: Recent work also revisits the recurrent cell design to improve efficiency and parallelism. Quasi-RNNs combine convolutional computation with lightweight recurrent pooling to increase throughput while maintaining order sensitivity [29]. Simple recurrent units [20] target parallelizable recurrence and report substantial speedups over LSTMs in NLP tasks [27] modifying the cell architecture. While alternative activations such as Swish and other smooth nonlinearities improve optimization in feedforward networks, recurrent gating still relies predominantly on sigmoid and tanh due to their bounded outputs. This highlights an opportunity to improve gate selectivity through simple contrast-enhancing transformations without increasing model complexity.

While there are attempts to introduce specialized recurrent activations, the existing techniques do not fully address the constraints posed by sequential data, particularly in sparse, low-data regimes. Therefore, we propose SST, explicitly designed to enhance training and generalization in classical GRU, when dealing with small sequential datasets containing sparse patterns. SST squares the outputs of the sigmoid and tanh activation functions, thereby augmenting gradient propagation and signal filtering, which is crucial for memory retention as signals evolve over time. This approach is directly targeted at mitigating the long-range dependency

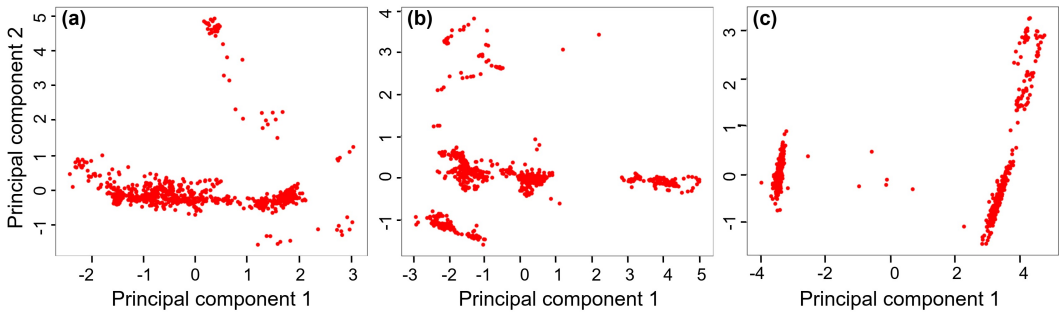


Figure 1. PCA visualization of frames for three distinct ISL classes (a) friend, (b) phone call, and (c) location.

limitations inherent in existing activation schemes. Through our empirical studies, we demonstrate SST's effectiveness in enhancing classical GRU, providing new insights and empirical evidence for an open problem in neural networks.

3. Motivation and Intuition behind SST

In certain cases, the performance of sequential models can be limited when training data exhibit sparse patterns. sigmoid and tanh nonlinearities, due to their saturation behaviour, may suppress gradient flow and reduce sensitivity to subtle variations in temporal signals. This is particularly relevant in tasks such as sign language recognition, where informative patterns may occur infrequently and require careful preservation during learning. To illustrate this challenge, principal component analysis (PCA) is applied to the Indian sign language (ISL) dataset [14], as shown in Figure 1. Three representative sign classes (friend, phone call, location) are selected for visualization. The PCA projections show regions where feature representations are relatively sparse, indicating subtle temporal distinctions that conventional gating nonlinearities may fail to capture. To address this limitation, SST is introduced to improve sensitivity to weak but informative signals. Squaring the activation sharpens contrast between weak and strong gate responses, improving information selectivity in recurrent updates. While alternative sharpening strategies (e.g., $sigmoid(x)^p$) could also modify gate behaviour, the squared form is chosen as a minimal, smooth, and parameter-free transformation that preserves stable optimization.

4. Squared sigmoid-tanh (SST)

The classical GRU plays a crucial role in modeling temporal dependencies, especially in modelling short-term and long-term sequences. A GRU cell contains two main components: a reset gate (r_t) and an update gate (z_t). These gates modulate the flow of information inside the GRU cell using sigmoid activation, controlling whether to forget the previous state or consider the latest input. The output (h_t) is a linear interpolation between the previous activation (h_{t-1}) and the GRU's candidate activation ($\widetilde{h}_t$) that processes the current input and previous state through a tanh layer. Although GRU exhibits a remarkable capacity to perform well in situations where training sequential data is limited [7], the classical GRU struggles to learn sparse patterns in the data and is unable to effectively propagate signals through time and layers [27]. To address this issue, a novel idea, SST is introduced. SST squares the sigmoid activation in the update and reset gates, and the tanh activation in the final dense layer, while the candidate hidden state activation within the GRU remains unchanged, as shown in Figure 2. The idea of SST is that, for instance, after squaring sigmoid activation function, the higher input probability value gets relatively higher than the lower input probability value. This approach enhances model learning efficiency and improves the accuracy of the neural network. In the following subsections, we discuss the properties of the squared sigmoid and squared tanh activation functions to formulate the SST.

4.1. Squared sigmoid (SS)

The SS is achieved by squaring the generic sigmoid function (SF). It is defined by

$$SS(x) = (SF(x))^2, \quad where \quad SF(x) = \frac{1}{1+e^{-x}}. \quad (1)$$

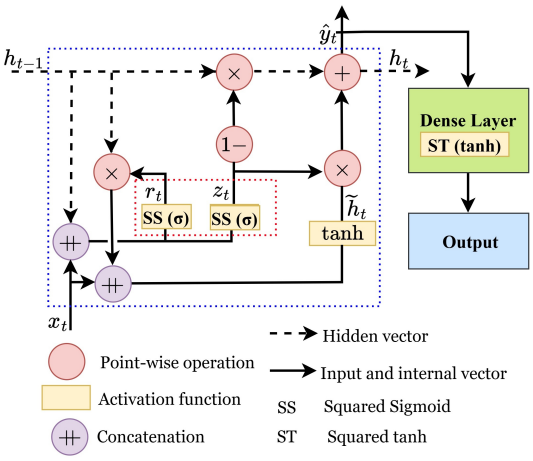


Figure 2. SST-activation function in a GRU cell.

The behaviour of SF and SS is displayed in Figure 3a. The typical SF results in [0,1]. When SS is applied, the higher output probability value of SF is retained relatively higher when compared to the lower value getting very small. For example, $(0.9)^2$ and $(0.1)^2$ result in 0.81 and 0.01, respectively. This suggests that neurons are activated to focus on the larger value in retaining the sparse patterns. To emphasize the applicability of SS as activation function in sequential models, it is essential to discuss the properties of a generic activation function. They are boundedness, nonlinearity, continuity, and differentiability.

(a) Range: SS produces a similar S-shaped curve stretched between 0 and 1.

Proof: To find the range of $SS(x)$, we show that $\forall_{x\in\mathbb{R}}, 0 \leqslant SS(x) \leqslant 1$. Since SF is bounded by [0,1], squaring both sides results in [0,1] stretching the curve.

(b) Nonlinearity: SS introduces more nonlinearity to the model than the traditional SF.

Proof: Rate of change of SS is a function. When $\frac{d}{dx}SS(x)$ is taken, it results in $2 * SF(x) * SF'(x)$, which is nonlinear, as shown in Figure 3a. Hence, the SS has a more pronounced curvature in the middle of the curve compared to the SF. It means that it can map more complex sparse patterns in the data, as SS is more sensitive to small changes in the input.

(c) Continuity: SS is continuous at all points in the curve.

Proof: To prove $SS(x)$ is continuous, we consider the composition rule of continuity which states that if $g(x)$ is continuous at $x = a$ and $h(x)$ is continuous at $x = g(a)$, then the composite function $f(x) = g(h(x))$ is also continuous at $x = a$. In our case, we apply the composition rule to $g(x) = x^2$ and $h(x) = SF(x)$ to show that $SS(x) = g(h(x))$ is continuous. We know that $\forall_x, h(x)$ and $g(x)$ are continuous. Therefore, by the composition rule for continuity, $\forall_x, SS(x) = g(h(x)) = (SF(x))^2$ is also continuous.

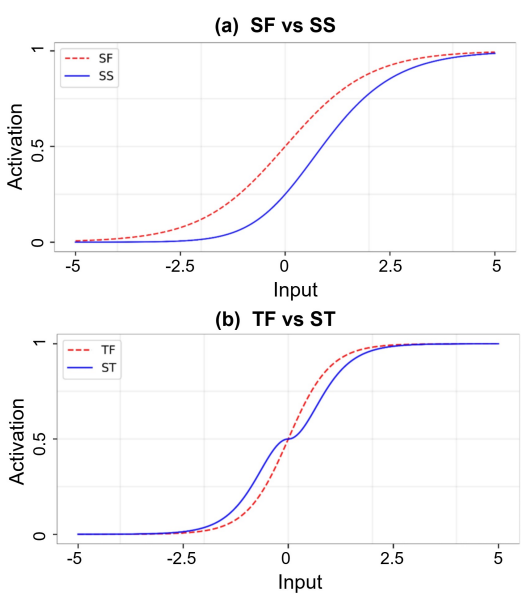


Figure 3. Behaviour of (a) SS and (b) ST.

(d) Differentiability: SS is differentiable.

Proof: To prove $SS(x)$ is differentiable $\forall x$, we show that there exists derivative at every point in its domain. Using the chain rule and the derivative of the logistic function, we find the derivative of SF and SS as follows,

$$SF'(x) = SF(x) * (1 - SF(x)). \tag{2}$$

$$SS'(x) = 2 * SF(x) * SF'(x). \tag{3}$$

Substituting (2) in (3), we get

$$SS'(x) = 2 * (SF(x))^2 * (1 - SF(x)). \tag{4}$$

Since SF is differentiable $\forall x$, the product of two differentiable functions is differentiable. Therefore, SS is also differentiable $\forall x$ in its domain.

4.2. Squared tanh (ST)

ST is obtained by squaring the classical hyperbolic tangent function (TF) and mathematically defined by

$$ST(x) = \begin{cases} -(TF(x))^2 & \text{if } x < 0 \\ (TF(x))^2 & \text{if } x \geqslant 0 \end{cases}, TF(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{5}$$

Here, squaring the -ve value is returned as -ve of the squared value, as given in Eq. (5). Otherwise, the behaviour of ST is not adhered to the classical TF. The behaviour of TF and ST is shown in Figure 3b, in which ST displays much higher nonlinearity away from the origin. As TF is modified, we discuss how it impacts and ensures the properties of activation functions.

(a) Range: ST has a similar S-shaped curve like TF but with a narrow waist in the middle, as shown in Figure 3b and it is bounded to [-1,1].

Proof: To prove the boundedness of ST outcome [-1,1] $\forall x, x \in \mathbb{R}$, consider the following two cases.

- For $x \geqslant 0$, since $TF(x)$ ranges [0,1], $(TF(x))^2$also ranges [0,1].
- For $x < 0$, since $TF(x)$ ranges [-1, 0), we use $-(TF(x))^2$which ranges [-1, 0).

Combining these cases together, Eq. (5) is bounded [-1,1] $\forall_x, x \in \mathbb{R}$.

(b) Nonlinearity: ST is non-linear.

Proof: To prove ST is non-linear, we show that it does not satisfy the property of additivity. It means that $ST(a + b) \neq ST(a) + ST(b)$. For given $a, b \in \mathbb{R}$, nonlinearity is obvious for ST when $a, b \geqslant 0$. Therefore, we provide a clarity on nonlinearity for ST when $a, b < 0$. From Eq. (5), we get $ST(a + b) = -(TF(a + b))^2$ when $a + b <$

0. To prove nonlinearity, the following inequality holds $-(TF(a+b))^2 \neq -(TF(a)+TF(b))^2$. Hence, additivity property is not applicable for the two cases considered above.

(c) Continuity: ST is continuous.

Proof: To prove ST is continuous, it is sufficient to show that the ST is continuous at $x = 0$. Consider the limit of $ST(x)$ as $x \to 0$ from the left and right side of a number line.

- For $x \in \mathbb{R}^-$,
  $\lim_{x\to 0} -(TF(x))^2 = \lim_{x\to 0} -(0)^2 = 0$
- For $x \in \mathbb{R}^+$,
  $\lim_{x\to 0}(TF(x))^2 = \lim_{x\to 0}(0)^2 = 0$

Since both the left and right limits are equal to 0, we can conclude that the ST is continuous at $x = 0$.

(d) Differentiability: ST is differentiable.

Proof: To show that ST is differentiable at any $\mathbb{R}$, let us consider the chain rule, which states that if there is a composite function $f(g(x))$, then the derivative of $f(g(x))$ is given by $f'(g(x)) * g'(x)$. Let $f(x) = TF(x)$ and $h(x) = 2I(x) - 1$, where $I(x)$ is the indicator function. Then the ST can be expressed as $g(x) = f(x)^2 * h(x)$ and the derivative of $g(x)$ is given by

$$g'(x) = 2 * f(x) * f'(x) * h(x) + f(x)^2 * h'(x). \quad (6)$$

Using the fact that $f'(x) = Sech^2(x)$, $h(x)$ is a piecewise constant and $h'(x)$ is 0 almost everywhere (except at $x = 0$), we simplify Eq. (6) as

$$g'(x) = 2TF(x) * Sech^2(x)[2I(x) - 1], \quad if \quad x \neq 0. \quad (7)$$

It turns out that $g(0) = 0$. Note that $Sech^2(x)$ and $2I(x) - 1$ are both continuous functions at any $\mathbb{R}$, so their product is also continuous at any $\mathbb{R}$. Thus ST is differentiable at any $\mathbb{R}$, except possibly at $x = 0$, where we have a corner point.

4.3. SST (SS+ST)

To improve the performance of the classical GRU further, the SST is conceived based on observing and understanding their role in the gating mechanism, as well as their properties discussed in the previous sections. The basic concept of the SST is that to make the small values even smaller and the large values even larger, by squaring the output of SF in r_t and z_t, and TF in the dense layers, such that SST

- improve gradient sensitivity and learning dynamics in sparse regimes.
- improves selective retention of informative signals and helps better recall information from previous t.
- provides a greater degree of nonlinearity, which is particularly important for r_t to capture complex patterns and determine what information should be retained.
- emphasizes the robustness of GRU cells to noise input values by focusing on important information, particularly for z_t, which determines how much of h_{t-1} to retain.

It is important to observe that we did not replace the default TF in $\widetilde{h}_t$ of the GRU, as it can make it difficult to update $\widetilde{h}_t$ over time. Also, the reason for not using the widely used ReLU activation function in the dense layer is to avoid the "zero-gradient" problem for the negative inputs and to provide smooth gradient transitions to improve the overall stability and convergence rate of the neural network. Even though ReLU has been shown to work well in many cases, the use of ST in the dense layers was superior to ReLU in terms of test accuracy. Ultimately, as expected, SST improved the overall performance and learning efficiency of the neural networks.

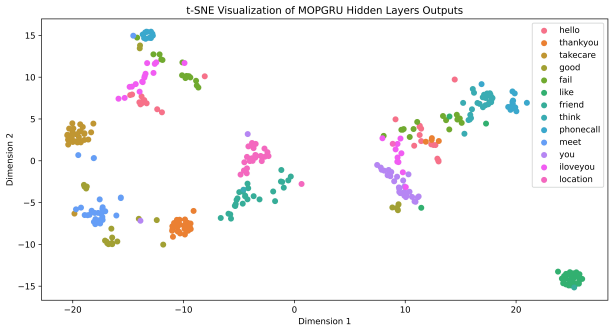
5. Experiments

We perform controlled comparisons between standard GRU and GRU-SST to isolate the effect of contrast-enhanced gating under low-data conditions. Models use identical architectures and training settings to ensure fair comparison. We comprehensively analyse the effectiveness of SST compared to traditional functions (sigmoid and tanh) on different tasks, such as sign language recognition, gait classification, and human activity recognition tasks, all of which involve temporal modelling of human motion signals commonly studied in computer vision and multimodal perception literature.

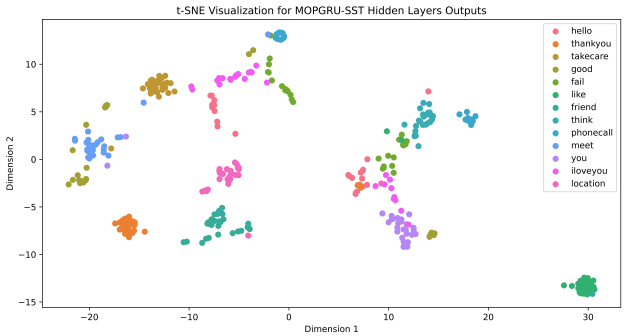
5.1. Sign Language Recognition

Sign language recognition is a critical task in human-computer interaction, enabling seamless communication with individuals who are deaf. For the experiments, the configuration is set according to [14]. The authors employ mediapipe optimized gated recurrent unit (MOPGRU) model to improve the performance with limited datasets. It consisted of real-time video recordings of 13 distinct Indian sign gestures as mentioned in [14]. Each sign gesture was represented by a collection of 30 videos, each comprising 30 frames, and all videos shared a uniform size of 640 × 480 pixels. To understand the data better, PCA visualization tool was adopted. From the plot, we observed that there exists sparse patterns in the ISL data, as shown in Figure 1. To address data sparsity and improve predictive performance, we incorporated the SST activation function into the MOPGRU model, resulting in the MOPGRU-SST configuration. The performance of MOPGRU-SST is compared against the standalone MOPGRU model, and the results are discussed in the following section.

T-SNE visualization of learned representations: The visualization of hidden layer outputs as shown in Figure 4a and 4b offers a compelling view into the semantic representation capabilities of the MOPGRU and MOPGRU-SST models. In the t-SNE plots, we observe that MOPGRU forms broader clusters for words such as "think", "phone call", and "meet", which could be indicative of its capacity

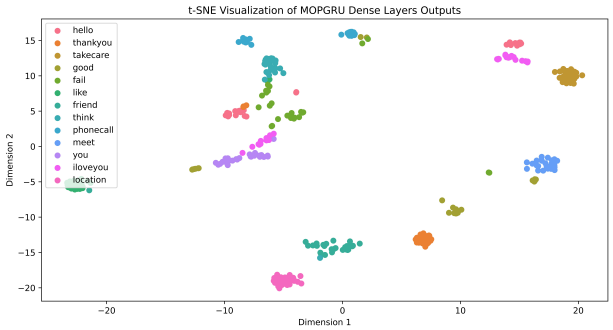


(a) MOPGRU

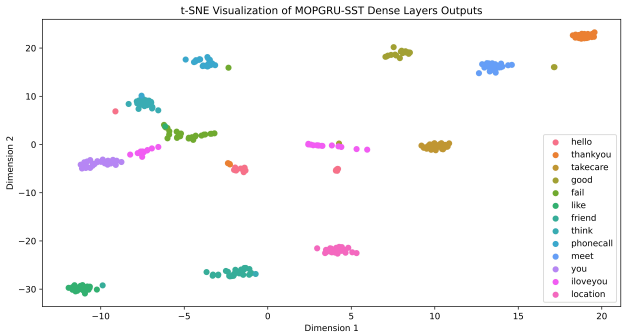


(b) MOPGRU-SST

Figure 4. T-SNE visualization of hidden layer outputs.



(a) MOPGRU



(b) MOPGRU-SST

Figure 5. T-SNE visualization of dense layer outputs.

to capture and represent semantic relationships. In contrast the separation is more prominent in MOPGRU-SST, signifying its ability to encode knowledge representations in a highly distinguished manner. This differentiation can enhance recognition capabilities by efficiently mapping varying inputs to more granular model concepts. Analysing the dense layer visuals reveals similar insights, with MOPGRU-SST in Figure 5b producing clearer delineation between clusters representing different terms. The evident distinction of semantic groupings compared to MOPGRU in Figure 5a indicates that MOPGRU-SST develops richer hierarchical data representations. Overall, by comparing the test accuracy of both models, MOPGRU-SST achieved a higher accuracy of 100% than baseline MOPGRU which achieved 95%. Although near-perfect accuracy is observed on this dataset, the number of gesture samples per class is limited and the task may be relatively structured in feature space. Therefore, this result should be interpreted as evidence of improved separability on a controlled low-data benchmark rather than as a definitive indication of general superiority. Evaluating SST on larger and more diverse gesture datasets remains an important direction for future work.

5.2. Gait classification

We utilize an accelerometer-based gait dataset from [24] comprising raw acceleration sequences collected from 244 subjects during a 3-minute walking task. The dataset provides a robust test application for evaluating recurrent models for classifying dynamic human activities that are sequential in nature. The original dataset, although suitable for analysing age-related differences in gait patterns as explored in [24], does not contain sufficient sparsity to assess model robustness under missing data conditions within data constraints. Since real-world sensor feeds often encounter missing information, we introduce sparsity to simulate this practical challenge. Specifically, 20% of values in the 1024-sample input sequences were randomly zeroed out to induce missing data points across temporal dimensions. In the induced sparse dataset, a sharp peak at zero indicates significant conversion of values to a single number, effectively creating sparse sequences. With 1460 sequences originally, this process generates 1168 non-sparse and 292 sparse patterns in the final training dataset. The incorporation of sparse data simulations allows comprehensive benchmarking of SST-powered GRU models against traditional GRU in handling sparse temporal inputs. Training the model, the experimental setup and model training parameters follow the same as mentioned in [24] with the additional incorporation of SST activation function in the classical GRU. For evaluating the classification performance, widely used evaluation metrics such as accuracy, precision, recall and F1 score were used. In addition, receiver operating characteristic (ROC) curves and the area under the curve (AUC) were generated.

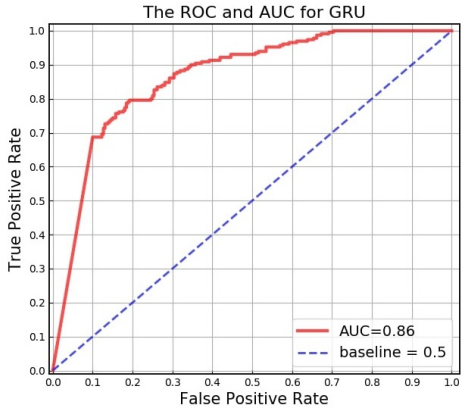
Classification performance: The integration of the proposed SST activation consistently improves GRU performance over the baseline across evaluation metrics as ob-

Table 1. Classification performance of GRU and GRU-SST.

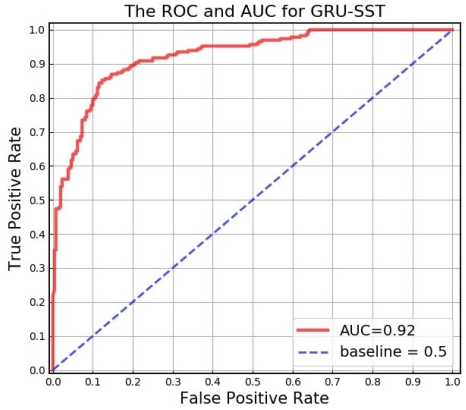
Model	Test Accuracy	Precision	Recall	F1-score	AUC
GRU	79.8%	78.9%	77.8%	78.3%	0.86
GRU-SST	84.3%	87.7%	77.4%	82.2%	0.92

served in Table 1. GRU-SST achieves a superior test accuracy of 84.3% compared to classical GRU's 79.8%. Precision increases from 78.9% to 87.7% with SST incorporation. Further, GRU-SST obtains an AUC of 0.92, substantially superior to the baseline GRU's 0.86 AUC, indicating higher discrimination between classes. With consistent gains in accuracy, precision, recall, and AUC, SST demonstrates concrete effectiveness in addressing GRU limitations when learning from sparse temporal data. The results validate SST's strengths in transforming classical recurrent models within constraints to boost predictive performance.

ROC and AUC curve analysis: The ROC curve shown in Figure 6 illustrates the performance of baseline GRU and GRU-SST models on a classification task. The red line represents the true positive rate (TPR) versus the false positive rate (FPR) at various threshold levels. An ideal model would have a curve that reaches towards the top left corner, indicating a high TPR and low FPR. The ROC analysis provides evidence on SST's efficacy with GRU-SST producing a higher AUC of 0.92 over the baseline's 0.86. This increased AUC value from 0.86 to 0.92 signifies a notable improvement in model performance, highlighting the potential of SST to improve the robustness and predictive accuracy of GRU networks in classification performance as illustrated by the ROC curve.



(a) GRU



(b) GRU-SST

Figure 6. ROC curves.

5.3. Human activity recognition

Human activity recognition (HAR) using wearable sensors provides an attractive test application for evaluating sequential modelling techniques. Sensor feeds capturing motions over time exhibit temporal dependencies well-suited to GRU networks. We leverage two standard HAR datasets to evaluate our proposed SST activation in enhancing GRU models. The first is the WISDM dataset [30] comprising 1,098,207 samples from 36 subjects performing daily activities like walking, jogging, sitting, and standing. The second is the UCI Smartphone dataset [31] containing 10,299 instances of sensor readings from 30 volunteers aged 19 to 48 years executing 6 standard activities - standing, sitting, lying, walking, upstairs and downstairs using a mobile phone. Each subject performed the protocol twice with varying device placements. 50Hz triaxial acceleration and angular velocity data were recorded and manually labelled. We induced 20% sparsity into both datasets to simulate real-world sensor feed challenges. For training, we followed [32] and compared classical GRU and GRU-SST performance for both datasets. The comparative evaluation results summarized in Table 2 clearly validate the effectiveness of the proposed SST activation in enhancing classical GRU models for human activity recognition, consistently across both datasets.

Table 2. Comparative performance of GRU and GRU-SST models on WISDM and UCI-HAR datasets.

Model	Dataset	Training Accuracy (%)	Testing Accuracy (%)
GRU	WISDM	99.5	97.08
GRU-SST	WISDM	100	99.28
GRU	UCI-HAR	99	93.08
GRU-SST	UCI-HAR	100	98.30

On sparse WISDM accelerometer data, SST allows GRU models to achieve close to 100% training accuracy. More importantly, test accuracy increases over 2% with GRU-SST, reaching 99.28% despite introduced sparsity. Similarly on Smartphone data, SST again shows generalization capability improving GRU test accuracy by 5% to 98.3%. The results validate SST's effectiveness in tackling GRU limitations on multivariate time-series across application domains. By reshaping classical activations, SST unlocks substantial accuracy and convergence gains. The 100% training accuracy observed in some HAR settings should be interpreted with caution, as these benchmarks are relatively well studied and may permit strong fitting under the current experimental setup. More informative than the training accuracy is the

Table 3. Loss and MSE comparison for GRU and GRU-SST on gold price forecasting.

Model	Training Loss	Validation Loss	Test Loss
GRU	0.70	0.08	0.28
GRU-SST	0.54	0.06	0.08

consistent improvement in test performance, which suggests that SST improves gating behaviour under the evaluated conditions. Nevertheless, broader validation across additional datasets and repeated trials would further strengthen conclusions about robustness and generalization.

5.4. Time-series forecasting

In financial forecasting, the ability to accurately predict future prices of commodities like gold is a testament to a model's predictive prowess. Gold, a commodity of high economic interest, presents a challenging yet valuable dataset for time-series analysis. The historical gold price dataset from Yahoo Finance, with its comprehensive range of data points from 2000 to 2026, was chosen for its complexity and relevance. The dataset's granular details, including open, high, low, close, and adjusted close prices, offer a multifaceted view of market behaviour, allowing the model to learn from sparse temporal patterns. The choice to use this dataset aligns with our objective to assess SST's ability to enhance GRU model performance. The daily fluctuations and trends encapsulated within gold price data require a model that discerns long-term dependencies. SST's design to amplify gradient flow and improve information retention is put to the test, making this dataset suitable for demonstrating SST's potential to refine predictions in a complex and unpredictable domain like financial time-series. For training, we utilized the model configuration from [33] based on GRU architecture by replacing the LSTM layers. To analyze SST improvements, mean squared error (MSE) loss was calculated between predicted and ground truth gold prices over the test period. As evidenced in Table 3, SST integration leads to substantial reductions in error metrics consistently across training and validation. GRU-SST lowers MSE compared to baseline GRU in terms of training and validation loss by over 20%. The test set MSE declines even further from 0.28 to 0.08. These consistent MSE reductions across phases validate SST's capability in accurately modelling temporal dependencies for financial forecasting with GRU. By tuning activations to sequential data, SST unlocks substantial predictive performance gains over classical approaches.

6. Conclusion

We presented squared sigmoid-tanh (SST), a zero-parameter modification to GRU gating that increases activation contrast by squaring gate outputs. SST is designed for low-data and sparse-signal settings where standard sigmoid/tanh gating may be overly smooth. Across diverse sequential tasks, SST-GRU consistently improved performance over a vanilla GRU without altering the GRU architecture. On sign language recognition, SST achieved 100% test accuracy compared to the baseline on this dataset. On gait classification, SST improved test accuracy from 79.8% to 84.3%. On human activity recognition, SST increased test accuracy to 99.28% (gain $>$2.2%) and reached 98.3% on the Smartphone dataset (gain 5%). For forecasting, SST reduced gold price prediction MSE from 0.28 to 0.08, accompanied by lower training and validation losses. These results show that a targeted nonlinearity change in GRU gating can improve performance in data-scarce settings with negligible computational cost.

Limitations. Our evaluation focuses on GRU-based pipelines and low-data benchmarks; future work should test SST under broader distribution shifts, longer sequence horizons, and stronger baselines such as lightweight Transformers or state-space models. Our study isolates the effect of contrast-enhanced gating within GRU architectures under limited-data conditions. Broader comparisons with alternative recurrent models (e.g., LSTM), activation variants (e.g., hard-sigmoid or parameterized nonlinearities), and modern sequence models such as Transformers or state-space approaches remain future work. Because SST sharpens gate responses, very small activations may be further suppressed, which could affect extremely long or weak-signal sequences.

References

[1] W. Duch and N. Jankowski, "Survey of Neural Transfer Functions," *Neural Computing Surveys*, vol. 2, pp. 163–212, 1999. https://fizyka.umk.pl/publications/kmk/99ncs.pdf

[2] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. https://doi.org/10.1038/nature14539

[3] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proc. 27th Int. Conf. Machine Learning (ICML)*, 2010, pp. 807–814. https://dl.acm.org/doi/10.5555/3104322.3104425

[4] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *Proc. 30th Int. Conf. Machine Learning (ICML)*, 2013.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2015, pp. 1026–1034. https://doi.org/10.1109/ICCV.2015.123

[6] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies," in *A Field Guide to Dynamical Recurrent Networks*, 2001, pp. 237–243. https://www.bioinf.jku.at/publications/older/ch7.pdf

[7] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," in *Proc. 30th Int. Conf. Machine Learning (ICML)*, 2013. https://proceedings.mlr.press/v28/pascanu13.pdf

[8] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[9] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," in *Proc. 8th Workshop Statistical Machine Translation*, 2014, pp. 103–111. https://doi.org/10.3115/v1/W14-4012

[10] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2014, pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179

[11] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, 1989. https://doi.org/10.1007/BF02551274

[12] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. https://doi.org/10.1109/72.279181

[13] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000. https://doi.org/10.1162/089976600300015015

[14] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K. H. Park, and J. Kim, "An integrated mediapipe-optimized GRU model for Indian sign language recognition," *Scientific Reports*, vol. 12, no. 1, art. 11964, Dec. 2022. https://doi.org/10.1038/s41598-022-15998-7

[15] Y. Vijay and H. Subramanyan, "FaCTR: Factorized Channel-Temporal Representation Transformers for Efficient Time Series Forecasting," arXiv:2506.05597, Jun. 2025. https://doi.org/10.48550/arXiv.2506.05597

[16] N. Q. Anh, P. Tran-Truong, S. Ha, T. Nhan, P. Nguyen, T. Tuan, and N. Kim,, "LiteFormer: A lightweight encoder-only Transformer for efficient financial time series forecasting across global stock indices," *Engineering Applications of Artificial Intelligence*, vol. 167, art. 113681, Mar. 2026. https://doi.org/10.1016/j.engappai.2025.113681

[17] S. Chakraborty, I. Delibasoglu, and F. Heintz, "Scaling transformers for time series forecasting: do pretrained large models outperform small-scale alternatives?," *Artificial Intelligence Review*, vol. 59, no. 2, art. 124, Feb. 2026. https://doi.org/10.1007/s10462-025-11481-7

[18] Fei Hao, Junhai Qiu, Xiaofeng Zhang, Yepeng Liu, Hua Wang, Yujuan Sun, Pengbin Zhang, "New perspectives on multivariate time series forecasting: Lightweight networks combined with multi-scale hybrid state space models," *Expert Systems with Applications*, vol. 272, art. 128845, Jan. 2026. https://doi.org/10.1016/j.eswa.2025.128845

[19] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proc. 14th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 315–323. https://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf

[20] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, "Simple Recurrent Units for Highly Parallelizable Recurrence," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2018, pp. 4470–4481. https://doi.org/10.18653/v1/D18-1477

[21] Z. Ye, "ss-Mamba: Semantic-Spline Selective State-Space Model," arXiv:2506.14802, Jun. 2025. https://doi.org/10.48550/arXiv.2506.14802

[22] D. Amodei *et al.*, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," in *Proc. 33rd Int. Conf. Machine Learning (ICML)*, 2016, pp. 173–182. https://proceedings.mlr.press/v48/amodei16.pdf

[23] B. Verma, "A two stream convolutional neural network with bi-directional GRU model to classify dynamic hand gesture," *Journal of Visual Communication and Image Representation*, vol. 87, art. 103554, Aug. 2022. https://doi.org/10.1016/j.jvcir.2022.103554

[24] X. Zheng, E. Otten, M. F. Reneman, and C. J. Lamoth, "Explaining deep learning models for age-related gait classification based on acceleration time series," *Computers in Biology and Medicine*, vol. 185, art. 109338, 2025. https://doi.org/10.1016/j.compbiomed.2024.109338

[25] S. Semeniuta, A. Severyn, and E. Barth, "Recurrent Dropout without Memory Loss," in *Proc. 26th Int. Conf. Computational Linguistics (COLING)*, 2016, pp. 1757–1766. https://aclanthology.org/C16-1165.pdf

[26] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2016. https://doi.org/10.48550/arXiv.1511.07289

[27] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," arXiv:1710.05941, Oct. 2017. https://doi.org/10.48550/arXiv.1710.05941

[28] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning Activation Functions to Improve Deep Neural Networks," arXiv:1412.6830, Apr. 2015. https://doi.org/10.48550/arXiv.1412.6830

[29] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-Recurrent Neural Networks," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017. https://doi.org/10.48550/arXiv.1611.01576

[30] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity Recognition using Cell Phone Accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011. https://doi.org/10.1145/1964897.1964918

[31] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition using Smartphone," in *Proc. 21st European Symp. Artificial Neural Networks (ESANN)*, 2013, p. 581. https://www.esann.org/sites/default/files/proceedings/legacy/es2013-84.pdf

[32] S. Mohsen, "Recognition of human activity using GRU deep learning algorithm," *Multimedia Tools and Applications*, vol. 82, no. 30, pp. 47733–47749, Dec. 2023. https://doi.org/10.1007/s11042-023-15571-y

[33] W. M. P. Dhuhita, M. F. Al Farid, A. Yaqin, H. Haryoko, and A. A. Huda, "Gold Price Prediction Based On Yahoo Finance Data Using Lstm Algorithm," in *Proc. Int. Conf. Informatics, Multimedia, Cyber and Information Systems (ICIMCIS)*, 2023, pp. 420–425. https://doi.org/10.1109/ICIMCIS60089.2023.10349035