PIECEWISE FEW-SPIKES NEURONS: NONLINEAR FUNCTION APPROXIMATION FOR PROVABLE AND EFFICIENT ANN-TO-SNN CONVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Due to their potential for energy-efficient deployment on neuromorphic hardware, Spiking Neural Networks (SNNs) have gained significant attention. While conversion from pre-trained Artificial Neural Networks (ANNs) serves as a popular route for constructing deep spiking models, its theoretical capabilities and underlying approximation dynamics remain insufficiently understood. In this paper, we categorize common conversion methods into two main paradigms and analyze their core mechanics, exposing their primary limitations concerning non-linear representations and activation resolution. To address these constraints, we propose Piecewise Few-Spike (PWFS) neurons, which introduce time-dependent parameter dynamics to enable accurate nonlinear function approximation without excessive spike counts. Unlike existing approaches that rely on heuristic optimizations, PWFS provides an explicit theoretical convergence guarantee with provable error bounds, establishing a precise mathematical trade-off between approximation accuracy and inference latency. We conduct comprehensive experiments to validate the efficacy of PWFS, showcasing its favorable accuracy in ANN-to-SNN conversion.

1 INTRODUCTION

Artificial Neural Networks (ANNs) have achieved remarkable success across complex visual, sequence, and representation learning tasks (Krizhevsky et al., 2012; Vaswani et al., 2017). However, their reliance on continuous-valued, high-precision operations creates significant computational and energy demands (Thompson et al., 2021). Often referred to as the third generation of neural networks (Maass, 1997), spiking neural networks (SNNs) offer a promising low-power alternative enabled by their event-driven computational dynamics inspired by biological neurons and compatibility with neuromorphic hardware (Mehonic et al., 2024). By communicating through sparse, discrete binary signals over time, SNNs may drastically reduce the energy required for inference (Eshraghian et al., 2023). Architecturally, SNNs process information across both *spatial dimensions* (layer-to-layer feature transformations) and the *temporal dimension* (discrete time-steps governing internal membrane dynamics), operating much like recurrent neural networks in the temporal domain (Elman, 1990).

To realize these efficiency benefits in practice, the implementation of deep SNNs generally follows two prevailing paradigms: *direct training* and *ANN-to-SNN conversion* (Eshraghian et al., 2023). Direct training of deep SNNs is challenging due to the non-differentiable nature of spiking activations. Although backpropagation through time with surrogate gradients (Lee et al., 2016; Neftci et al., 2019) enables gradient-based learning, it is prone to optimization instabilities and requires high memory costs that scale with the temporal dimension, limiting its applicability to architectures with small temporal dependencies (Eshraghian et al., 2023). Consequently, ANN-SNN conversion (Cao et al., 2015; Rueckauer & Liu, 2018; Stöckl & Maass, 2021; Jiang et al., 2024) has established itself as an attractive alternative: taking a pre-trained, high-performing continuous ANN and translating its learned representations into an SNN without costly retraining.

The fundamental challenge in ANN-to-SNN conversion stems from approximating continuous activations using SNN native output: a finite *sequence of binary spikes* (Rueckauer et al., 2017; Stöckl & Maass, 2021; Bu et al., 2022a). Traditional approaches rely on heuristic function approximations that lack formal error bounds and are highly sensitive to hyperparameter configurations. In deep

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

network architectures, having formal guarantees is particularly critical, as the error from approximating activation functions compounds across successive layers, incurring a well-documented drop in accuracy compared to the source ANN (Sengupta et al., 2019; Bu et al., 2022a). To mitigate this drop in accuracy, existing frameworks, such as those proposed by Jiang et al. (2024) or Stöckl & Maass (2021), typically increase spatial model capacity or expand the temporal dimension. However, these heuristic adjustments significantly inflate computational costs, eroding the primary energy advantages of event-driven spiking execution without ensuring significant improvements in conversion accuracy.

In this work, we propose a simple yet effective neuron model, termed Piecewise Few-Spike (PWFS) neurons, designed to approximate arbitrary nonlinear activations with provable error bounds. PWFS serves as a drop-in replacement for standard continuous activation functions, making it readily adaptable to existing ANN-to-SNN conversion pipelines. Unlike existing methods that rely on heuristic parameter tuning, we establish a theoretical convergence guarantee for PWFS neurons with an explicit error bound relative to the number of time steps. This enables a precise, mathematically grounded trade-off between the number of time steps on a PWFS neuron and its convergence accuracy.

Contributions. Specifically, our main contributions are as follows:

- We theoretically analyze the two prevailing paradigms in ANN-to-SNN conversion, clarifying their effective regimes while uncovering why they struggle with activation resolution and representing non-linearities.
- We propose a novel conversion framework using Piecewise Few-Spike (PWFS) neurons, which introduces time-dependent parameter dynamics to enable accurate nonlinear function approximation at a fixed inference latency.
- We provide formal convergence guarantees for the PWFS conversion framework and outline systematic optimizations for practical deployment. Finally, comprehensive experiments across diverse tasks and network architectures demonstrate that our method consistently achieves higher accuracy than competing methods, without compromising on efficiency.

2 Related works

Depending on how the continuous activations in the source ANN are encoded in the spike domain, ANN-to-SNN conversion methods can be broadly be categorized into two sub-classes: rate-coding and temporal-coding based conversion.

Rate-coding based conversion. Traditional approaches employ integrate-and-fire (IF or LIF) neurons, leveraging firing rates (i.e., the total number of spikes divided by the available time-steps) to approximate continuous activations (Cao et al., 2015; Diehl et al., 2015; Rueckauer et al., 2017; Sengupta et al., 2019; Han et al., 2021; Deng & Gu, 2021; Li et al., 2021). This paradigm suffers from two critical limits: (1) firing rates are constrained to a discrete grid, imposing a hard bound on convergence, and (2) as demonstrated in Section 3.1, the linear dynamics of IF/LIF neurons intrinsically align with piecewise-linear functions (like ReLU) but severely struggle with complex nonlinearities. To mitigate this, subsequent works have introduced surrogate gradient training (Bu et al., 2022a; You et al., 2024; Liu et al., 2025), multi-threshold models (Huang et al., 2024; Hao et al., 2024; Huang et al., 2025), population coding (Jiang et al., 2024), or mixed-precision networks that retain continuous nonlinearities (You et al., 2024; Huang et al., 2024; Cao et al., 2026; Zheng et al., 2026). However, deploying multiple spiking neurons per activation, utilizing multi-threshold grids, or retaining floating-point operations significantly degrades energy efficiency and compatibility with standard neuromorphic hardware. In contrast, our PWFS model approximates complex nonlinearities within a single, hardware-friendly neuron dynamic.

Temporal-coding based conversion. Time-to-first-spike (TTFS) coding offers exceptional energy efficiency by restricting activity to at most one spike per neuron (Rueckauer & Liu, 2018; Zhang et al., 2019; Park et al., 2020; Stanojevic et al., 2023; 2024; Zhao et al., 2025), but exhibit a fundamental convergence limit. Relaxing this strict constraint, the few-spike (FS) neuron model (Stöckl & Maass, 2021) introduces a parametric dynamic with weighted spikes. This enables them to express $\mathcal{O}(2^T)$ activation values, vastly outperforming the $\mathcal{O}(T)$ values of rate-coding while maintaining hardware compatibility. Yet, despite their theoretical capacity to approximate ReLUs, FS

neurons struggle to fit complex activations like GELU or SiLU, exhibiting high sensitivity to training procedures (Wang et al., 2026). While recent studies attempt to refine FS optimization (Jeong et al., 2024; Chen et al., 2026; Zheng et al., 2026; Wang et al., 2026), they lack formal approximation guarantees. Alternatively, methods proposing stable hyperparameter optimization (Wang et al., 2025) require input-dependent parameter selection during inference. However, this dependency necessitates online adaptive computation or custom neuromorphic hardware. Unlike these heuristic or hardware-restrictive modifications, our PWFS framework introduces time-dependent parameter dynamics that provably approximate arbitrary nonlinearities, providing rigorous convergence bounds without requiring specialized hardware modifications.

3 THEORETICAL ANALYSIS OF EXISTING FRAMEWORKS

We organize the existing ANN-to-SNN literature broadly into two major operational frameworks. The first paradigm builds upon classical Integrate-and-Fire (IF) neuron models (Gerstner et al., 2014; Eshraghian et al., 2023), leveraging rate-based approximations rooted in biological neuronal dynamics. The second paradigm utilizes Few-Spike (FS) neuron formulations (Stöckl & Maass, 2021) and their variants to improve the conversion. Below, we synthesize the core principles of both paradigms and conduct a rigorous mathematical analysis of their underlying conversion error bounds, setting aside improvement- and implementation-specific details.

3.1 INTEGRATE-AND-FIRE (IF) NEURONS AND RATE-BASED CONVERSION

Traditional ANN-to-SNN conversion frameworks (Cao et al., 2015; Rueckauer et al., 2017) primarily employ the simple IF neuron model while encoding information in the total spike count over T time steps.

Definition 3.1 (IF neurons). *Let $u(0) \in \mathbb{R}$ (initial membrane potential), $\theta \in \mathbb{R}_+$ (threshold). Then an **integrate-and-fire (IF) neuron** is a* sequence-to-sequence *function*

$$G := G_{u(0),\theta} : \mathbb{R}^{\mathbb{N}} \mapsto \{0,1\}^{\mathbb{N}}, \quad \big(x(t)\big)_{t=1}^{\infty} \mapsto \big(s(t)\big)_{t=1}^{\infty},$$

defined by the following dynamics

$$\begin{cases} u_{pre}(t) & = u(t-1) + x(t), \\ s(t) & = H(u_{pre}(t) - \theta), \\ u(t) & = u_{pre}(t) - \theta s(t), \end{cases}$$

where $H(\cdot)$ is the Heaviside step function, defined as $H(u) = 1$ if $u \geq 0$, and 0 otherwise.

At its core, the rate-coding pipelines operate by replacing continuous activation units in the source ANN with Integrate-and-Fire (IF) neurons. To understand the fundamental limitations of this mechanism, the following theorem provides a precise error bound for IF approximation, mathematically revealing why this dynamic is intrinsically restricted to piecewise-linear functions like ReLU.

Proposition 3.2 (Convergence of IF-neurons with rate-coded inputs). *Let $x \in \mathbb{R}$, $\big(x(t)\big)_{t=1}^{\infty} \subset \mathbb{R}$ be a sequence and $C > 0$ be a constant such that*

$$\left| \frac{1}{t} \sum_{i=1}^{t} x(i) - x \right| \leq \frac{C}{t} \qquad \forall t \in \mathbb{N}.$$

Let $\theta > 0$ and let $G = G_{u(0),\theta} : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$ be an IF neuron that satisfies

$$\frac{u(0) - C}{\theta} - 1 \leq 0 \leq \frac{u(0) + C}{\theta}.$$

Let $\big(s(t)\big)_{t\in\mathbb{N}} := G\Big(\big(x(t)\big)_{t\in\mathbb{N}}\Big)$. Then for some constant $C' > 0$ depending only on C, $u(0)$ and θ, it holds

$$\left| \frac{1}{t} \sum_{i=1}^{t} s(i) - \mathrm{ReLU1}\left(\frac{x}{\theta}\right) \right| \leq \frac{C'}{t} \qquad \forall t \in \mathbb{N},$$

where $\mathrm{ReLU1}$ *is defined as* $\mathrm{ReLU1}\,(z) := \min\{\mathrm{ReLU}(z), 1\}$, *for $z \in \mathbb{R}$.*

Proposition 3.2 (proven in Appendix B) theoretically justifies rate-based conversion: via standard *threshold balancing* or *weight normalization* (Diehl et al., 2015; Sengupta et al., 2019; Rueckauer et al., 2017), the IF firing rate faithfully recovers the continuous ReLU activation up to scale. This local guarantee straightforwardly extends to deep architectures, achieving an optimal asymptotic rate of $\mathcal{O}(1/T)$. While heuristic optimizations, such as optimizing initial potentials (Bu et al., 2022b) or tracking residual membrane voltages (Hao et al., 2023), help mitigate the error constants that grow with network depth, they cannot overcome this fundamental $\mathcal{O}(1/T)$ rate barrier.

More importantly, Proposition 3.2 exposes the core representational limit of standard IF neurons: their time-averaged behavior converges to a piecewise-linear $\mathrm{ReLU1}$ mapping. This formalizes their rigid attachment to ReLU networks and demonstrates why extending IF dynamics to general non-linearities is problematic. Consequently, recent workaround approaches to handle complex activations, such as deploying population-coded clusters (Jiang et al., 2024) or retaining floating-point operations from the source model (You et al., 2024; Cao et al., 2026), ultimately sacrifice the computational efficiency required for low-power neuromorphic deployment.

3.2 Few-Spikes (FS) neurons

To overcome the key limitations of rate-based conversion using standard IF neurons (and also of TTFS-based conversion), namely their $\mathcal{O}(1/T)$ resolution bottleneck and restriction to ReLU, Few-Spike (FS) neurons (Stöckl & Maass, 2021) replace static parameters with time-dependent dynamics, enabling a much higher representational capacity.

Definition 3.3 (FS neuron)**.** *Given the number of time steps $T \in \mathbb{N}$, a **few-spikes (FS) neuron** is a real-valued function $g = g_{\theta,h,d} : \mathbb{R} \to \mathbb{R}$ with* thresholds $\theta(t) \in \mathbb{R}$, potential reset values $h(t) \in \mathbb{R}$, $t \in [T]$, *and* spike weights $d(t) \in \mathbb{R}$, *for $t \in \{0\} \cup [T]$, as (hyper-)parameters defined through the following dynamics*

$$\begin{cases} s(t) & = H\Big(v(t) - \theta(t)\Big) \\ v(t+1) & = v(t) - h(t)s(t) \end{cases}, \quad t \in [T], \tag{1}$$

where the input is given as $v(1) = x \in \mathbb{R}$ and the output is

$$g(x) := d(0) + \sum_{t=1}^{T} d(t)s(t).$$

In contrast to rate-coded neurons whose expressivity scales linearly as $\mathcal{O}(T)$, FS neurons achieve an exponential number of distinct function values, scaling up to $\mathcal{O}(2^T)$. Crucially, despite this enhanced representational capacity, FS neurons strictly maintain local, binary spike accumulation, ensuring seamless execution on neuromorphic hardware platforms (Stöckl & Maass, 2021). The following result shows that FS neurons with appropriate choice of parameters in fact approximate the ReLU functions with an error of $\mathcal{O}(2^{-T})$.

Proposition 3.4 (FS for linear activations)**.** *Let $T \in \mathbb{N}$, $B > 0$, $g_T : \mathbb{R} \to \mathbb{R}$ be an FS neuron with parameters $d(0) = 0$, $\theta(t) = d(t) = h(t) = 2^{-t}B$, $t \in [T]$ and $x \in (-\infty, B)$ be an arbitrary input. Then the* spike train $\big(s(t)\big)_{t\in[T]}$ *corresponds to the coefficients in the binary representation of $\frac{x}{B}$ in $[0, 1)$ with T bits. In particular, it holds*

$$|g_T(x) - \mathrm{ReLU}\,(x)| \le 2^{-T}B \quad \text{for all } x \in (-\infty, B],$$

and $g_T(x) = \sum_{t=1}^{T} 2^{-t}B$ for $x > B$.

Proposition 3.4 (proven in Appendix C.1) confirms that FS neurons approximate ReLU functions with an $\mathcal{O}(2^{-T})$ error rate, achieving the theoretical optimal limit for an output constructed from T binary events. However, despite this exponential approximation capacity, FS neurons remain fundamentally inadequate for general nonlinearities. While prior literature largely attributes this to empirical training instabilities (Wang et al., 2026; Jeong et al., 2024), we trace the failure to a structural mismatch rooted in their parametric rigidity: the magnitude of change in the FS output upon toggling a spike is strictly fixed by its parameters (i.e., $d(t)/h(t)$). Consequently, it cannot track a target function f whose local slope continuously varies, causing the approximation error to saturate even as $T \to \infty$ (for a more detailed discussion see Appendix C.2).

This critical limitation directly motivates our Piecewise Few-Spike (PWFS) model introduced next. By partitioning the input domain into subintervals where f is locally affine (i.e., of constant slope), PWFS dynamically rescales the FS parameters to accurately match each piecewise component.

4 The Piecewise Few-Spike (PWFS) Neuron Model

In this section, we formulate the Piecewise Few-Spike (PWFS) neuron to approximate a continuous target activation function $f : [a, b] \to \mathbb{R}$. The core intuition is to first approximate f with a continuous piecewise-linear function f_M over M subintervals defined by knot points $a = x_0 < x_1 < \cdots < x_M < x_{M+1} = b$, and subsequently simulate each linear piece using an conventional FS dynamic. Governed by two primary hyperparameters—the number of knot points M and the number of FS steps K—the PWFS framework operates via an offline preparation step followed by a two-phase inference dynamic requiring $M + K$ total time steps.

Offline preparation. The offline phase operates independently of runtime inference to numerically compute an optimal piecewise-linear approximator f_M for a given continuous target function f and knot count M. Because knot placement for piecewise-linear and general spline approximators is a classical and extensively studied problem in numerical analysis (de Boor, 1973; 1978; Burden & Faires, 2010; Camponogara & Nazari, 2015), this optimization can be solved efficiently using off-the-shelf algorithms. Additional implementation details are provided in Appendix A.2.

Inference Phase 1: Interval search (M time steps). Assuming the knot sequence $(x_m)_{m\in[M]}$ has been optimized offline, the first inference phase determines which subinterval contains the input x. For an input $x \in [a, b]$ residing in the m-th subinterval $[x_m, x_{m+1}]$, the goal is to emit a single-spike one-hot vector $s = e_m^\top = (0, \ldots, 0, 1, 0, \ldots, 0) \in \{0, 1\}^M$, while retaining a residual membrane voltage of $v(M + 1) = x - x_m$. To achieve this, we employ the standard FS dynamic equation 1, starting with the initial membrane potential $v(1) := x$. The voltage reset values are fixed to the knot points in reverse order, $h(m) := x_{M-m+1}$ for $m \in [M]$, while the thresholds update as

$$\theta(m) = x_{M-m+1} + \infty \cdot \sum_{i=1}^{m-1} s(i), \quad m \in [M].$$

With the convention $\infty \cdot 0 = 0$, the threshold is pushed to infinity immediately after the first spike, strictly enforcing a maximum of one spike over the M time steps. Overall, this dynamic executes a sequential search across $x_M, x_{M-1}, \ldots$ until it finds an m where $x \geq x_m$. By construction, the voltage resets to $v(m) = x - x_m$ and remains unchanged until the end of the phase.

Inference Phase 2: Dyadic FS representation (K time steps). The second phase adjusts the neuron parameters based on the output of Phase 1 to compute the final approximation of $f(x)$. Taking the residual voltage $v(M + 1) = x - x_m$ from Phase 1 as its input, this phase continues the FS dynamic for $t \in [K]$ as

$$\begin{cases} \tilde{s}(t) & = H\Big(\tilde{v}(t) - \tilde{\theta}(t)\Big), \\ \tilde{v}(t+1) & = \tilde{v}(t) - \tilde{h}(t)\tilde{s}(t), \end{cases} \tag{2}$$

where $\tilde{a}(t)$ denotes $a(M + t)$ for variables $a \in \{s, v, h, \theta\}$. To scale the parameters correctly, we use the one-hot spike train s from Phase 1 to select the corresponding interval width E (base value for the thresholds and reset voltages), adaptive slope A, and bias B

$$E := \sum_{m=1}^{M} (x_{m+1} - x_m)s(m), \quad A := \sum_{m=1}^{M} (y_{m+1} - y_m)s(m), \quad B := \sum_{m=1}^{M} y_m s(m).$$

Because s contains only one spike at index m, these sums reduce perfectly to the specific parameters of the active subinterval (e.g., $E = x_{m+1} - x_m$). The thresholds and voltage subtract values for Phase 2 are then defined as $\tilde{h}(t) := \tilde{\theta}(t) := E/2^t$. Finally, we define the base bias $d(0) := B$ and the spike weights $\tilde{d}(t) := A/2^t$, yielding the final output of the PWFS neuron

$$y := d(0) + \sum_{t=1}^{K} \tilde{d}(t)\tilde{s}(t). \tag{3}$$

A theoretical error bound for this approximation is derived in Theorem 4.2 below, and a summarized pseudo code of the complete PWFS method is provided in Algorithm 1 (Appendix A.1).

Definition 4.1 (PWFS neuron)**.** *Given hyperparameters $M, K \in \mathbb{N}$, a **piecewise few-spikes (PWFS) neuron** is a real-valued function $g = g_{\theta,h,d,x_{\text{knot}},y_{\text{knot}}} : [a,b] \to \mathbb{R}$, where $x_{\text{knot}} = (x_m)_{m\in[M]}$ is a given knot sequence, $y_{\text{knot}} = (y_m)_{m\in[M]}$ a sequence of knot values, and $\tilde{\theta} = (\tilde{\theta}(t))_{t\in[K]}$, $\tilde{d} = (\tilde{d}(t))_{t\in[K]}$, $\tilde{h} = (\tilde{h}(t))_{t\in[K]}$, are as in equation 2. For a given input $x \in [a,b]$, applying Phase 1 and Phase 2 described above we obtain $g(x) = y$, given in equation 3, as the neuron output.*

Notice that while the offline preparation step above is an important step in the PWFS framework when used to approximate a given target, Definition 4.1 is independent of this step, since the knot points $(x_m)_{m\in[M]}$ and values $(y_m)_{m\in[M]}$ are taken as parameters of the model.

The next theorem provides a rigorous error bound for the convergence of the PWFS framework.

Theorem 4.2 (PWFS approximation guarantee)**.** *Let $f : [a,b] \to \mathbb{R}$ be a continuously twice differentiable target function, $M, K \in \mathbb{N}$. Let $a = x_0 < x_1 < \ldots < x_{M+1} = b$ be the knot points chosen on $[a,b]$ such that the corresponding piecewise linear interpolator f_M satisfies*

$$\|f - f_M\|_\infty \leq CM^{-2} \tag{4}$$

for a positive constant $C > 0$ (depending on $b-a$ and f''). There exists $g : [a,b] \to \mathbb{R}$ a PWFS neuron as in Definition 4.1, for which it holds

$$\|g - f\|_\infty \leq C'(M^{-2} + 2^{-K}),$$

where $C' > 0$ is a positive constant depending on $b-a$, $\|f''\|_\infty$ and $\|f'\|_\infty$. Additional, g can be computed by an algorithm with $M + K$ steps.

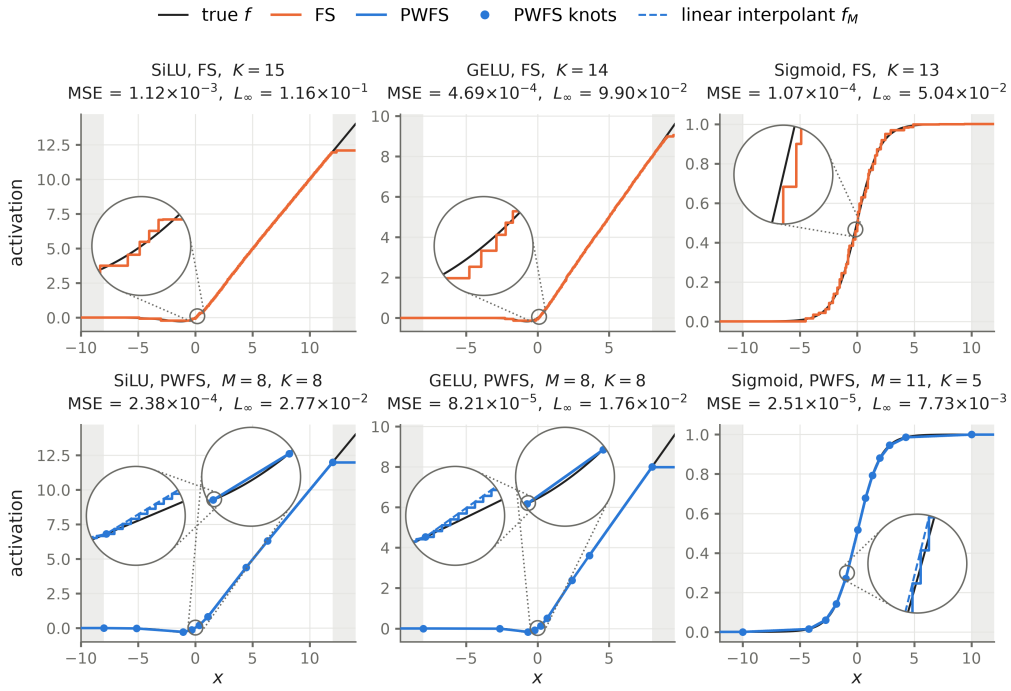

Figure 1: An illustration on how continuous activation functions are approximated by FS and PWFS neurons on the interval $[-10, 10]$. Here Sigmoid ($\sigma(x) = 1/(1+e^{-x})$), SiLU ($x \cdot \sigma(x)$), and GELU ($x\Phi(x)$, where Φ is the standard Gaussian CDF).

A few remarks are in order. First, the efficient algorithm to compute g mentioned in Theorem 4.2 corresponds to Algorithm 1 in Appendix A.1. Second, condition equation 4 is easily fulfilled: the equidistant grid already satisfies it with $C = (b-a)^2 \|f''\|_\infty / 8$, and optimising the knots improves

this constant—though not the rate—by standard techniques (see Appendix A.2). Thus, Theorem 4.2 provides an explicit trade-off between conversion accuracy and inference latency

$$\|g - f\|_\infty \lesssim \min_{\substack{M,K\in\mathbb{N} \\ M+K=T}} M^{-2} + 2^{-K}\,, \tag{5}$$

where T is the total number of time steps. Here, M denotes the number of linear pieces used to approximate f, capturing its deviation from linearity, while K quantifies the local approximation error. Note that this is a worst-case bound for general functions f, as the M^{-2} term vanishes when f is already piecewise linear.

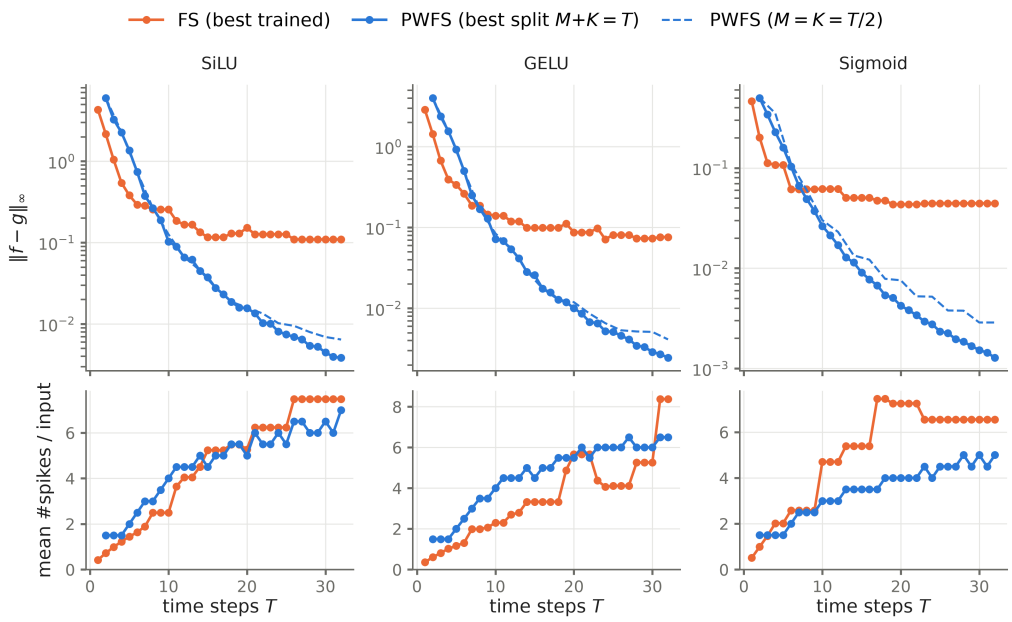

Figure 2: A comparison, in terms of the $\|\cdot\|_\infty$ norm and mean spike count, of FS and PWFS for approximating the Sigmoid, SiLU and GELU nonlinear activation functions.

Figure 1 illustrates representative approximation profiles for FS and PWFS under fixed temporal hyperparameters, while Figure 2 provides a quantitative comparison between the two frameworks across varying temporal dimensions. To account for the heuristic parameter optimization in FS, we execute multiple random initializations and report results from the optimal run.

As shown in the top row of Figure 2, the approximation error of the standard FS framework plateaus beyond a sufficiently large T. This aligns with our theoretical finding in Appendix C.2 regarding their approximation capacity, namely that scaling T in standard FS neurons may fail to yield accuracy gains even after allowing the spike count to grow exponentially (up to 2^T). In constrast, the error under PWFS exhibits a consistent decay as T increases, empirically validating the convergence guarantee in Theorem 4.2, in Appendix C.2.

The second row of Figure 2 plots the average spike count per neuron across all inputs within the fixed interval $[-10, 10]$. Due to the inherent instability of heuristic parameter training in the conventional FS framework, its spike activity exhibits irregular behavior, making difficult to control the spike rate. In contrast, for PWFS, K is selected based on the optimal partitioning of $M + K = T$. Because the spike count per neuron is strictly bounded by $2^K + 1$, PWFS offers significantly better control over spiking activity. Empirically, the plot confirms that PWFS yields a more stable and predictable trend, with spike activity increasing only modestly as T grows.

5 EXPERIMENTS

To evaluate our approach, we apply PWFS conversion to a range of architectures and benchmark tasks and compare it against standard FS conversion. Both neuron types are fitted to the activation function on the input range measured for each model and activation type. For PWFS, the knots of a configuration (M, K) are placed by a minimax dynamic program over a grid of $2{,}001$ candidate points, which minimizes the error $\|f - g\|_\infty$ of the resulting neuron, including the truncation error of its K Phase-2 steps; the placement is deterministic and exact on the candidate grid. For FS, the parameters are trained by gradient descent following (Stöckl & Maass, 2021), from three random initializations per configuration, of which we keep the one with the lowest Mean Squared Error (MSE). We perform a grid search over $K \in \{10, 12, \ldots, 18\}$ for FS (latency $T = K$) and over all 30 pairs $M \in \{4, 6, \ldots, 14\}$, $K \in \{10, 12, \ldots, 18\}$ for PWFS (latency $T = M + K$). Given a latency budget T, every activation is replaced by the fitted neuron with the lowest error among those with at most T time steps, and we report its latency T alongside the results.

We notice that training directly over the full activation intervals caused severe convergence issues for FS conversion because large bounding intervals force smooth non-linearities (GELU, SiLU) into crude, ReLU-like approximations. To stabilize training, capping the input range was preferable to sacrificing the entire activation function dynamics, see Table 4 in Appendix D. However, in EfficientNet (see Table 1), large input values are heavily present, causing standard conversion to fail (as reflected in its low baseline accuracy). Our PWFS cleanly resolves this bottleneck by dynamically handling these high-magnitude activations without clipping or sacrificing non-linear fidelity.

For image classification on ImageNet-1k (Russakovsky et al., 2015), we evaluate our approach on a diverse set of vision backbones: EfficientNet-B0 (Tan & Le, 2019) to assess standard SiLU non-linearities, ConvNeXt-Tiny (Liu et al., 2022) as a modern convolutional architecture, and DeiT-Small (Touvron et al., 2021) as a vision transformer architecture. Table 1 demonstrates that PWFS conversion suffers virtually no accuracy degradation compared to the pre-trained source ANN, systematically bypassing the performance of optimal FS configurations.

Model	Conversion	K	M	T	ImageNet-1k
EfficientNet-B0	-	-	-	-	77.72±0.19
	FS	16/12	-	16/12	15.69±0.16
	PWFS	18/12	14	32/26	77.43±0.19
ConvNeXt-T	-	-	-	-	82.06±0.17
	FS	12	-	12	81.73±0.17
	PWFS	18	14	32	81.97±0.17
DeiT-S	-	-	-	-	79.83±0.18
	FS	12	-	12	77.60±0.19
	PWFS	12	14	26	79.70±0.18

Table 1: Top-1 accuracy (%, mean ± standard error) on the ImageNet-1k.

Next, we extend our evaluation to natural language understanding (NLU) tasks to determine whether the benefits of PWFS conversion generalize to language models. Specifically, we apply our conversion approach to two model families at different parameter scales: Pythia (2.8B and 6.9B) (Biderman et al., 2023) and Qwen3 (1.7B and 8B) (Yang et al., 2025) and evaluate zero-shot classification performance across standard NLU benchmarks: PIQA (Bisk et al., 2020), ARC-Easy (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), Winogrande (Sakaguchi et al., 2021), COPA (Roemmele et al., 2011), WSC273 (Levesque et al., 2012), and RTE (Wang et al., 2019).

Table 2 presents the zero-shot accuracy metrics on the primary task suite, with remaining datasets and additional baseline comparisons detailed in Appendix D. Across varying architectural scales, PWFS consistently maintains high fidelity to the unconverted baseline, yielding negligible drop-offs that remain well within standard error margins.

Finally, we evaluate the generative capability of PWFS and standard FS conversion applied to a pretrained GPT-2 model (Radford et al., 2019). Table 3 reports performance on the WikiText-103 (perplexity) (Merity et al., 2017) and enwik8 (bits-per-character) (Hutter, 2006) benchmarks.

Model	Conversion	PIQA	ARC-e	OBQA	Wino
pythia-2.8b	-	73.99±1.02	64.27±0.98	23.80±1.91	59.59±1.38
	PWFS	73.72±1.03	63.97±0.99	24.20±1.92	59.27±1.38
pythia-6.9b	-	75.24±1.01	67.13±0.96	25.80±1.96	60.62±1.37
	PWFS	74.97±1.01	67.34±0.96	26.00±1.96	61.25±1.37
Qwen3-1.7B	-	75.84±1.00	73.36±0.91	30.00±2.05	63.93±1.35
	PWFS	75.84±1.00	73.61±0.90	30.20±2.06	64.25±1.35
Qwen3-8B	-	79.33±0.94	81.61±0.79	32.60±2.10	71.82±1.26
	PWFS	79.11±0.95	82.74±0.78	31.20±2.07	72.69±1.25

Table 2: Zero-shot accuracy (%, mean ± standard error) on PIQA, ARC-e, OBQA, and Winogrande.

Conventional FS conversion introduces notable degradation across both datasets, with WikiText-103 perplexity increasing from the baseline model's 37.38 to 42.62 ($+5.24$ points) and enwik8 rising from 1.23 to 1.29 bits-per-character. In contrast, PWFS conversion largely eliminates this gap, recovering baseline performance almost completely with a perplexity of 37.47 on WikiText-103 and matching the baseline enwik8 score of 1.23.

Model	Conversion	K	M	T	wikitext103	enwik8
GPT-2	-	-	-	-	37.38	1.23
	FS	12	-	12	42.62	1.29
	PWFS	18	14	32	37.47	1.23

Table 3: Perplexity (WikiText-103) and bits-per-character (enwik8) comparison for GPT-2 under FS and PWFS conversions. Lower values indicate better performance.

In general, our empirical results demonstrate that PWFS effectively preserves continuous representation capacity, enabling converted spike-based models to closely match original model performance across diverse tasks.

6 Conclusion

In this work, we presented a theoretical analysis of existing rate- and temporal-coded ANN-to-SNN conversion methods, focusing on the core principles of standard Integrate-and-Fire (IF) and Few-Spikes (FS) neuron models. We demonstrated that IF neurons suffer not only from coarse activation resolution but also from an intrinsic restriction to ReLU activations. While FS neurons improve upon IF resolution, their expressive capacity to approximate arbitrary non-linear functions remains fundamentally constrained. To overcome these limitations, we introduced Piecewise Few-Spikes (PWFS), a novel conversion framework that partitions the input domain into subintervals—where the target function is closely approximated by linear segments, augmented by input-dependent, dynamic adjustments of temporal parameters during inference. We derived theoretical error bounds for PWFS and empirically validated our framework through extensive experiments across diverse tasks and architectures.

Limitations and Future Work. Our analysis focuses on neuron-level maximum error bounds, which facilitates deriving rigorous theoretical guarantees. Consequently, our current framework does not explicitly account for empirical data distributions. Future research could extend the adaptive temporal parameter mechanism to incorporate input statistics, further enhancing conversion accuracy while reducing energy consumption.

AI USE STATEMENT

In this work, we used generative AI tools for text rephrasing and language polishing, searching and cross-checking related literature, and code editing and debugging. We have not used generative AI tools for primary scientific hypothesis generation, mathematical derivations, or theoretical proofs, and data collection tasks are not applicable to this work. Additionally, we used generative AI tools to assist in improving prose clarity across technical sections and converting equations into clean LaTeX markup. We have thoroughly reviewed all AI-assisted work; specifically, all AI-suggested related works and citations were manually verified against original publications to ensure accuracy and prevent hallucinated references, and all edited code components were unit-tested and validated for computational correctness by the authors. We take full responsibility for the final content of this work, including all text, code artifacts, and theoretical claims produced with the aid of generative AI.

REFERENCES

Richard Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284, 1961.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, pp. 2397–2430. PMLR, 2023.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Tong Bu, Wei Fang, Jianhao Ding, Penglin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2022a.

Tong Bu Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Optimized potential initialization for fast and accurate ann-to-snn conversion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11–20, 2022b. doi: 10.1609/aaai.v36i1.19874.

R.L. Burden and J.D. Faires. *Numerical Analysis*. Cengage Learning, 2010. ISBN 9780538733519.

Eduardo Camponogara and Luiz Fernando Nazari. Models and algorithms for optimal piecewise-linear function approximation. *Mathematical Problems in Engineering*, (1):876862, 2015.

Honglin Cao, Shuai Wang, Zijian Zhou, Ammar Belatreche, Wenjie Wei, Yu Liang, Yu Yang, Rui Xi, Malu Zhang, and Haizhou Li. Towards training-free and accurate ANN-to-SNN conversion via activation-aware redistribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1703–1711, 2026. doi: 10.1609/aaai.v40i3.37148.

Yongqiang Cao, Yang Chen, and Deepak Khosla. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66, May 2015. ISSN 1573-1405. doi: 10.1007/s11263-014-0788-3.

Long Chen, Xiaotian Song, and Yanan Sun. Las: loss-less ann-snn conversion for fully spike-driven large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

C. de Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer New York, 1978. ISBN 9783540903567.

Carl de Boor. Good approximation by splines with variable knots. In A. Meir and A. Sharma (eds.), *Spline Functions and Approximation Theory*, pp. 57–72. Birkhäuser Basel, 1973. ISBN 978-3-0348-5979-0. doi: 10.1007/978-3-0348-5979-0_3.

Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations*, 2021.

R.A. DeVore and G.G. Lorentz. *Constructive Approximation*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1993.

Peter Diehl, Dan Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *International Joint Conference on Neural Networks, IJCNN*, 2015.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 0364-0213. doi: https://doi.org/10.1016/0364-0213(90)90002-E.

Jason K. Eshraghian, Max Ward, Emre O. Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, USA, 2014.

Bingjiang Han, Jianhao Ding, and Guoqi Li. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, pp. 2329–2335, 2021.

Zecheng Hao, Tong Bu, Jianhao Ding, Tiejun Huang, and Zhaofei Yu. Reducing ann-snn conversion error through residual membrane potential. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*. AAAI Press, 2023. ISBN 978-1-57735-880-0.

Zecheng Hao, Xinyu Shi, Yujia Liu, Zhaofei Yu, and Tiejun Huang. LM-HT SNN: Enhancing the performance of SNN to ANN counterpart through learnable multi-hierarchical threshold model. In *Neural Information Processing Systems*, 2024.

Zihan Huang, Xinyu Shi, Zecheng Hao, Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Towards high-performance spiking transformers from ANN to SNN conversion. In *ACM Multimedia 2024*, 2024.

Zihan Huang, Wei Fang, Tong Bu, Peng Xue, Zecheng Hao, Wenxuan Liu, Yuanhong Tang, Zhaofei Yu, and Tiejun Huang. Differential coding for training-free ANN-to-SNN conversion. In *Forty-second International Conference on Machine Learning*, 2025.

Marcus Hutter. The human knowledge compression contest. `http://prize.hutter1.net`, 2006.

Dayena Jeong, Jaewoo Park, Jeonghee Jo, Jongkil Park, Jaewook Kim, Hyun Jae Jang, Suyoun Lee, and Seongsik Park. A more accurate approximation of activation function with few spikes neurons. *arXiv/abs/2409.00044*, 2024.

Yizhou Jiang, Kunlin Hu, Tianren Zhang, Haichuan Gao, Yuqian Liu, Ying Fang, and Feng Chen. Spatio-temporal approximation: A training-free SNN conversion for transformers. In *The Twelfth International Conference on Learning Representations*, 2024.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25, 2012.

Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10, 2016.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6316–6325. PMLR, 18–24 Jul 2021.

Chang Liu, Jiangrong Shen, Xuming Ran, Mingkun Xu, Qi Xu, Yi Xu, and Gang Pan. Efficient ANN-SNN conversion with error compensation learning. In *Forty-second International Conference on Machine Learning*, 2025.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, 2022.

Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(97)00011-7.

Adnan Mehonic, Daniele Ielmini, Kaushik Roy, Onur Mutlu, Shahar Kvatinsky, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, Sabina Spiga, Sergey Savel'ev, Alexander G. Balanov, Nitin Chawla, Giuseppe Desoli, Gerardo Malavena, Christian Monzio Compagnoni, Zhongrui Wang, J. Joshua Yang, Syed Ghazi Sarwat, Abu Sebastian, Thomas Mikolajick, Stefan Slesazeck, Beatriz Noheda, Bernard Dieny, Tuo-Hung (Alex) Hou, Akhil Varri, Frank Brückerhoff-Plückelmann, Wolfram Pernice, Xixiang Zhang, Sebastian Pazos, Mario Lanza, Stefan Wiefels, Regina Dittmann, Wing H. Ng, Mark Buckwell, Horatio R. J. Cox, Daniel J. Mannion, Anthony J. Kenyon, Yingming Lu, Yuchao Yang, Damien Querlioz, Louis Hutin, Elisa Vianello, Sayeed Shafayet Chowdhury, Piergiulio Mannocci, Yimao Cai, Zhong Sun, Giacomo Pedretti, John Paul Strachan, Dmitri Strukov, Manuel Le Gallo, Stefano Ambrogio, Ilia Valov, and Rainer Waser. Roadmap to neuromorphic computing with emerging technologies. *APL Materials*, 12(10): 109201, 10 2024. ISSN 2166-532X.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2381–2391, 2018.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Seongsik Park, Seijoon Kim, Byunggook Na, and Sungroh Yoon. T2fsnn: Deep spiking neural networks with time-to-first-spike coding. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2020.

Juan-Carlos Perez and Enrique Vidal. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15(8):743–750, 1994. ISSN 0167-8655.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

Bodo Rueckauer and Shih-Chii Liu. Conversion of analog to spiking neural networks using sparse temporal coding. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2018.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, Volume 11 - 2017, 2017. ISSN 1662-453X. doi: 10.3389/fnins.2017.00682.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252, 2015.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Larry Schumaker. *Spline Functions: Basic Theory*. Cambridge Mathematical Library. Cambridge University Press, 3 edition, 2007.

Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019.

Ana Stanojevic, Stanisław Woźniak, Guillaume Bellec, Giovanni Cherubini, Angeliki Pantazi, and Wulfram Gerstner. An exact mapping from relu networks to spiking neural networks. *Neural Networks*, 168:74–88, 2023. ISSN 0893-6080.

Ana Stanojevic, Stanisław Woźniak, Guillaume Bellec, Giovanni Cherubini, Angeliki Pantazi, and Wulfram Gerstner. High-performance deep spiking neural networks with 0.3 spikes per neuron. *Nature Communications*, 15(1):6793, 2024. doi: 10.1038/s41467-024-51110-5.

Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3(3):230–238, Mar 2021.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pp. 6105–6114. PMLR, 2019.

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. Deep learning's diminishing returns: The cost of improvement is becoming unsustainable. *IEEE Spectrum*, 58(10): 50–55, 2021. doi: 10.1109/MSPEC.2021.9563954.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, 2017.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

J. Wang, X. Deng, W. Wei, D. Zhang, S. Wang, Q. Sun, and M. Zhang. Training-free ann-to-snn conversion for high-performance spiking transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2128–2136, 2026.

Tianqi Wang, Qianzi Shen, Xuhang Li, Yanting Zhang, Zijian Wang, and Cairong Yan. Precise spiking neurons for fitting any activation function in ann-to-snn conversion. *Applied Intelligence*, 55(7):463, Feb 2025. ISSN 1573-7497.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Kang You, Zekai Xu, Chen Nie, Zhijie Deng, Qinghai Guo, Xiang Wang, and Zhezhi He. SpikeZIP-TF: Conversion is all you need for transformer-based SNN. In *Forty-first International Conference on Machine Learning*, 2024.

Lei Zhang, Shengyuan Zhou, Tian Zhi, Zidong Du, and Yunji Chen. Tdsnn: From deep neural networks to deep spike neural networks with temporal-coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1319–1326, 2019.

Lusen Zhao, Zihan Huang, Jianhao Ding, and Zhaofei Yu. TTFSFormer: A TTFS-based lossless conversion of spiking transformer. In *Forty-second International Conference on Machine Learning*, 2025.

Hanyuan Zheng, Haozhen Zhang, Tianshuo Chen, Zhaogeng Liu, Yi Chang, and Bin Gu. Distribution-aware multi-granularity phase coding: Towards lower conversion error for spike-driven large language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

A MORE DETAILS ON PWFS CONVERSION FRAMEWORK

This appendix provides more details on the PWFS conversion: Subsection A.1 presents a pseudo code for the algorithm in Section 4 in the main text, while Subsection A.2 and A.3 discuss the offline phase as well as the boundary input values.

A.1 SUMMARIZING ALGORITHM

Algorithm 1: Piecewise FS (PWFS) Neuron Inference

Input: $x \in \mathbb{R}$; knot points $x_1 < \cdots < x_{M+1}$; $y_m = f(x_m)$; step counts $M, K \in \mathbb{N}$
Output: Approximation $y \approx f(x)$

// **Phase 1: Subinterval Identification**
$v(1) \leftarrow x$
has_spiked $\leftarrow 0$
for $m = 1$ **to** M **do**
  $\theta(m) \leftarrow x_{M-m+1} + \infty \cdot$ has_spiked
  $h(m) \leftarrow x_{M-m+1}$
  $s(m) \leftarrow H\Big(v(m) - \theta(m)\Big)$
  $v(m+1) \leftarrow v(m) - h(m) \cdot s(m)$
  has_spiked $\leftarrow$ has_spiked $\vee\, s(m)$

// **Phase 2: Dyadic FS**
$E \leftarrow \sum_{m=1}^{M} (x_{m+1} - x_m) s(m)$
$A \leftarrow \sum_{m=1}^{M} (y_{m+1} - y_m) s(m)$
$B \leftarrow \sum_{m=1}^{M} y_m s(m)$
$v'(1) \leftarrow v(M+1)$ `// Residual voltage from Phase 1`
for $t = 1$ **to** K **do**
  $h'(t) \leftarrow \theta'(t) \leftarrow \frac{E}{2^t}$
  $s'(t) \leftarrow H\Big(v'(t) - \theta'(t)\Big)$
  $v'(t+1) \leftarrow v'(t) - h'(t) \cdot s'(t)$

// **Output Reconstruction**
$d(0) \leftarrow B$
for $t = 1$ **to** K **do**
  $d'(t) \leftarrow \frac{A}{2^t}$
$y \leftarrow d(0) + \sum_{t=1}^{K} d'(t) s'(t)$
return y

A.2 OFFLINE PREPARATION DETAILS

In this section, we review existing literature on knot placement strategies in piecewise linear approximation, demonstrating that condition (4), which is required for our PWFS error estimation, is mild and readily satisfied by standard knot selection algorithms.

The task of identifying optimal knots for piecewise linear or spline approximators of a smooth target function $f : [a, b] \to \mathbb{R}$ is a classic problem in numerical analysis (de Boor, 1973; Burden & Faires, 2010; de Boor, 1978; DeVore & Lorentz, 1993; Camponogara & Nazari, 2015; Schumaker, 2007). In particular, the following local error bound follows from standard interpolation theory

Lemma A.1 (de Boor (1973); Schumaker (2007))**.** *Let $a, b \in \mathbb{R}$, $f : [a, b] \to \mathbb{R}$ be a continuously differentiable function. Consider arbitrary knot points $a =: x_1 < \ldots < x_M < x_{M+1} := b$ for some $M \in \mathbb{N}$. Let $f_M : [a, b] \to \mathbb{R}$ be the corresponding piecewise linear interpolator. Then for every $m \in [M]$ it holds*

$$\max_{x \in [x_m, x_{m+1}]} |f(x) - f_M(x)| \leq \frac{1}{8} (x_{m+1} - x_m)^2 \max_{x \in [x_m, x_{m+1}]} |f''(x)| \,.$$

Lemma A.1 implies that even an equidistant grid yields an approximation error of $\mathcal{O}(M^{-2})$:

$$\|f - f_M\|_\infty \leq \frac{(b-a)^2}{8M^2} \|f''\|_\infty .$$

While this naive grid choice is obviously suboptimal in general, we can see that the rate of $\mathcal{O}(M^{-2})$ required by Theorem 4.2 is easily achievable. Furthermore, this simple baseline already outperforms the $\mathcal{O}(M^{-1})$ rate obtained by conventional rate-based conversion methods.

More generally, the local bound in Lemma A.1 guarantees a global $\mathcal{O}(M^{-2})$ uniform error (Schumaker, 2007) under a mild mesh regularity condition,

$$\frac{\max_m (x_{m+1} - x_m)}{\min_m (x_{m+1} - x_m)} \leq \mathcal{O}(1), \quad \text{(i.e. bounded independent of } M),$$

or simply $\max_m (x_{m+1} - x_m) = \mathcal{O}(M^{-1})$.

To optimize knot selection in practice, de Boor (1973) proposed *de Boor's principle* of curvature equidistribution, placing knots such that:

$$\int_{x_m}^{x_{m+1}} |f''(x)|^{1/2} \, dx = \frac{1}{M} \int_a^b |f''(x)|^{1/2} \, dx \quad \text{(i.e. constant in } m).$$

More generally, for any monitor function $r : [a, b] \to \mathbb{R}$ satisfying $\inf_{x \in (a,b)} r(x) > 0$, selecting knots according to

$$\int_{x_m}^{x_{m+1}} r(x)dx = \frac{1}{M} \int_a^b r(x)dx \quad \text{(constant in m)},$$

preserves the optimal $\mathcal{O}(M^{-2})$ asymptotic decay rate, with the leading constant depending on $\inf_x r(x)$ (de Boor, 1973). Alternative knot placement strategies include dynamic programming algorithms (Bellman, 1961; Perez & Vidal, 1994) and heuristic offline tuning procedures similar to standard FS neurons.

A.3 Boundary cases

We consider the algorithm in the exceptional cases where the input value x does not belong to any subinterval $[x_m, x_{m+1})$ for any $m \in [M]$. These correspond to the boundary cases, where $x > b = x_{M+1}$ and $x < x_1$.

First note that the case $x > x_{M+1}$ can actually be treated as x belongs to the last subinterval $[x_M, x_{M+1})$. Indeed, the first inference produces exactly one spike (at the first time step) and the rest voltage $x - x_M$. Since this surpasses every threshold $(x_{M+1} - x_m)/2^t$ in the second phase, this returns the all-one output spike train. The neuron output thus becomes

$$g(x) = g(x_{M+1}).$$

Now consider the case $x < x_1$, which leads to no spike at all in Phase 1 with the rest voltage equal to x. This means that the scalars A and B in Phase 2 both vanish, and thus, the final output approximation is $y = 0$.

Note that this matches the way the original FS neurons work: if the gate input x is too small, then there is no input spike, the neuron remains silent, and outputs zero. This intuition works well for most practical activation functions, as they map highly negative inputs to approximately zero. In cases of other activations for which this does not hold true, one can simply modify the method by specifying a reference output value y_0 for the very negative inputs and set

$$B = y_0 + \sum_{m=1}^{M} (y_m - y_0)s(m).$$

This obviously makes no change to the usual cases, where x belongs to one of the intervals $[x_m, x_{m+1})$.

B THEORETICAL ANALYSIS OF IF NEURONS

Here we provide the proof of Proposition 3.2 from Section 3.

Proposition B.1 (Convergence of IF-neurons with rate-coded inputs)**.** *Let $x \in \mathbb{R}$, $\big(x(t)\big)_{t=1}^{\infty} \subset \mathbb{R}$ be a sequence and $C > 0$ be a constant such that*

$$\left| \frac{1}{t} \sum_{i=1}^{t} x(i) - x \right| \leq \frac{C}{t} \quad \forall t \in \mathbb{N}. \tag{6}$$

Let $\theta > 0$ and let $G = G_{u(0),\theta} : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$ be an IF neuron that satisfies

$$\frac{u(0) - C}{\theta} - 1 \leq 0 \leq \frac{u(0) + C}{\theta}. \tag{7}$$

Let $\big(s(t)\big)_{t\in\mathbb{N}} = G\Big(\big(x(t)\big)_{t\in\mathbb{N}}\Big)$. Then for some constant $C' > 0$ depending only on C, $u(0)$ and θ, it holds

$$\left| \frac{1}{t} \sum_{i=1}^{t} s(i) - \mathrm{ReLU1}\left(\frac{x}{\theta}\right) \right| \leq \frac{C'}{t}.$$

Proof. First, we rewrite the assumption (6) on the input sequence $\big(x(t)\big)_t$ as

$$tx - C \leq \sum_{i=1}^{t} x(i) \leq tx + C \quad \forall t \in \mathbb{N}. \tag{8}$$

We denote by $u(t)$ and $s(t)$ the membrane potential and spike activation as in Definition 3.1. By the IF dynamics, it holds

$$u(i) = u(i-1) + x(i) - \theta s(i), \quad \forall i \in \mathbb{N}.$$

Taking the sum over $i = 1, \ldots, t$, we obtain

$$u(t) = u(0) + \sum_{i=1}^{t} x(i) - \theta \sum_{i=1}^{t} s(i),$$

and hence,

$$s(t+1) = H\Big(u(t) + x(t+1) - \theta\Big) = H\left(u(0) + \sum_{i=1}^{t+1} x(i) - \theta\left(1 + \sum_{i=1}^{t} s(i)\right)\right). \tag{9}$$

Now we will show by induction over $t \in \mathbb{N}$ that

$$t\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0) - C}{\theta} - 1 \leq \sum_{i=1}^{t} s(i) \leq t\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0) + C}{\theta}. \tag{10}$$

For $t = 0$, the spike count vanishes and (10) holds true due to the assumption (7). For the induction step $t \to t+1$, assume that the bound (10) holds as the induction hypothesis, we aim to prove

$$(t+1)\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0) - C}{\theta} - 1 \leq \sum_{i=1}^{t+1} s(i) \leq (t+1)\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0) + C}{\theta}.$$

- **Lower bound:**
  - **Case $s(t+1) = 1$:** By induction hypothesis, we have

$$\begin{aligned} \sum_{i=1}^{t+1} s(i) = 1 + \sum_{i=1}^{t} s(i) &\geq 1 + t\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0) - C}{\theta} - 1 \\ &\geq (t+1)\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0) - C}{\theta} - 1. \end{aligned}$$

– **Case** $s(t+1)=0$**:** By the derivation (9) of $s(t+1)=0$, it follows that

$$1+\sum_{i=1}^{t+1} s(i) = 1+\sum_{i=1}^{t} s(i) > \frac{u(0)+\sum_{i=1}^{t+1} x(i)}{\theta} \overset{(8)}{\geq} (t+1)\frac{x}{\theta} + \frac{u(0)-C}{\theta}.$$

In case $x \geq 0$, this means

$$\sum_{i=1}^{t+1} s(i) > (t+1)\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0)-C}{\theta} - 1.$$

If $x<0$, then $\mathrm{ReLU1}\,(x)=0$, it follows

$$\sum_{i=1}^{t+1} s(i) \geq 0 \overset{(7)}{\geq} \frac{u(0)-C}{\theta} - 1 = (t+1)\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0)-C}{\theta} - 1.$$

- **Upper bound:**

– **Case** $s(t+1)=1$**:** From the formulation (9) of $s(t+1)=1$ it follows

$$\sum_{i=1}^{t+1} s(i) = 1+\sum_{i=1}^{t} s(i) \leq \frac{u(0)+\sum_{i=1}^{t+1} x(i)}{\theta} \overset{(8)}{\leq} (t+1)\frac{x}{\theta} + \frac{u(0)+C}{\theta}.$$

If $x \leq \theta$, which means $\frac{x}{\theta} \leq \mathrm{ReLU1}\left(\frac{x}{\theta}\right)$, this yields

$$\sum_{i=1}^{t+1} s(i) \leq (t+1)\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0)+C}{\theta}.$$

If $x>\theta$, we have

$$\sum_{i=1}^{t+1} s(i) \leq t+1 \overset{(7)}{\leq} (t+1)\underbrace{\mathrm{ReLU1}\left(\frac{x}{\theta}\right)}_{=1} + \frac{u(0)+C}{\theta}.$$

– **Case** $s(t+1)=0$**:** By induction hypothesis, we get

$$\sum_{i=1}^{t+1} s(i) = \sum_{i=1}^{t} s(i) \leq t\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0)+C}{\theta} \leq (t+1)\,\mathrm{ReLU1}\left(\frac{x}{\theta}\right) + \frac{u(0)+C}{\theta}.$$

To summarize, we have proven the bound (10) for any $t \in \mathbb{N}$ and deduce that

$$\frac{u(0)-C-\theta}{t\theta} \leq \frac{\sum_{i=1}^{t} s(i)}{t} - \mathrm{ReLU1}\left(\frac{x}{\theta}\right) \leq \frac{u(0)+C}{t\theta}.$$

which implies

$$\left|\frac{\sum_{i=1}^{t} s(i)}{t} - \mathrm{ReLU1}\left(\frac{x}{\theta}\right)\right| \leq \frac{C'}{t}$$

for $C' := \max\left\{\frac{u(0)-C-\theta}{\theta}, \frac{u(0)+C}{\theta}\right\} > 0$. □

C THEORETICAL ANALYSIS OF FS AND PWFS NEURONS

This section is dedicated to continue the theoretical analysis of FS and PWFS neurons from the main paper. In particular, we will show in Subsection C.1 that FS neurons can approximate linear patterns well, and in Subsection C.2 that they, however, struggle with highly nonlinear activation functions. Finally, based on the results in C.1, we provide a proof for Theorem 4.2 in Subsection C.3.

C.1 Compatibility of FS with ReLU and Linear Activations

Proposition C.1 (FS for linear activations)**.** *Let $T \in \mathbb{N}$, $B > 0$, $g_T : \mathbb{R} \to \mathbb{R}$ be an FS neuron with parameters $d(0) = 0$, $\theta(t) = d(t) = h(t) = 2^{-t}B$, $t \in [T]$ and $x \in (-\infty, B)$ be an arbitrary input. Then the spike train $\big(s(t)\big)_{t\in[T]}$ is the coefficients in the binary representation of $\frac{x}{B}$ in $[0, 1)$ with T bits. In particular, it holds*

$$|g_T(x) - \mathrm{ReLU}\,(x)| \leq 2^{-T}B \quad \text{for all } x \in (-\infty, B],$$

and $g_T(x) = \sum_{t=1}^{T} 2^{-t}B$ for $x > B$.

Proof. Without loss of generality, we set $B = 1$, as generalizing to arbitrary B requires straightforward rescaling. Let $x \in [0, 1]$ be arbitrary, we will show by induction over $t \in [T]$ that

$$0 \leq \underbrace{\mathrm{ReLU}\,(x)}_{=x} - \sum_{i=1}^{t} d(i)s(i) \leq 2^{-t}. \tag{11}$$

For $t = 0$, because the weighted spike sum is an empty sum, the inequality (11) becomes $0 \leq x \leq 1$, which holds trivially. Assume that (11) holds until $t \leq T - 1$, we will now prove the statement for $t + 1$. By definition, it holds

$$v(i + 1) = v(i) - h(i)s(i) = u(i) - 2^{-i}s(i),$$

which, by taking the sum over $i = 1, \ldots, t$, yields

$$v(t + 1) = v(1) - 2^{-i}s(i) = x - 2^{-i}s(i).$$

Inserting this into the definition of $s(t + 1)$ gives

$$s(t + 1) = H\Big(u(t + 1) - \theta(t + 1)\Big) = H\Big(x - \sum_{i=1}^{t} 2^{-i}s(i) - 2^{-t-1}\Big).$$

We consider two cases.

- Case $s(t + 1) = 1$, then it holds

$$1 \geq x \geq \sum_{i=1}^{t} 2^{-i}s(i) + 2^{-t-1}.$$

  This, together with the induction hypothesis, namely $x - \sum_{i=1}^{t} 2^{-i}s(i) \leq 2^{-t}$, implies

$$0 \leq x - \sum_{i=1}^{t+1} 2^{-i}s(i) = x - \sum_{i=1}^{t} 2^{-i}s(i) - 2^{-t-1} \leq 2^{-t-1}.$$

- Case $s(t + 1) = 0$, then it holds

$$0 \leq x < \sum_{i=1}^{t} 2^{-i}s(i) + 2^{-t-1}.$$

  Combining this with the induction hypothesis, namely $0 \leq x - \sum_{i=1}^{t} 2^{-i}s(i)$, we obtain

$$0 \leq x - \sum_{i=1}^{t} 2^{-i}s(i) = x - \sum_{i=1}^{t+1} 2^{-i}s(i) \leq -2^{-t-1}.$$

Finally, we consider inputs that lie outside of the 'trust region', i.e. $x \notin [0, 1]$. If $x \leq 0$, it is clear that $s(i) = 0$ for every $i \in [T]$, and similarly, if $x \geq 1$, it is easy to show that $s(i) = 1$ for every $i \in [T]$. □

Corollary C.2. *Let $a, b \in \mathbb{R}$, $f : [a, b] \to \mathbb{R}$ be an affine linear function. Let $T \in \mathbb{N}$, $g_T : \mathbb{R} \to \mathbb{R}$ be an FS neuron with hyperparameters*

$$\begin{aligned} d(0) &:= f(a), \\ d(t) &:= \Big(f(b) - f(a)\Big)2^{-t}, \\ h(t) &:= \theta(t) = (b-a)2^{-t} \end{aligned}$$

that starts with $v(1) := x - a$ for any $x \in [a, b]$. Then the dynamic

$$\begin{cases} s(t) & = H\Big(v(t) - \theta(t)\Big) \\ v(t+1) & = v(t) - h(t)s(t) \end{cases}$$

leads to $g_T(x) := d(0) + \sum_{t=1}^{T} d(t)s(t)$ that satisfies

$$|g_T(x) - f(x)| \le 2^{-T} L(b-a),$$

where $L := \left|\frac{f(b)-f(a)}{b-a}\right|$ is the absolute slope of f.

Proof. Note that the given dynamic can be rewritten as

$$s(t) = H\Big(v(t) - (b-a)2^{-t}\Big) = H\Big(v'(t) - 2^{-t}\Big)$$

with $v'(t) := \frac{v(t)}{b-a}$, starting with $v'(1) = \frac{x-a}{b-a} \in [0, 1]$. By Proposition C.1 we have that $\Big(s(t)\Big)_{t\in[T]}$ is the binary representation of $\frac{x-a}{b-a}$ and

$$\left|\sum_{t=1}^{T} 2^{-t} s(t) - \frac{x-a}{b-a}\right| \le 2^{-T}.$$

By the affine linearity of f on $[a, b]$, we have that $\frac{x-a}{b-a} = \frac{f(x)-f(a)}{f(b)-f(a)}$. Therefore, it holds

$$|g_T(x) - f(x)| = \left|\sum_{t=1}^{T} d(t)s(t) - \big(f(x) - f(a)\big)\right| \le 2^{-T} \left|f(b) - f(a)\right|.$$

□

C.2 Limitation of FS neurons

In this section, we examine the fundamental limitations of FS neurons in approximating highly nonlinear functions. For simplicity, we consider here a target function $f : [0, 1] \to \mathbb{R}$ with normalized input domain. Recall that an FS neuron operates as a parametric function

$$g(x) = g_\Theta(x) = d(0) + \sum_{t=1}^{T} d(t)s(t; x), \quad x \in [0, 1],$$

governed by the parameters $\Theta := \{d(0)\} \cup \{\theta(t), h(t), d(t)\}_{t\in[T]}$ with $s(t; x) = H\Big(x - \sum_{i=1}^{t-1} h(i)s(i) - \theta(t)\Big)$ being the step-wise firing mechanism. Unlike in PWFS, the parameter set Θ remains fixed during inference, i.e. independent of the input x. Below, we demonstrate how this rigidity severely restricts the approximation capacity of FS neurons.

Unlike existing literature (Jeong et al., 2024; Wang et al., 2026), which focuses on empirical training dynamics, we examine the intrinsic approximation capacity of FS neurons. Prior work largely emphasizes the practical difficulties of solving the empirical risk minimization problem

$$\mathcal{L}(\Theta) = \min_{\Theta} \quad \frac{1}{N} \sum_{i=1}^{N} \left|g_\Theta(x^{(i)}) - f(x^{(i)})\right|^2$$

evaluated over sample points $\{x^{(i)}\}_{i\in[N]} \subseteq [0,1]$. Specifically, as the Heaviside function H is non-differentiable, the numerical optimization relies on surrogate gradients, leaving the performance sensitive to parameter initialization and surrogate selection. Here, instead of focusing on optimization mechanics, we investigate a more fundamental question: whether the approximation error measured in the maximum norm can converge to zero as T increases.

Observe that in general g is a piecewise constant function, where the domain $[0,1]$, by the endpoints

$$P := \left\{ \sum_{i=1}^{t} h(i)s_i + \theta(t) : t \in [T], s_i \in \{0,1\} \right\},$$

is divided into subintervals, on each of which g stays constant. In general, it is difficult to analyze the function g, because the endpoints in P might not realize all possible 2^T subintervals in $[0,1]$, where some subintervals are empty due to the replacement of those endpoints.

Here we provide a result for a simplified setting, where P partitions the input domain fully 2^T different nonempty subintervals. While obtaining the highest resolution, this setting leads to a saturation in the approximation error measured in the maximum norm.

Proposition C.3. *Let $f : [0,1] \to \mathbb{R}$ be a twice continuously differentiable function that satisfies*

$$f''(x) \geq C_f \quad \text{for all } x \in [0,1] \text{ for some constant } C_f > 0, \tag{12}$$

and $g = g_\Theta$ be a FS neuron with parameters $\Theta = \{\theta(t), h(t), d(t)\}_{t=1}^{T} \in \mathbb{R}^T$, $T \in \mathbb{N}$. We assume that the set of endpoints P produced by the piecewise constant function g, namely

$$P := \left\{ \sum_{i=1}^{t-1} h(i)s_i + \theta(t) : t \in [T], s_i \in \{0,1\} \right\},$$

partitions the input domain $[0,1]$ into 2^T nonempty subintervals. Furthermore, assume that the parameters $\theta(t)$ are chosen such that

$$\max_{t\in[T]} \theta(t), \max_{t\in[T-1]} h(t) \geq C_\Theta \quad \text{for some constant } C_\Theta > 0. \tag{13}$$

Then asymptotically as $T \to \infty$, it holds

$$\|f - g\|_\infty \geq \frac{1}{8} C_f C_\Theta^2$$

Here we use the condition (12) to measure the oscilation of f, for simplicity we assume here a global condition. On the other hand, since $\{\theta(t), h(t)\}_t$ can be thought of as defining a multiscale system on the input domain, it is reasonable to assume that their maximum values are of constant order.

Proof.

Step 1: First we introduce some notations. Let $p : \{0,1\}^T \setminus \{0\} \to P$ be the bijective mapping defined by

$$S = (s_1, \ldots, s_{t_S-1}, 1, 0, \ldots, 0) \mapsto \sum_{i=1}^{t_S-1} h(i)s_i + \theta(t_S).$$

where

$$t_S := \min\{t \in [T] : s_t = 0 \ \forall t > t_S\}$$

is the last spike time in S. Furthermore, for a binary representation $S := (s_i)_{i\in[T]}$ we define $S \oplus 1$ to be the next binary number. We use the convention $p\big((0,\ldots,0)\big) = -\infty$ and $p\big((1,\ldots,1) \oplus 1\big) = \infty$.

By the assumption that P realizes all possible subintervals, it follows that $\big(h(t)\big)_t$ and $\big(\theta(t)\big)_t$ are both decreasing. Thus, in particular $h(1)$ and $\theta(1)$ are the largest among all T values.

Step 2: Let $x \in \big[p(S), p(S \oplus 1)\big)$ for some $S = (s_i)_{i\in[T]} \in \{0,1\}^T$ with $t_S < T$ and $s_j = 0$ for some fixed $j \in [t_S - 1]$. We will show that

$$g\big(x + h(j)\big) - g(x) = d(j), \tag{14}$$

for which it is sufficient to prove

$$x + h(j) \in \big[p(S'), p(S' \oplus 1)\big),$$

where S' is given by S with a flipped j-th bit, i.e. $s'_i = s_i \ \forall i \neq j$ and $s'_j = 1$.

Indeed, it holds

$$\begin{aligned} x + h(j) &\geq p(S) + h(j) \\ &= \sum_{i=1}^{t_S - 1} h(i)s_i + h(j) + \theta(t_S) \\ &= \sum_{i=1}^{t_{S'} - 1} h(i)s'_i + \theta(t_{S'}) \\ &= p(S'). \end{aligned}$$

On the other hand, observe that since $t_S < T$, the increment in the binary domain only involves the last bits, in particular the bits after t_S. Therefore, the increment of S' is obtained by flipping the j-th bit of S from 0 to 1 and hence we get

$$x + h(j) \leq p(S \oplus 1) + h(j) = p(S' \oplus 1).$$

Step 3: Now we specify $j = 1$ and let x, y be such that $x, y < \theta(1)$ or equivalently $s_1 = 0$ for both. Without loss of generality we assume $x < y$. By the argument in Step 2 we know that

$$g\big(x + h(1)\big) - g(x) = d(1) = g\big(y + h(1)\big) - g(y), \tag{15}$$

By the triangle inequality, we have

$$\begin{aligned} \|f - g\|_\infty &\geq \frac{1}{2}\Big(\big|g(x + h(1)) - f(x + h(1))\big| + \big|g(x) - f(x)\big| \Big) \\ &\geq \frac{1}{2}\Big| \Big(f\big(x + h(1)\big) - f(x)\Big) - \Big(g\big(x + h(1)\big) - g(x)\Big)\Big| \\ &\overset{(15}{=} \frac{1}{2}\Big| \Big(f\big(x + h(1)\big) - f(x)\Big) - d(1)\Big| \end{aligned}$$

and similarly

$$\|f - g\|_\infty \geq \frac{1}{2}\Big| \Big(f\big(y + h(1)\big) - f(y)\Big) - d(1)\Big|.$$

Combining these two inequalities yields

$$\begin{aligned} \|f - g\|_\infty &\geq \frac{1}{4}\Big| \Big(f\big(x + h(1)\big) - f\big(y + h(1)\big)\Big) - \Big(f(x) - f(y)\Big)\Big| \\ &= \frac{1}{4}(y - x) \cdot |f'(z_1) - f'(z_2)| \end{aligned}$$

where z_1 lies between x and y, and z_2 lies between $x + h(1)$ and $y + h(1)$. Applying the intermediate value theorem again implies

$$\|f - g\|_\infty \geq \frac{C_f}{4}(y - x)\Big(h(1) - (y - x)\Big).$$

Recall that x and y were chosen arbitrarily such that they are smaller than $\theta(1)$ and thus $y - x$ can take any value between 0 and $\theta(1)$. If $\theta(1) \geq \frac{h(1)}{2}$, then one could take $y - x = \frac{h(1)}{2}$ to get

$$\|f - g\|_\infty \geq \frac{C_f h(1)^2}{16} \geq \frac{C_f C_\Theta^2}{16}.$$

On the other hand, if $\theta(1) < \frac{h(1)}{2}$, one can choose $y - x = \theta(1)$ to get that

$$\|f - g\|_\infty \geq \frac{C_f}{4}\theta(1)\Big(h(1) - \frac{h(1)}{2}\Big) \geq \frac{C_f C_\Theta^2}{8}.$$

□

While the setting of Proposition C.3 seems a bit restrictive, it already explains why the intuitively reasonable choice $h(t) = \theta(t) = 2^{-t}$ does not work in general, as well as why the approach of pre-fixing $d(t) = 2^{-t}$ in (Wang et al., 2025) would require a look-up table rather than only single values of $\theta(t)$ and $h(t)$ at each time step.

On the other hand, the proof of Proposition C.3 reveals why a parameter set $\{d(t), \theta(t), h(t)\}$ that remains fixed during inference and independent of inputs fails to drive the FS approximation error to zero quickly when approximating highly non-linear target functions. Specifically, consider two inputs x and $x + h(j)s(j) \in [0, 1]$, corresponding to toggling a single spike $s(j)$ at time step j. While the input shift is $h(j)$, the resulting change in the approximator's output g is fixed at $d(j)$ due to the constant spike weighting. Consequently, the local slope $\frac{d(j)}{h(j)}$ remains invariant across all feasible input values x. While this static ratio suffices for affine-linear functions, it creates an irreconcilable structural mismatch when f deviates significantly from linearity.

C.3 Theoretical guarantee for PWFS conversion

Theorem C.4 (PWFS approximation guarantee)**.** *Let $f : [a, b] \to \mathbb{R}$ be a $\mathcal{C}^2$ (continuously twice differentiable) target function, $M, K \in \mathbb{N}$. Let $a = x_0 < x_1 < \ldots < x_{M+1} = b$ be the knot points chosen on $[a, b]$ such that the corresponding piecewise linear interpolator f_M satisfies*

$$\|f - f_M\|_\infty \leq CM^{-2}$$

for a positive constant $C > 0$ (depending on $b - a$ and f''). Let $g : [a, b] \to \mathbb{R}$ be the approximation to f computed by Algorithm 1. Then it holds

$$\|g - f\|_\infty \leq C'(M^{-2} + 2^{-K}),$$

where $C' > 0$ is a positive constant depending on $b - a$, $\|f''\|_\infty$ and $\|f'\|_\infty$.

Proof. Let $x \in [a, b]$ be arbitrary. Furthermore, let $m \in [M]$ be such that $x \in [x_m, x_{m+1})$. By construction, Phase 1 of Algorithm 1 returns $s(m) = 1$ and $s(i) = 0$ for every $i \in [M] \setminus \{m\}$ combined with $v(M + 1) = x - x_m$. Accordingly, we obtain

$$E = x_{m+1} - x_m, \quad A = f(x_{m+1}) - f(x_m), \quad B = f(x_m).$$

By Corollary C.2, it holds

$$|g(x) - f_M(x)| \leq |f(x_{m+1}) - f(x_m)|\, 2^{-K} \leq 2^{-K} \|f'\|_\infty (x_{m+1} - x_m).$$

Overall, we obtain

$$|g(x) - f(x)| \leq |g(x) - f_M(x)| + |f_M(x) - f(x)| \leq 2^{-K} \|f'\|_\infty (b - a) + CM^{-2}.$$

□

D Additional experimental results

As discussed in Section 5 in the main paper, FS caused severe convergence problems when trained on full activation range. Table 4 reports the input range we used for training FS parameters, as well as the input range for PWFS (same as measured from pretrained models).

Table 5 and 6 provide additional results on the experiments with NLU tasks from Section 5.

		FS		PWFS	
Model	Activation	Interval	K	Interval	(M, K)
Pythia-2.8B	GELU	$[-8, 8]$	12	$[-105, 92]$	$(14, 18)$
Pythia-6.9B	GELU	$[-8, 8]$	12	$[-54, 53]$	$(14, 18)$
GPT-2	GELU (tanh)	$[-8, 8]$	12	$[-38, 63]$	$(14, 18)$
Qwen3-1.7B-Base	SiLU	$[-8, 12]$	16	$[-95, 163]$	$(14, 18)$
Qwen3-8B-Base	SiLU	$[-8, 12]$	16	$[-10, 134]$	$(14, 14)$
ConvNeXt-T	GELU	$[-8, 8]$	12	$[-31, 30]$	$(14, 18)$
DeiT-S	GELU	$[-8, 8]$	12	$[-26, 22]$	$(14, 12)$
EfficientNet-B0	SiLU	$[-8, 12]$	16	$[-10, 40]$	$(14, 16)$
	Sigmoid	$[-10, 10]$	12	$[-24, 25]$	$(14, 12)$

Table 4: Neurons used per model and activation: FS uses fixed hand-fitted neurons, PWFS the most accurate neuron fitted on the model's measured input range.

Model	Conversion	K	M	T	PIQA	ARC-e	OBQA	Wino
pythia-2.8b	-	-	-	-	73.99±1.02	64.27±0.98	23.80±1.91	59.59±1.38
	FS	12	-	12	72.25±1.04	61.66±1.00	25.80±1.96	57.46±1.39
	PWFS	18	14	32	73.72±1.03	63.97±0.99	24.20±1.92	59.27±1.38
pythia-6.9b	-	-	-	-	75.24±1.01	67.13±0.96	25.80±1.96	60.62±1.37
	FS	12	-	12	74.21±1.02	64.27±0.98	25.40±1.95	60.62±1.37
	PWFS	18	14	32	74.97±1.01	67.34±0.96	26.00±1.96	61.25±1.37
Qwen3-1.7B	-	-	-	-	75.84±1.00	73.36±0.91	30.00±2.05	63.93±1.35
	FS	16	-	16	74.97±1.01	74.33±0.90	29.40±2.04	64.64±1.34
	PWFS	18	14	32	75.84±1.00	73.61±0.90	30.20±2.06	64.25±1.35
Qwen3-8B	-	-	-	-	79.33±0.94	81.61±0.79	32.60±2.10	71.82±1.26
	FS	16	-	16	79.16±0.95	80.81±0.81	32.60±2.10	70.72±1.28
	PWFS	14	14	28	79.11±0.95	82.74±0.78	31.20±2.07	72.69±1.25

Table 5: Zero-shot accuracy (%, mean ± standard error) on PIQA, ARC-e, OBQA, and Wino.

Model	Conversion	K	M	T	COPA	WSC	RTE
pythia-2.8b	-	-	-	-	79.00±4.09	77.66±2.53	47.29±3.01
	FS	12	-	12	77.00±4.23	73.63±2.67	49.82±3.01
	PWFS	18	14	32	79.00±4.09	77.29±2.54	48.01±3.01
pythia-6.9b	-	-	-	-	80.00±4.02	77.29±2.54	59.93±2.95
	FS	12	-	12	81.00±3.94	73.63±2.67	57.76±2.97
	PWFS	18	14	32	81.00±3.94	78.02±2.51	59.93±2.95
Qwen3-1.7B	-	-	-	-	82.00±3.86	78.02±2.51	75.81±2.58
	FS	16	-	16	80.00±4.02	76.92±2.55	71.84±2.71
	PWFS	18	14	32	82.00±3.86	80.22±2.42	77.62±2.51
Qwen3-8B	-	-	-	-	88.00±3.27	85.35±2.14	75.09±2.60
	FS	16	-	16	92.00±2.73	82.78±2.29	75.45±2.59
	PWFS	14	14	28	88.00±3.27	84.25±2.21	76.90±2.54

Table 6: Zero-shot accuracy (%, mean ± standard error) on COPA, WSC, and RTE.