RETHINKING THE SPATIOTEMPORAL DISTRIBUTION FOR HIGH-FIDELITY PARALLEL ANN-TO-SNN CONVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking Neural Networks (SNNs) have attracted increasing attention for their low power consumption and constant-time inference on neuromorphic hardware. Among existing approaches, ANN-to-SNN conversion is one of the most effective ways to obtain deep SNNs with accuracy comparable to traditional ANNs, and recent work has even extended it to *parallel* conversion, where the full spike train is emitted in a single pass. Despite this promise, we find that ANN-to-SNN parallel conversion suffers from severe performance degradation at ultra-low timesteps ($T \leq 4$), limiting its practical use. In this work, we analyze the source of this performance gap and demonstrate that it originates from assumptions in the standard quantization–clip–floor–shift (QCFS) formulation, which, under the one-shot firing rule, introduces a step-dependent bias. To overcome this, we propose a *distribution-aware parallel calibration* that corrects spatiotemporal mismatches while leaving the backbone and firing rule unchanged. Our method consists of two stages: (1) **spatial recalibration**, which adapts normalization layers to spike-domain statistics, and (2) **temporal correction**, which learns a per-channel, time-collapsed aggregated membrane potential bias to offset timestep-dependent errors. On ImageNet-1k, our approach boosts ResNet-18 top-1 accuracy from $\mathbf{25.20}\% \rightarrow \mathbf{62.28}\%$ at $T = 4$ and ResNet-34 from $\mathbf{50.67}\% \rightarrow \mathbf{68.23}\%$ at $T = 8$. These results demonstrate that revisiting—and correcting—standard QCFS premises in the *parallel setting* is essential for accurate, low-latency SNNs without retraining the backbone.

1 INTRODUCTION

Spiking Neural Networks (SNNs) are widely regarded as a promising paradigm for energy-efficient computation, thanks to their sparse, event-driven communication that mirrors biological neurons. Among deployment strategies, *ANN-to-SNN conversion* has emerged as one of the most effective techniques, as it leverages the maturity of deep learning frameworks to achieve strong performance without the cost of direct SNN training (Bu et al., 2022). Within this paradigm, *parallel conversion* (Hao et al., 2025) is particularly attractive: by generating the entire spike train in a single pass, it enables inference with constant time complexity, regardless of the number of timesteps (T). This property makes ultra-low-latency inference (e.g., $T \leq 4$) theoretically attainable.

In practice, however, parallel conversion strategies suffer from accuracy degradation, especially when converting modern architectures dominated by ReLU activations. The core issue is a *distributional mismatch*: ANN activations are continuous-valued and static, while SNN representations are discrete and rely on temporally averaged firing rates. Conversion assumes that firing rates faithfully approximate ANN activations, but this assumption breaks down under thresholding and discrete dynamics. The resulting approximation errors accumulate across layers, severely reducing accuracy.

We illustrate this phenomenon in Figure 1, which compares activations from a pre-trained ResNet-18 and its converted parallel SNN counterpart. At $T = 4$, the SNN firing-rate distributions are compressed, shifted, and markedly sparser, indicating a loss of representational richness. Increasing T to 16 partially alleviates the mismatch but undermines the benefit of low latency. Thus, the

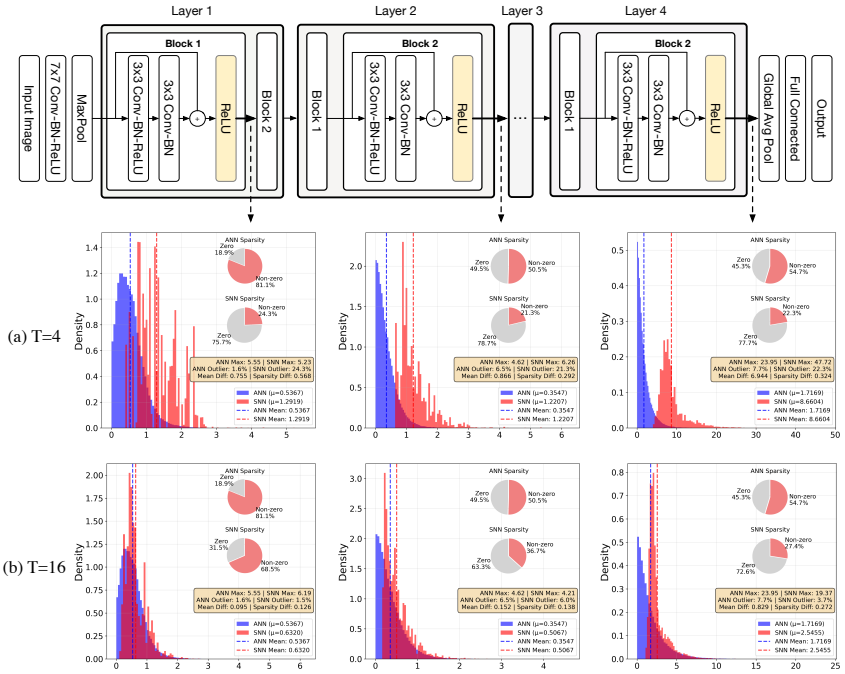

Figure 1: **Distributional Mismatch Between ANN Activations and SNN Firing Rates in ResNet-18.** We compare the output distributions of three representative ReLU/*Parallel IF* layers — early (2nd), middle (8th), and final (16th) layer — from a pre-trained ANN with their counterparts in *converted* parallel SNNs (Hao et al., 2025). **(a)** The blue histograms show the distribution of the original ANN's ReLU activation values. **(b)** The red histograms show the distribution of the SNN's average firing rate. The analysis is conducted at ultra-low timesteps of $T = 4$ (top) and $T = 16$ (bottom). At $T = 4$, the SNN firing rates are compressed and misaligned with the ANN's, and activation sparsity increases dramatically (e.g., from 18.9% to 75.7% in the 2nd ReLU/*Parallel IF*). Increasing the timestep to $T = 16$ partially mitigates this shift but fails to fully restore the original distribution, highlighting a fundamental challenge in low-latency conversion that our work addresses.

coupled *spatiotemporal distributional shift* between ANN activations and SNN responses emerges as the central obstacle to accurate, low-latency inference.

Prior work has offered partial remedies, typically addressing spatial and temporal mismatches in isolation. In contrast, we explicitly target their interplay by introducing a correction framework that disentangles and compensates for both sources of error. Our key insight is that high-fidelity parallel conversion requires a two-stage strategy: (1) recalibrating spatial statistics to align BatchNorm layers with the spike domain, and (2) introducing a temporal correction that offsets systematic biases caused by discrete-time dynamics.

Concretely, we propose a two-stage strategy. First, a **Spatial Recalibration**, which updates BatchNorm statistics and affine parameters using SNN activations, thereby stabilizing feature distributions across layers. The second stage, **Temporal Correction**, introduces a lightweight, learnable bias to membrane potentials. Trained with surrogate gradients, this bias shifts potential trajectories to counter discretization errors while preserving the constant-time property of parallel inference.

Our contributions are threefold:

1. We identify and empirically characterize the spatiotemporal distributional shift that underlies the performance gap in parallel ANN-to-SNN conversion.
2. We propose a two-stage correction framework—spatial recalibration of BatchNorm statistics and temporal bias correction—that jointly resolves these discrepancies.
3. Our approach is highly efficient, requiring fine-tuning of only a small parameter subset. On ImageNet, we raise the top-1 accuracy of ResNet-18 at $T = 4$ from 25.20% to **62.28**%

(ANN accuracy is 69.76%), and ResNet-34 at $T = 8$ from 50.67% to 68.23%, setting a new state of the art for ultra-low-latency parallel conversion.

2 Related Work

Research on ANN-to-SNN conversion and low-latency spiking inference has advanced along two tightly coupled directions: (i) parallel and efficient architectures that eliminate temporal bottlenecks, and (ii) methods that mitigate spatial and temporal mismatches introduced by quantization, thresholds, and discrete dynamics. On the parallelization side, constant-time spiking computation and conversion frameworks demonstrate that inference complexity can be decoupled from the number of timesteps, enabling ultra-low-latency inference Hao et al. (2025). Complementary efforts develop training and architectural techniques tailored for parallel SNNs, including constant-time parallel training Feng et al. (2025), multi-parallel implicit stream architectures for efficient optimization Cao et al. (2024), Spiking State-Space Models for long sequence processing Shen et al. (2025), parallel spiking neurons designed to capture long-term dependencies Fang et al. (2023), and temporally reversible SNNs that trade training memory for $O(1)$ inference Hu et al. (2024). While these approaches substantially reduce simulation and training cost, they primarily address computational efficiency rather than conversion fidelity.

A complementary line of work focuses directly on improving fidelity in ANN-to-SNN conversion. Early studies identified errors from clipping, flooring, and thresholding, and introduced remedies such as threshold balancing and quantization-aware adjustments Bu et al. (2022). More recent methods address temporal mismatches: for example, initial membrane-potential shifts and offset-spike calibration mitigate one-spike discrepancies Hao et al. (2023a), while forward temporal bias calibration introduces timestep-wise biases to correct firing-rate drift without costly backpropagation through time (BPTT) Wu et al. (2024). Other strategies exploit distillation or attention to align intermediate ANN and SNN features Hong & Wang (2025), or adopt phase-coding and one-spike encoding schemes to minimize conversion loss Hwang & Kung (2024). Beyond conversion, *Guo et al.* introduce MPBN Guo et al. (2023)—a *direct-from-scratch* SNN training method that normalizes the pre-firing membrane potential and folds the normalization into firing thresholds for inference, unlike conversion methods that calibrate around fixed ANN features.

3 Preliminaries

LIF Neuron Dynamics and Serial Computation. We consider an L-layer Spiking Neural Network (SNN) composed of Leaky Integrate-and-Fire (LIF) neurons Hodgkin & Huxley (1952), evolving over T discrete time steps. For a neuron in layer l at time t, its dynamics are governed by:

$$\mathbf{I}^{l,t} = \mathbf{W}^l \mathbf{s}^{l-1,t} \theta^{l-1}, \qquad \mathbf{v}_{\text{pre}}^{l,t} = \lambda^l \mathbf{v}^{l,t-1} + \mathbf{I}^{l,t}, \tag{1a}$$

$$\mathbf{s}^{l,t} = \Theta\big(\mathbf{v}_{\text{pre}}^{l,t} - \theta^l\big), \qquad \mathbf{v}^{l,t} = \mathbf{v}_{\text{pre}}^{l,t} - \mathbf{s}^{l,t}\theta^l. \tag{1b}$$

where $\mathbf{v}^{l,t}$ is the membrane potential, $\mathbf{I}^{l,t}$ is the input current, $\mathbf{s}^{l,t}$ is the binary spike train, θ^l is the firing threshold, and λ^l is the leakage factor. The temporal recurrence in (1) necessitates sequential computation across timesteps, making it the primary performance bottleneck for conventional SNN inference.

Rate-Based Conversion and QCFS Activation. ANN-to-SNN conversion methods bridge the two paradigms by equating ANN activations with SNN firing rates. A central tool for enabling low-latency, high-fidelity conversion is the Quantization–Clip–Floor–Shift (QCFS) activation Bu et al. (2022):

$$\mathbf{a}_{\text{QCFS}}^l = \frac{\theta^l}{\tilde{T}} \operatorname{Clip}\left(\left\lfloor \frac{\mathbf{W}^l \mathbf{a}^{l-1} \tilde{T} + \boldsymbol{\psi}^l}{\theta^l} \right\rfloor, 0, \tilde{T} \right), \tag{2}$$

where $\boldsymbol{\psi}^l$ is a learnable channel-wise shift. This formulation defines an integer spike target

$$k^l = \operatorname{Clip}\left(\left\lfloor \frac{\mathbf{W}^l \mathbf{a}^{l-1} \tilde{T} + \boldsymbol{\psi}^l}{\theta^l} \right\rfloor, 0, \tilde{T} \right), \tag{3}$$

to be matched by the converted SNN.

A distribution-aware variant of QCFS (DA-QCFS) was proposed in Hao et al. (2025), which augments QCFS with channel-wise scaling and shifting to better match activation statistics:

$$\mathbf{a}_{\mathrm{DA}}^{l,\tilde{T}} = \frac{\theta^l + \phi_{\mathrm{DA}}^l}{\tilde{T}} \mathrm{Clip}\left(\left\lfloor \frac{(\mathbf{W}^l \mathbf{a}^{(l-1),\tilde{T}} + \psi_{\mathrm{DA}}^l)\tilde{T} + \boldsymbol{\psi}^l}{\theta^l} \right\rfloor, 0, \tilde{T} \right). \tag{4}$$

Here $\phi_{\mathrm{DA}}^l, \psi_{\mathrm{DA}}^l \in \mathbb{R}^C$ denote learnable per-channel scaling and shifting factors (C is the number of channels). They are iteratively optimized using mean conversion errors before and after activation, aiming to align SNN firing-rate distributions with those of the pretrained ANN. Nonetheless, such distribution reshaping remains *suboptimal*, as it does not fully address the structural mismatch between ANN activations and SNN firing statistics.

Assumptions underpinning QCFS. Two premises are standard when analyzing the "optimal" shift and the zero-mean conversion error in (2) and (3). *Parameter matching:* the same threshold parameter θ^l is used both to scale ANN activations in (2) and as the firing threshold in the SNN dynamics of (1b). In addition, the membrane potential is initialized at the bin midpoint, $\mathbf{v}^{l,0} = \frac{1}{2}\theta^l$, so that the shift term $\boldsymbol{\psi}^l$ corresponds to mid-bin rounding. *Uniform in-bin statistics:* the pre-activation $\mathbf{W}^l \mathbf{a}^{l-1}$ (or its integer quantization) is assumed to be uniformly distributed within each quantization interval $\left[(m-1)\frac{\theta^l}{\tilde{T}},\, m\frac{\theta^l}{\tilde{T}}\right]$, for $m = 1, \ldots, \tilde{T}$. Under these premises, setting $\boldsymbol{\psi}^l$ to the mid-level value $\frac{1}{2}\theta^l$ makes rounding errors cancel in expectation, yielding zero expected conversion error for arbitrary T and $\tilde{T}$, and exact equality when $T = \tilde{T}$.

The Parallel Conversion Framework and its Lossless Mapping. Recently, the QCFS conversion framework was extended for parallel inference (Hao et al., 2025), i.e., enabling the computation of spike targets in *one-shot* rather than sequentially. Concretely, the entire output spike train $\mathbf{s}^l = [\mathbf{s}^{l,1}, \ldots, \mathbf{s}^{l,T}]^\top$ is generated through a single thresholding operation:

$$\mathbf{s}^l = \Theta\left(\boldsymbol{\Lambda}_{\mathrm{pc}}^l \mathbf{I}^l + \mathbf{b}^l - \theta^l\right), \tag{5}$$

where $\mathbf{I}^l$ is the temporal sequence of input currents and

$$\boldsymbol{\Lambda}_{\mathrm{pc}}^l = \begin{bmatrix} \frac{1}{T} & \cdots & \frac{1}{T} \\ \frac{1}{T-1} & \cdots & \frac{1}{T-1} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}, \qquad \mathbf{b}^l = \left[\frac{\boldsymbol{\psi}^l}{T}, \frac{\boldsymbol{\psi}^l}{T-1}, \ldots, \boldsymbol{\psi}^l\right]^\top. \tag{6}$$

A neuron fires at time t iff

$$[\boldsymbol{\Lambda}_{\mathrm{pc}}^l \mathbf{I}^l]_t + b^{l,t} \geq \theta^l \iff \frac{1}{T-t+1}\left(\sum_{j=1}^{T} \mathbf{I}^{l,j} + \boldsymbol{\psi}^l\right) \geq \theta^l \iff \sum_{j=1}^{T} \mathbf{I}^{l,j} + \boldsymbol{\psi}^l \geq (T-t+1)\theta^l, \tag{7}$$

and, using $\sum_j \mathbf{I}^{l,j} \approx \mathbf{W}^l \mathbf{a}^{l-1} T$ with $T = \tilde{T}$ for simplicity, the total spike count equals the QCFS target:

$$\sum_{t=1}^{T} s^{l,t} = \mathrm{Clip}\left(\left\lfloor \frac{\mathbf{W}^l \mathbf{a}^{l-1} T + \boldsymbol{\psi}^l}{\theta^l} \right\rfloor, 0, T \right) = k^l. \tag{8}$$

Thus, the *lossless* mapping of the parallel conversion is anchored in the same target construction as QCFS. In particular, analyses of its lossless and *sorting* properties, as well as derivations of the *optimal shifting distance* per step, implicitly depend on QCFS premises: *parameter matching* between activation range and threshold, and *uniform in-bin* statistics for rounding. As we show next, these assumptions are fragile in practice, especially at ultra-low T—and the aggregation weights in (5) make the precomputed *time-invariant* shift ψ^l suboptimal. This motivates our distribution-aware calibration in Sec. 4, which corrects the resulting step-dependent bias without altering the constant-time mapping.

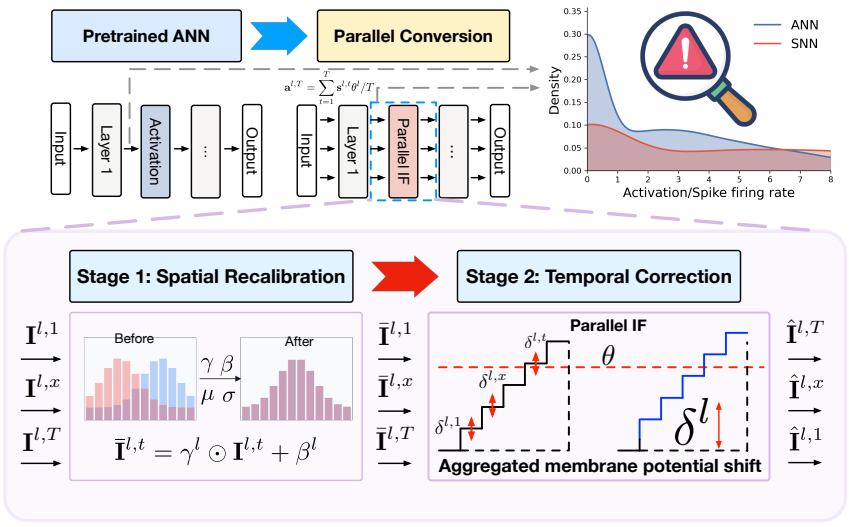

Figure 2: **Overall pipeline motivated by the *challenges* in parallel ANN-to-SNN conversion. Left (Observed issues).** Converting a pretrained ANN to a *parallel* SNN replaces serial accumulation with a one–shot test against a *descending threshold ladder* $(T-t+1)\theta$, which compresses activation ranges and amplifies late–timestep sensitivity. This creates two key discrepancies: (i) a *spatial* shift, as normalization layers expect ANN statistics but encounter spike–domain ones; and (ii) a *temporal* bias, as rounding residuals are distributed unevenly across timesteps. **Right (Remedy).** The pipeline addresses these causes by aligning operator–induced feature moments with backbone expectations and compensating ladder–induced bias in a channel–wise, time–collapsed form. By correcting the *why*—statistical shift and threshold asymmetry—it restores consistency at ultra–low T while preserving constant–time, one–shot firing.

4 METHOD

Design goal. In this work, we aim to recover high accuracy for *parallel* ANN-to-SNN conversion at ultra-low T, while preserving constant-time inference and keeping the one-shot firing rule unchanged. We identify two sources of error undermine fidelity in this regime. First, a *spatial mismatch* arises when BatchNorm layers apply ANN-trained statistics to spike-domain activations, introducing mean and variance distortions that accumulate across depth (Sec. 4.1). Second, we discuss that the parallel operator compares a single accumulated statistic against a descending threshold ladder $(T-t+1)\theta$, which compresses the effective decision margin and renders late timesteps disproportionately decision-critical (Sec. 4.2). At small T, this induces a systematic *temporal skew*: spikes are deferred and concentrate in the last k^l slots, yielding a biased distribution (11) even though the total spike count matches the QCFS target. This skew is directly tied to the harmonic weighting $(T-t+1)^{-1}$ that governs stepwise sensitivities, effectively suppressing early contributions while magnifying late ones. To mitigate the effects of this imbalance, we propose a temporal correction bias.

4.1 MOTIVATION I: CALIBRATION ERROR OF ANN-TO-SNN CONVERSION

A central challenge in ANN-to-SNN conversion lies in the distribution shift after conversion. Empirical evidence shows that this assumption that pre–BatchNorm activations retain the same distribution breaks down, particularly at ultra-low timesteps T. In these regimes, spike-domain activations exhibit increased variance, causing their statistical moments to deviate from those observed in the ANN. These errors grow as the timestep T decreases, since the quantization step $\Delta^l = \theta^l/T$ enlarges and clipping becomes more frequent. As a result, BatchNorm layers introduce a systematic biases that accumulate across layers.

Stage 1: Operator-Aware Spike-Domain Statistics Alignment (OASSA). To correct this mismatch, we propose to *recalibrate* BN statistics under the same operator used at inference. Con-

cretely, we switch BN layers into training mode and forward a small calibration set $\mathcal{D}_{\text{cal}}$ through the SNN, yielding to new estimates

$$\widehat{\mu}_s^l \;=\; \frac{1}{|\mathcal{D}_{\text{cal}}|}\sum_{x\in\mathcal{D}_{\text{cal}}}\Phi_{\text{par}}^l(x), \qquad (\widehat{\sigma}_s^l)^2 \;=\; \frac{1}{|\mathcal{D}_{\text{cal}}|}\sum_{x\in\mathcal{D}_{\text{cal}}}\big(\Phi_{\text{par}}^l(x)-\widehat{\mu}_s^l\big)^2. \tag{9}$$

We replace $(\mu^l,\sigma^l)\leftarrow(\widehat{\mu}_s^l,\widehat{\sigma}_s^l)$ and fine-tune them together with the affine terms (γ^l,β^l) on $\mathcal{D}_{\text{cal}}$, keeping other weights fixed.

By aligning BN statistics with the spike-domain distribution, we reduce the normalization distortion, which becomes increasingly severe at low T. Crucially, this recalibration is *operator-aware*: it is performed under the same parallel rule that governs inference, ensuring that the normalization is faithful to the actual execution regime. The procedure is lightweight—requiring only a small calibration set and no gradient updates—and forms the *Spatial Recalibration* stage of our two-part solution.

4.2 Motivation II: Temporal Error of Parallel Conversion

The QCFS-based parallel conversion rule enables constant-time execution by generating an entire spike train in one shot (5). While this guarantees that the *total* number of spikes matches the integer target k^l up to rounding, it does not control how these spikes are distributed across timesteps. Concretely, the firing condition at step t is

$$s^{l,t}=\mathbf{1}\big\{\,U^l+\psi^l\;\geq\;(T{-}t{+}1)\theta^l\,\big\}, \qquad U^l=\sum_{j=1}^{T}I^{l,j}, \tag{10}$$

which compares the cumulative input U^l against a *decreasing sequence of thresholds* $(T{-}t{+}1)\theta^l$. Once this inequality is satisfied, all subsequent timesteps will fire, so spikes concentrate systematically toward the end of the window.

To make this effect precise, we define the *temporal error* as the deviation from an ideal uniform allocation of k^l spikes across T steps: $\varepsilon_{\text{temp}}^{l,t}:=s^{l,t}-\frac{k^l}{T},\quad t=1,\dots,T$. This error satisfies the conservation law $\sum_{t=1}^{T}\varepsilon_{\text{temp}}^{l,t}=0$, since both the actual and uniform allocation yield k^l total spikes, but it exhibits a structured stepwise bias:

$$\varepsilon_{\text{temp}}^{l,t}=\begin{cases}-\frac{k^l}{T}, & t<T-k^l+1,\\ 1-\frac{k^l}{T}, & t\geq T-k^l+1.\end{cases} \tag{11}$$

Thus, spikes are consistently delayed toward later timesteps. While negligible for large T, this step-dependent bias becomes pronounced at ultra-low latency (small T), where it manifests as a systematic temporal mismatch even when the total spike count is correct.

4.3 Stepwise loss sensitivity under the parallel rule

The temporal skew in (11) can also be understood from a gradient perspective. Let $\sigma(\cdot)$ be a smooth surrogate of the Heaviside and

$$z_t \;:=\; v_{\text{pre}}^{l,t}-\theta^l \;=\; \frac{U^l+\psi^l}{T-t+1}-\theta^l, \qquad U^l \;:=\; \sum_{j=1}^{T}I^{l,j}. \tag{12}$$

Define the channel-wise spike estimate $\hat{s}^{l,t}=\sigma(z_t)$ and a loss $L(\{\hat{s}^{l,t}\}_{t=1}^{T},\,y)$. The *loss sensitivity to the pre-threshold potential* at step t is

$$g_t \;:=\; \frac{\partial L}{\partial v_{\text{pre}}^{l,t}} \;=\; \frac{\partial L}{\partial \hat{s}^{l,t}}\,\sigma'(z_t), \tag{13}$$

where the decision boundary is $z_t=0 \iff U^l+\psi^l=(T-t+1)\,\theta^l$. By the chain rule and the structure of z_t. Because $\frac{\partial z_t}{\partial U^l}=1/(T{-}t{+}1)$, the gradient aggregates as

$$\frac{\partial L}{\partial U^l}=\sum_{t=1}^{T}\frac{g_t}{T{-}t{+}1}. \tag{14}$$

Eq. (14) shows that each timestep contributes with a *harmonic weight* $(T-t+1)^{-1}$. Early steps are downweighted, and late steps are upweighted. In addition, $\sigma'(z_t)$ peaks near the boundary $z_t = 0$, which is reached later because $\Theta_t = (T-t+1)\theta^l$ decreases in t. Together, these effects induce a systematic *late-step sensitivity bias* that explains the temporal skew in spike allocation at small T.

4.4 Step-dependent correction of the parallel rule

To mitigate the late-step skew while preserving constant-time inference, we propose a correction strategy that modifies the bias term of the parallel rule (5). Concretely, we introduce a *temporal correction* $\delta^{l,t}$ that modifies the per-step bias term,

$$z_t(\delta) \;\coloneqq\; \frac{U^l + \psi^l}{T-t+1} - \theta^l \;+\; \delta^{l,t}, \qquad s^{l,t}(\delta) \approx \sigma\big(z_t(\delta)\big), \tag{15}$$

which is equivalent to replacing $b^{l,t}$ in (6) by $b^{l,t} \leftarrow b^{l,t} + \delta^{l,t}$, designed to redistribute spikes more evenly across timesteps on the constant-time mapping. We consider two variants of explicitly counteracting the time-varying boundary $\Theta_t = (T-t+1)\theta^l$ by injecting a step-dependent shift.

Stage 2: Ladder-Weighted First-Order Correction Field (LW-FOCF). One could minimize the the temporal error via a learnable profile that captures the late-step sensitivity using training dataset as the calibration set. Alternatively, we decide rather than introducing an unconstrained per-step bias with $O(TC_l)$ parameters, we restrict temporal corrections to the *ladder* direction $w_t \propto 1/(T-t+1)$, which coincides with the dominant mode of stepwise loss sensitivity (14). For each layer l with C_l channels we learn a per-channel coefficient $\boldsymbol{\delta}^l \in \mathbb{R}^{C_l}$ and define

$$\Delta^l(t,c) \;=\; w_t\, \delta^l_c, \qquad \mathbf{s}^l \;=\; \Theta\Big(\mathbf{\Lambda}^l_{\mathrm{pc}}\mathbf{I}^l + \mathbf{b}^l + \Delta^l - \theta^l\Big), \tag{16}$$

which preserves constant-time inference (only bias pre-addition) and introduces only $O(C_l)$ learnable parameters. This correction matches the temporal bias profile in (11) and aligns with the sensitivity weights $(T-t+1)^{-1}$.

Proposition 4.1 (Projected steepest descent in the ladder subspace)**.** *Let $g \in \mathbb{R}^T$ denote the stepwise gradient $g_t = \frac{\partial L}{\partial v^{l,t}_{\mathrm{pre}}}$ and let $\Delta \in \mathbb{R}^T$ satisfy $\|\Delta\|_2 \le \varepsilon$. The first-order loss decrease is $\delta\mathcal{L} \approx \langle g, \Delta\rangle$. The unconstrained steepest descent is $\Delta^\star \propto g$. If Δ is restricted to the ladder subspace $\mathcal{S} = \{\alpha\,\mathbf{w} : \alpha \in \mathbb{R}\}$ with $\mathbf{w} = (w_t)_t$, the optimal choice is the projection*

$$\Delta^\star_{\mathrm{lad}} = \varepsilon\, \frac{\langle g, \mathbf{w}\rangle}{\|\mathbf{w}\|_2}\, \frac{\mathbf{w}}{\|\mathbf{w}\|_2}, \tag{17}$$

i.e., the best first-order decrease achievable within $\mathcal{S}$.

Structured correction therefore aligns directly with the dominant temporal sensitivity mode while using only $O(C_l)$ parameters. Unlike per-step moment matching in OASSA, the LW-FOCF approach is both theoretically principled—via the projection interpretation in Proposition 4.1—and computationally lightweight, while still preserving constant-time inference. In both cases, the aggregation $\mathbf{\Lambda}^l_{\mathrm{pc}}$ and constant-time property remain unchanged.

5 Experiment

We validate the performance of our proposed method on the CIFAR-10/100 Krizhevsky et al. (2009) and ImageNet Deng et al. (2009) datasets, utilizing common VGG Simonyan & Zisserman (2014) and ResNet He et al. (2016) architectures. Our approach is benchmarked against a comprehensive set of state-of-the-art SNN training and conversion paradigms. These include direct training methods (e.g., TAB Jiang et al. (2024), TTS Guo et al. (2024)), hybrid training (e.g., LM-H Hao et al. (2023b)), conversion with subsequent rectification (e.g., SNM Wang et al. (2022), FTBC Wu et al. (2024)), and both standard (e.g., QCFS Bu et al. (2022), TPP Bojkovic et al. (2025)) and parallel ANN-to-SNN conversion (FS-PC Hao et al. (2025)). Detailed experimental configurations and hyperparameters are provided in the Appendix.

Table 1: Comparison of our method with other state-of-the-art approaches on CIFAR-10, CIFAR-100, and ImageNet. The symbol † indicates methods that adopt the error calibration defined in Eq. 4 from Hao et al. (2025).

Dataset	Method	Type	ANN Acc.(%)	Arch.	T	SNN Acc.(%)
CIFAR-10	QCFS Bu et al. (2022) ICLR	ANN-SNN Conversion	95.52	VGG-16	2, 4, 8	91.18, 93.96, 94.95
	FTBC Wu et al. (2024) ECCV	Conversion Rect.	95.92	VGG-16	2, 4	92.08, 94.67
	SNM Wang et al. (2022) ICLR	Conversion Rect.	94.09	VGG-16	32	93.43
	FS-PC Hao et al. (2025) ICML	Parallel Conversion	95.43	VGG-16	2	94.16
	Ours	**Parallel Conversion**	95.43	VGG-16	2	**94.32**
CIFAR-100	LM-H Hao et al. (2023b) ICLR	Hybrid Training	-	VGG-16	4	73.11
	QCFS Bu et al. (2022) ICLR	ANN-SNN Conversion	76.28	VGG-16	2, 4, 8	63.79, 69.62, 73.96
	TPP Bojkovic et al. (2025) ICML	ANN-SNN Conversion	76.21	VGG-16	4, 8	73.93, 76.03
	SNM Wang et al. (2022) ICLR	Conversion Rect.	74.13	VGG-16	32	71.80
	FTBC Wu et al. (2024) ECCV	Conversion Rect.	76.21	VGG-16	4, 8	71.47, 75.12
	FS-PC Hao et al. (2025) ICML	Parallel Conversion	76.11	VGG-16	2, 4	72.71, 75.98
	Ours	**Parallel Conversion**	76.11	VGG-16	2, 4	**74.03, 76.42**
ImageNet-1k	QCFS Bu et al. (2022) ICLR	ANN-SNN Conversion	74.29	VGG-16	8, 16, 32	19.12, 50.97, 68.47
	SNM Wang et al. (2022) ICLR	Conversion Rect.	73.18	VGG-16	32	64.78
	Burst Li & Zeng (2022)IJCAI	Conversion Rect.	74.27	VGG-16	32	70.61
	FTBC Wu et al. (2024) ECCV	Conversion Rect.	73.91	VGG-16	8, 16	69.31, 72.98
	FS-PC Hao et al. (2025) ICML	Parallel Conversion	74.23	VGG-16	2, 4	36.93, 71.23
	FS-PC† Hao et al. (2025) ICML	Parallel Conversion	74.23	VGG-16	2, 4	56.50, 71.75
	Ours	**Parallel Conversion**	74.23	VGG-16	2, 4	**63.91, 72.23**
	Ours†	**Parallel Conversion**	74.23	VGG-16	2, 4	**61.84, 72.65**
	Dspike Li et al. (2021) NeurIPS	Direct Training	-	ResNet-34	6	68.19
	RecDis Guo et al. (2022) CVPR	Direct Training	-	ResNet-34	6	67.33
	GLIF Yao et al. (2022) NeurIPS	Direct Training	-	ResNet-34	4	67.52
	TAB Jiang et al. (2024) ICLR	Direct Training	-	ResNet-34	4	67.78
	SEENN-I Li et al. (2023) NeurIPS	Direct Training	-	ResNet-34	3.38	64.66
	GAC-SNN Qiu et al. (2024) AAAI	Direct Training	-	ResNet-34	6	70.42
	TTS Guo et al. (2024) AAAI	Direct Training	-	ResNet-34	4	70.74
	QCFS Bu et al. (2022) ICLR	ANN-SNN Conversion	74.32	ResNet-34	8, 16, 32	35.06, 59.35, 69.37
	TPP Bojkovic et al. (2025) ICML	ANN-SNN Conversion	74.32	ResNet-34	8, 16	67.32, 72.03
	FTBC Wu et al. (2024) ECCV	Conversion Rect.	74.32	ResNet-34	8, 16	65.28, 71.66
	FS-PC Hao et al. (2025) ICML	Parallel Conversion	74.30	ResNet-34	2, 4	42.45, 67.28
	FS-PC† Hao et al. (2025) ICML	Parallel Conversion	74.30	ResNet-34	2, 4	65.20, 72.90
	Ours	**Parallel Conversion**	74.30	ResNet-34	2, 4	**68.41, 73.24**
	Ours†	**Parallel Conversion**	74.30	ResNet-34	2, 4	**69.27, 73.10**

5.1 Comparison with State-of-the-Art Methods

We first evaluate our method using pre-trained ANNs with the QCFS activation function, a common practice in recent conversion works designed to achieve high performance at low latency. The results are presented in Table 1.

CIFAR Datasets: On CIFAR-100 with a VGG-16 backbone, our method achieves 74.03% ($T = 2$) and 76.42% ($T = 4$) accuracy. This surpasses the prior parallel conversion baseline (FS-PC) and outperforms strong conversion rectification (FTBC) and direct conversion (TPP) methods at identical timesteps. We observe similar state-of-the-art performance on CIFAR-10.

ImageNet Dataset: The advantages of our method are most prominent on ImageNet. For ResNet-34 at $T = 4$, we achieve 73.24% accuracy, exceeding leading direct training methods like TTS (70.74%) and GAC-SNN (70.42%) without requiring complex temporal-domain optimization. With calibration (†), our model reaches 73.10% at $T = 4$, consistently outperforming the FS-PC baseline. Substantial gains at ultra-low latencies are also observed on the VGG-16 architecture.

5.2 Performance on Standard Models

To demonstrate the generalizability of our method beyond specialized activation functions, we evaluate its performance on standard ANNs pre-trained with the conventional ReLU activation. This is a more challenging and practical scenario due to the larger potential for quantization errors during conversion. As shown in Table 2, our method consistently and significantly outperforms the FS-PC baseline across various ResNet architectures. For instance, on ResNet-18 at $T = 8$, our method reduces the accuracy drop from the source ANN from 22.64% down to 5.07%. This highlights our method's robustness and its ability to mitigate conversion errors effectively, even without modifications to the source ANN's architecture or activation functions.

Table 2: Comparison of conversion algorithms on ReLU-based ResNet models (ImageNet-1k).

Arch.	ANN Acc.(%)	**FS-PC** Hao et al. (2025)		**Ours**	
		$T = 8$	$T = 16$	$T = 8$	$T = 16$
ResNet-18	69.76	55.18 (-14.58)	66.26 (-3.50)	**68.25** (-1.51)	**68.27** (-1.49)
ResNet-34	73.31	50.67 (-22.64)	68.04 (-5.27)	**68.23** (-5.07)	**71.66** (-1.64)
ResNet-50	76.12	64.16 (-11.96)	73.59 (-2.53)	**70.79** (-5.36)	**73.82** (-2.33)
ResNet-101	77.38	60.59 (-16.79)	73.86 (-3.52)	**67.33** (-10.04)	**73.96** (-3.41)

Table 3: Ablation study on ReLU-based ResNet models. We report SNN accuracy for the baseline (FS-PC) and after sequentially adding our spatial recalibration, temporal correction, and both components (Full). *Additional layer-wise visualizations (2nd, 8th, and 16th layers) are provided in Appendix B.1.*

Arch.	ANN Acc.(%)	T	Baseline(FS-PC)	Spatial	Temporal	Full
ResNet-18	69.76	4	25.20	61.83	55.55	62.28
ResNet-18	69.76	8	55.18	68.08	63.66	68.25
ResNet-34	73.31	8	50.67	68.08	56.14	68.23
ResNet-50	76.12	8	64.42	71.28	70.60	70.79

5.3 Ablation Study

To dissect the individual contributions of our proposed components—*spatial recalibration* and *temporal correction*—we conduct a detailed ablation study. We evaluate four configurations on ReLU-based ResNet models: (a) a baseline parallel conversion (FS-PC), (b) the baseline with only spatial recalibration, (c) the baseline with only temporal correction, and (d) full method combining both.

The results, summarized in Table 3, reveal that both components provide substantial and complementary improvements over the baseline. On ResNet-18 at $T = 8$, spatial recalibration alone boosts accuracy from 55.18% to 68.08%, while temporal correction increases it to 63.66%. Combining both yields the best performance of 68.25%, confirming their synergistic effect. Figure 3 provides a visual analysis, illustrating how spatial recalibration aligns the magnitude of the SNN's average firing rate with the ANN's activation value, while temporal correction refines the underlying spike timing patterns. Together, they enable a more faithful emulation of the original ANN's activations, leading to higher accuracy in the resulting SNN.

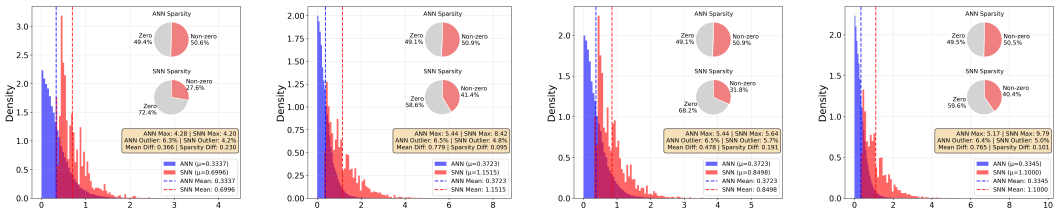

Figure 3: Visualization of the proposed two-stage calibration on the 8th ReLU / parallel IF layer of a ReLU-based ResNet-18 at $T = 8$.

6 Conclusion

This paper tackles the severe accuracy degradation in parallel ANN-to-SNN conversion at ultra-low timesteps ($T \leq 8$). We trace this issue to a spatiotemporal distributional shift, where the parallel firing mechanism induces both spatial mismatches in feature statistics and a systematic temporal bias. To resolve this, we introduced a two-stage calibration that recalibrates normalization layers for the spike domain and learns a temporal correction to offset firing-time errors, all while preserving constant-time inference. Our method dramatically boosts performance, raising ResNet-18 accuracy on ImageNet from $\mathbf{25.20}\%$ to $\mathbf{62.28}\%$ at $T = 4$. This work demonstrates that correcting the flawed statistical assumptions of the parallel framework is key to unlocking its potential for high-fidelity, low-latency inference.

Reproducibility Statement

We have made extensive efforts to ensure the reproducibility of our results. The main paper provides detailed descriptions of the proposed method and experiments. Additional implementation details, training configurations, and hyperparameters are documented in the appendix. Complete proofs of theoretical claims are also provided in the appendix. To facilitate replication, we will release all source code, scripts and logs in our experiments.

Ethics Statement

This work complies with the ethical standards set forth by the ICLR community. Our research does not involve human subjects, personally identifiable information, or sensitive data. All datasets used in our experiments are publicly available and widely adopted within the community (e.g., CIFAR, ImageNet), with appropriate licenses respected. We have taken care to report results transparently, ensure reproducibility through code and documentation, and minimize potential misuse by clarifying the intended scientific purpose. We do not anticipate direct societal risks beyond those commonly associated with advances in machine learning research.

References

Velibor Bojkovic, Xiaofeng Wu, and Bin Gu. Temporal misalignment in ANN-SNN conversion and its mitigation via probabilistic spiking neurons. *ICML*, 2025.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

Zhigao Cao, Meng Li, Xiashuang Wang, Haoyu Wang, Fan Wang, Youjun Li, and Zi-Gang Huang. Efficient training of spiking neural networks with multi-parallel implicit stream architecture. In *European Conference on Computer Vision*, pp. 422–438, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009.

Wei Fang, Zhaofei Yu, Zhaokun Zhou, Ding Chen, Yanqi Chen, Zhengyu Ma, Timothée Masquelier, and Yonghong Tian. Parallel spiking neurons with high efficiency and ability to learn long-term dependencies. *Advances in Neural Information Processing Systems*, 36:53674–53687, 2023.

Wanjin Feng, Xingyu Gao, Wenqian Du, Hailong Shi, Peilin Zhao, Pengcheng Wu, and Chunyan Miao. Efficient parallel training methods for spiking neural networks with constant time complexity. In *Forty-second International Conference on Machine Learning*, 2025.

Yufei Guo, Xinyi Tong, Yuanpei Chen, Liwen Zhang, Xiaode Liu, Zhe Ma, and Xuhui Huang. Recdis-snn: Rectifying membrane potential distribution for directly training spiking neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Yufei Guo, Yuhan Zhang, Yuanpei Chen, Weihang Peng, Xiaode Liu, Liwen Zhang, Xuhui Huang, and Zhe Ma. Membrane potential batch normalization for spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19420–19430, 2023.

Yufei Guo, Yuanpei Chen, Xiaode Liu, Weihang Peng, Yuhan Zhang, Xuhui Huang, and Zhe Ma. Ternary spike: Learning ternary spikes for spiking neural networks. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 2024.

Zecheng Hao, Jianhao Ding, Tong Bu, Tiejun Huang, and Zhaofei Yu. Bridging the gap between anns and snns by calibrating offset spikes. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023a.

Zecheng Hao, Xinyu Shi, Zihan Huang, Tong Bu, Zhaofei Yu, and Tiejun Huang. A progressive training framework for spiking neural networks with learnable multi-hierarchical model. In *The Twelfth International Conference on Learning Representations*, 2023b.

Zecheng Hao, Qichao Ma, Kang Chen, Yi Zhang, Zhaofei Yu, and Tiejun Huang. Faster and stronger: When ann-snn conversion meets parallel spiking calculation. In *Forty-second International Conference on Machine Learning*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Sun Jian. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952.

Di Hong and Yueming Wang. Self-attentive spatio-temporal calibration for precise intermediate layer matching in ann-to-snn distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

JiaKui Hu, Man Yao, Xuerui Qiu, Yuhong Chou, Yuxuan Cai, Ning Qiao, Yonghong Tian, Bo Xu, and Guoqi Li. High-performance temporal reversible spiking neural networks with $o(1)$ training memory and $o(1)$ inference cost. In *Forty-first International Conference on Machine Learning*, 2024.

Sangwoo Hwang and Jaeha Kung. One-spike snn: Single-spike phase coding with base manipulation for ann-to-snn conversion loss minimization. *IEEE Transactions on Emerging Topics in Computing*, 2024.

Haiyan Jiang, Vincent Zoonekynd, Giulia De Masi, Bin Gu, and Huan Xiong. Tab: Temporal accumulated batch normalization in spiking neural networks. 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yang Li and Yi Zeng. Efficient and accurate conversion of spiking neural network with burst spikes. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 2022.

Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. 2021.

Yuhang Li, Tamar Geller, Youngeun Kim, and Priyadarshini Panda. SEENN: towards temporal spiking early exit neural networks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Xuerui Qiu, Rui-Jie Zhu, Yuhong Chou, Zhaorui Wang, Liang-Jian Deng, and Guoqi Li. Gated attention coding for training high-performance and efficient spiking neural networks. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 2024.

Shuaijie Shen, Chao Wang, Renzhuo Huang, Yan Zhong, Qinghai Guo, Zhichao Lu, Jianguo Zhang, and Luziwei Leng. Spikingssms: Learning long sequences with sparse and parallel spiking state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Yuchen Wang, Malu Zhang, Yi Chen, and Hong Qu. Signed neuron with memory: Towards simple, accurate and high-efficient ann-snn conversion. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2501–2508. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/347. URL https://doi.org/10.24963/ijcai.2022/347. Main Track.

Xiaofeng Wu, Velibor Bojkovic, Bin Gu, Kun Suo, and Kai Zou. FTBC: forward temporal bias correction for optimizing ANN-SNN conversion. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXIX*, 2024.

Xingting Yao, Fanrong Li, Zitao Mo, and Jian Cheng. GLIF: A unified gated leaky integrate-and-fire neuron for spiking neural networks. 2022.

USE OF LARGE LANGUAGE MODELS (LLMS)

In preparing this work, we employed large language models (LLMs) as general-purpose assistive tools. Specifically, LLMs were used for: (i) grammar polishing and stylistic refinement of the manuscript text, (ii) assistance in reviewing and debugging baseline code implementations, and (iii) occasional support in summarizing related work for clarity. Importantly, no part of the research idea, methodology design, or experimental results relied on LLMs. The conceptual contributions, technical development, and validation of this work were conducted entirely by the authors. We take full responsibility for all content presented in this paper.

A APPENDIX

B EXPERIMENTAL DETAILS

For Table 1, we use VGG-16 and the widely adopted QCFS-pretrained ResNet-34. For Table 2, we adopt standard ResNet-18/34/50/101 models trained with ReLU activations.

In the QCFS-pretrained setting, activations produced by the quantization–clip–floor–shift (QCFS) operator are converted into parallel spiking neurons for constant-time inference. Entries marked with † employ the calibrated, channel-wise parallel integrate-and-fire variant; non-† models use the plain parallel neuron. For ReLU backbones, we follow this conversion flow: ReLU layers are first replaced by a recorder to capture activation ranges, converted once to QCFS with calibration, and finally substituted by parallel spiking neurons for inference.

Spiking activations. All constant-time SNNs use a *parallel integrate-and-fire* (ParaIF) mechanism that generates the entire T-step spike train in one pass without recurrent state. The basic **Parallel IF** aggregates the time-averaged input under a descending threshold ladder; our Stage-2 temporal correction adds a lightweight, per-channel bias to this potential. The **Calibrated, channel-wise Parallel IF** extends this with per-channel pre-threshold shifts and post-threshold amplitude adjustments, used for the error-calibrated (†) models. Both remain constant-time and introduce only 10^4-scale trainable parameters (approximately 7.6k for ResNet-34 and 12.4k for VGG-16). QCFS itself is used only for conversion and calibration, while a non-spiking recorder collects stable activation bounds before replacement.

Surrogate gradient design. Spiking nonlinearity is treated as a Heaviside step with a smooth surrogate derivative during fine-tuning. In all our experiments we select the triangle shape, which has unit slope within a small band around the threshold and zero outside.

Lightweight calibration. Stage 1 (spatial recalibration) updates only the affine parameters of normalization layers and refreshes running statistics using spiking activations. Stage 2 (temporal correction) optimizes the per-channel bias of the parallel spiking neuron to offset the late-step skew induced by the descending threshold ladder. Both stages use AdamW with a short warm-up and cosine decay; weight decay is omitted. Learning rates are 2×10^{-4} for ImageNet (one epoch for Stage 1, about three thousand steps for Stage 2) and $10^{-3} \rightarrow 10^{-4}$ for CIFAR in Stage 1 with 2×10^{-3} in Stage 2; gradient clipping is applied in Stage 2.

Timesteps and batch sizes. For Table 1, ImageNet experiments use VGG-16 with $T = 2$ and batch size 16, and ResNet-34 with $T = 2$ and batch size 16; CIFAR-10 uses VGG-16 with $T = 2$ and batch size 32; CIFAR-100 uses VGG-16 with $T = 4$ and batch size 64. For Table 2, ImageNet experiments use ResNet-18 with $T = 4$ (batch 16), ResNet-34 with $T = 16$ (batch 128), ResNet-50 with $T = 16$ (batch 32), and ResNet-101 with $T = 16$ (batch 16).

B.1 ABLATION STUDY

To provide a deeper insight into our ablation study, Figure 4 visualizes the impact of each proposed component on the output distributions at different depths of the network. We compare the source ANN's activation distributions (blue) with the converted SNN's average firing rates (red) at an early (2nd), middle (8th), and final (16th) layer of ResNet-18.

The baseline parallel conversion (a-c) exhibits a severe distribution mismatch, where the SNN's firing rates are significantly lower and sparser than the target ANN activations. Applying only Stage 1, Spatial Recalibration (d-f), almost completely resolves this magnitude discrepancy by aligning the means of the two distributions, demonstrating its role in correcting the output scale. While Stage 2, Temporal Correction (g-i), has a less pronounced effect on the static distribution by itself, combining it with Stage 1 (j-l) results in the most faithful emulation. Our full method consistently produces SNN firing rate distributions that closely match the ANN's statistics across all observed layers, effectively minimizing conversion error throughout the network.

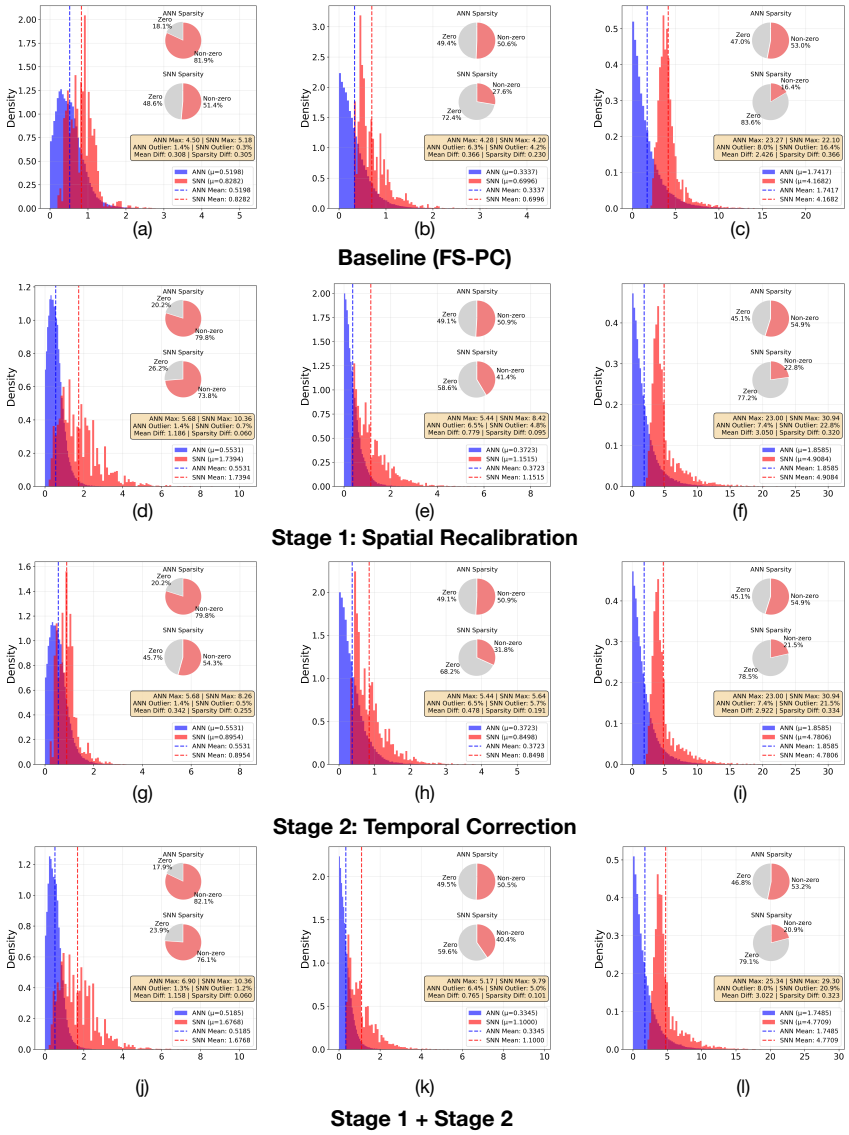

Figure 4: **Visualization of the proposed two-stage calibration on ReLU-based ResNet-18 at $T = 8$.** We compare the output distributions of three representative ReLU / *Parallel IF* layers (2nd, 8th, and 16th/final layer, from left to right) for four configurations: (a-c) Baseline (FS-PC), (d-f) Stage 1: Spatial Recalibration, (g-i) Stage 2: Temporal Correction, and (j-l) Stage 1 + Stage 2 (Our Full Method). Each plot shows the activation value distribution of the source ANN (blue) against the **average firing rate distribution** of the converted SNN (red), along with sparsity and key statistical metrics. Our full two-stage method (j-l) consistently achieves the closest match to the ANN distributions across all layers, minimizing divergence and balancing sparsity.