

C-STEP: Compute-Efficient Spiking Transformers with Temporal Exit and Early-Guided Pruning

Kyungchul Lee and Jongsun Park
School of Electrical Engineering, Korea University
Seoul, Republic of Korea
{12251kc, jongsun}@korea.ac.kr

Abstract—Spiking Transformers (STs) have emerged as an efficient alternative to artificial neural networks, yet they still incur substantial computation, motivating computational reduction techniques. In this paper, we present C-STEP, a unified technique for reducing computation in ST inference. First, we introduce LightSoftmax, a novel lightweight scoring technique that enables confidence-based temporal exit with negligible overhead, allowing inputs with confident predictions to terminate at early timesteps. Second, for the inputs that do not exit, we apply early-time-guided dynamic channel pruning to remove low-contribution channels in later timesteps. Third, we devise a synaptic computation scheme that decomposes spikes into a locally common component and token-specific residuals. The common component is computed once and reused across the tokens, preserving functional equivalence. We have designed an end-to-end SNN architecture that seamlessly executes the proposed low complexity schemes. C-STEP reduces synaptic operations by up to 65.4% relative to the original ST backbones.

Keywords—Spiking neural networks, Spiking transformers, Computation reduction technique, Accelerator

I. INTRODUCTION

Recent studies on spiking transformers (STs) report strong results in computer vision and natural language tasks [1], [2], [3], with accuracy comparable to artificial neural network (ANN) baselines. Despite these advantages, the computational cost of current STs remains a practical concern. Although spike-based inference removes many multiplications and benefits from sparsity through event-driven operation, STs still require substantial computation to reach ANN-level accuracy. As a result, practical efficiency is not yet fully realized, motivating methods that exploit spatio-temporal properties of STs to unlock their full potential as efficient alternatives to ANNs. In this paper, we propose C-STEP, a unified computation reduction technique for efficient ST acceleration. C-STEP integrates temporal early exit with early time guided channel pruning and is supported by a hardware-friendly execution strategy.

II. MOTIVATIONS

Our study begins with a simple question: **should every input traverse all timesteps?** We evaluate whether accurate inference is possible even when computation is restricted to early timesteps for STs. For an input image x , the number of classes M , and timestep t , we define a confidence score cs_t as following:

$$p_t = \text{softmax}(z_t(x)), \quad cs_t = p_{t, \text{argmax}(z_t(x))}, \quad (1)$$

where $z_t(x) \in \mathbb{R}^M$ denotes the output of ST model at timestep t , $p_t \in \mathbb{R}^M$ is the output probabilities, and the confidence score $cs_t \in \mathbb{R}$. Threshold τ_t is selected on a calibration set such that 50% of inputs satisfy $cs_t > \tau_t$ and would exit at time t . We then evaluate the classification accuracy on the skipped subset. As Fig. 1 (a) shows, the exited inputs at early timesteps

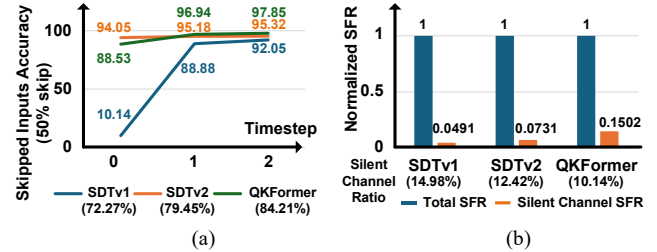


Fig. 1. (a) ImageNet-1K accuracy of the subset that exits at each timestep, with thresholds calibrated to achieve 50% exit. Numbers in parentheses indicate baseline accuracy with all four timesteps. (b) The normalized average spike-firing rates (SFRs) of silent channels. The percentages under each model name indicate the silent-channel ratio.

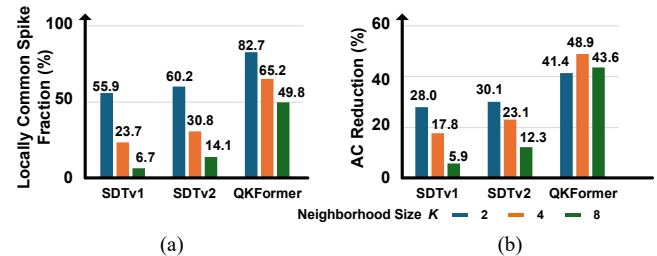


Fig. 2. (a) Locally common spike fraction for tokens in K neighborhood ($K=2,4,8$). (b) Estimated compute reduction when the locally shared component is computed once and reused across the K tokens.

already achieve high accuracy, up to 96.94% using only half of the total timesteps ($t = 1$).

Building on the evidence about the temporal early exit, we examine **how early time information can guide computation reduction on the non-exiting inputs**. For a given spike output $O \in \{0,1\}^{T \times N \times C}$, we define the early-window activity a_c and the silent channel mask scm_c as:

$$a_c = \sum_{t=0}^{t_0} \sum_{n=0}^{N-1} O_{t,n,c}, \quad scm_c = \begin{cases} 1, & a_c = 0 \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

here the early window uses the first two timesteps ($t_0=1$) and sums over all the tokens for each channel. The channel is labeled *silent* when no spike is observed in this window. Channel's spike firing rate (SFR) is measured over the remaining timesteps $t > t_0$. Fig. 1 (b) reports the normalized average SFR of total channels and the average SFR of *silent* channels. The channels that are *silent* remain largely inactive later, where the SFRs of these channels in the later window are about $6.7 \times -20 \times$ lower than the model-wide average.

When tokens within a local K neighbor share spike positions, **the shared contribution can be computed once and reused**. Fig. 2 (a) reports the fraction of spike positions that are common across all tokens within each K token neighborhood. With $K = 2$, up to 82.7% of positions are shared. Fig. 2 (b) illustrates the estimated reduction in accumulation operations when local common spike sharing is applied across tokens within K neighborhood. The results indicate substantial potential for computation reduction.

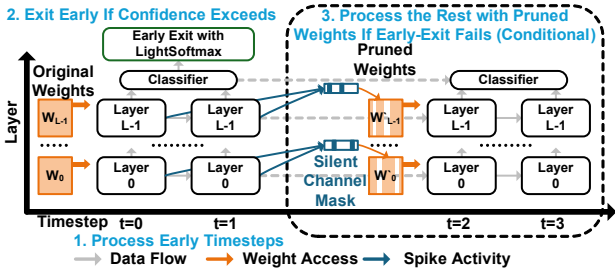


Fig. 3. Overview of C-STEP algorithm.

III. C-STEP ALGORITHM: TEMPORAL EXIT WITH LIGHTSOFTMAX AND EARLY-GUIDED PRUNING

Based on the observations from the previous section, we first present C-STEP algorithm. The overall flow is described in Fig. 3. First, process the first half timesteps ($t \leq 1$) and compute a confidence score with *LightSoftmax*. Second, if the confidence score exceeds the calibrated threshold τ_1 , the remaining timesteps are skipped. Third, for the inputs that do not exit, derive input-dependent silent-channel masks from early time spike activity using (2). Subsequent timesteps ($t > 1$) use pruned weights, reducing both weight accesses and synaptic operations.

To reduce the overhead of computing exponentiation operations for confidence score, the proposed *LightSoftmax* approximates non-predicted ST outputs as Gaussian random variables with mean μ and variance σ^2 . Using the expected value from the Gaussian moment generating function[4], the confidence score can be expressed as:

$$cs^t \approx \frac{e^{z_{t,pred}(x)}}{(M-1)e^{\mu + \frac{1}{2}\sigma^2} + e^{z_{t,pred}(x)}}. \quad (3)$$

Since the actual mean $\underline{\mu}$ and $e^{z_{t,pred}(x)}$ vary depending on the input, we add light corrections to maintain accuracy, by multiplying input-dependent factors to the offline constant:

$$(M-1)e^{\mu + \frac{1}{2}\sigma^2} = (M-1)e^{\underline{\mu} + \frac{1}{2}\sigma^2} \cdot \underbrace{e^{\underline{\mu} - \mu}}_{\text{online correction}}, \quad (4)$$

$$e^{z_{t,pred}(x)} = e^{z_{t,pred}(x)} \cdot \underbrace{e^{z_{t,pred}(x) - z_{t,pred}(x)}}_{\text{online correction}}, \quad (5)$$

here μ , σ^2 , and $z_{t,pred}(x)$ are preset on a calibration set, and at run time, only the input-dependent corrections with $\underline{\mu}$ and $z_{t,pred}(x)$ are computed.

IV. C-STEP ACCELERATOR: END-TO-END ST INFERENCE

C-STEP accelerator consists of global buffers, a scheduler, eight C-STEP processing elements (PEs), a special function unit (SFU), and a neuron unit. Fig. 4 illustrates the overall architecture. The weight buffer, spike buffer, word mask (WM) buffer, and output buffer store synaptic weights, nonzero spikes, word masks, and membrane potentials, respectively. The special function unit (SFU) computes the *LightSoftmax* confidence score, counts early window spikes to form silent channel masks, and provides MAC operations for spiking self-attention layer and classification layer. Synaptic operations are performed in the C-STEP PEs. We use eight PEs and assign each PE a pair of adjacent tokens within a single timestep. Two adjacent PEs are paired on the same token pair and assigned to the consecutive timesteps. Within each PE, the hardware supports local common spike sharing across the two adjacent tokens.

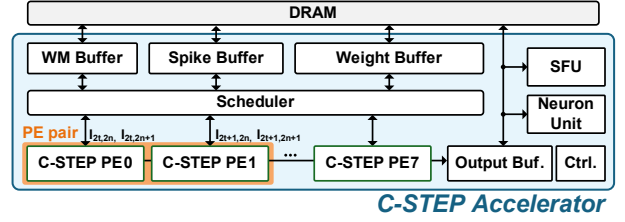


Fig. 4 C-STEP accelerator architecture.

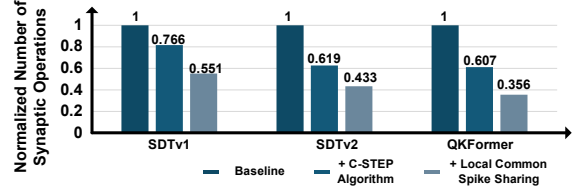


Fig. 5. Normalized synaptic operations per inference for SDTv1, SDTv2, and QKFormer. Baseline, C-STEP algorithm only, and C-STEP with local common spike sharing.

V. EVALUATION

To validate the effectiveness of our proposed C-STEP accelerator, we evaluate it on ImageNet-1K classification task [5] using publicly available pretrained ST backbone models: SDTv1 [3], SDTv2 [6], and QKFormer [7]. Models are 8-bit quantized with min-max symmetric linear quantization [8]. The confidence thresholds τ_1 and *LightSoftmax* parameters μ , σ^2 , and the preset max output $z_{t,pred}(x)$ are calibrated on 5K randomly sampled training images, with a target exit rate $r^* = 0.4$ and accuracy loss bound $\delta^* = 0.1$. Fig. 5 presents the normalized number of synaptic accumulation operations per inference relative to the original ST backbones. Applying the C-STEP algorithm, temporal early exit and silent channel pruning reduces operations to $0.607\times$ of baseline on QKFormer. With local common spike sharing additionally enabled in hardware, the total synaptic operations drop further to $0.356\times$ of baseline on QKFormer.

VI. CONCLUSION

In this paper, we present C-STEP, a computationally efficient approach that unifies an algorithmic method with a hardware accelerator. On the algorithmic side, we introduce *LightSoftmax* for negligible-overhead confidence scoring to enable temporal early exit, and we apply early time guided silent channel pruning. On the hardware side, we present an end-to-end accelerator that supports C-STEP algorithm and further reduces work through local common spike sharing. Across standard ST backbones, C-STEP reduces synaptic operations by as much as 64.4 percent. By identifying and exploiting early time confidence, persistent channel silence, and local token-level redundancy, C-STEP brings spiking transformers one step closer to being an efficient alternative to ANNs.

ACKNOWLEDGMENT

This work was supported in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00345481); in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00405495); in part by the Ministry of Trade, Industry and Energy(MOTIE) and Korea Institute for Advancement of Technology(KIAT) through the ‘‘International Cooperative R&D program’’ (Task No. P0028486)

REFERENCES

- [1] Z. Zhou *et al.*, “SPIKFORMER: WHEN SPIKING NEURAL NETWORK MEETS TRANSFORMER,” 2023.
- [2] C. Zhou *et al.*, “Spikingformer: Spike-driven Residual Learning for Transformer-based Spiking Neural Network,” May 19, 2023, *arXiv: arXiv:2304.11954*. doi: 10.48550/arXiv.2304.11954.
- [3] M. Yao *et al.*, “Spike-driven Transformer,” Jul. 04, 2023, *arXiv: arXiv:2307.01694*.
- [4] Bulmer, M. G, *Principles of statistics*.
- [5] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” Jan. 29, 2015, *arXiv: arXiv:1409.0575*
- [6] M. Yao *et al.*, “SPIKE-DRIVEN TRANSFORMER V2: META SPIKING NEURAL NETWORK ARCHITECTURE INSPIRING THE DESIGN OF NEXT-GENERATION NEUROMORPHIC CHIPS,” 2024.
- [7] C. Zhou *et al.*, “QKFormer: Hierarchical Spiking Transformer using Q-K Attention”.
- [8] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, “A Survey of Quantization Methods for Efficient Neural Network Inference,” Jun. 21, 2021, *arXiv: arXiv:2103.13630*.