BitShift-RoPE: Zero-FLOP Relative Positional Encoding for Spiking Neural Network Transformers

Seung-Kyu Hong* **Sangheum Hwang**

Hyuk-Yoon Kwon†
Seoul National University of Science and Technology
Seoul, Republic of Korea
{hjh9681, shwang, hyukyoon.kwon}@seoultech.ac.kr

Abstract

Spiking Neural Networks (SNNs) provide energy-efficient computation by using binary, spike-driven additions, but applying transformer architectures remains challenging because SNNs lack an SNN-compatible positional encoding (PE). Current PE methods either rely on floating-point arithmetic, which destroys the fundamental binary nature of SNNs, or fail to accurately capture relative spatial distances. To overcome this, we introduce Bit Shift Rotary Position Embedding (BitShift-RoPE), a relative PE framework that preserves the binary integrity of SNN queries and keys. By replacing the floating-point trigonometric rotations in standard RoPE with discrete cyclic-shift operations via hardware memory-pointer routing, our method achieves zero-FLOP position encoding. We analytically map multi-scale exponential-decay frequencies to integer shift steps and theoretically extend the formulation to 2D orthogonal spaces to capture spatial structure. Crucially, because the binary integrity of the queries and keys is preserved, the subsequent attention calculations rely solely on sparse bitwise AND operations. Our evaluations across time-series forecasting (R^2 0.758), text classification (Acc 69.91%), and image classification benchmarks (Acc 84.17%) demonstrate that BitShift-RoPE is consistently at least as good as existing SNN positional encodings while consuming the same energy as the vanilla SNN backbone (0.216 mJ/sample, identical to vanilla Spikformer). The code is available at `https://github.com/bigbases/Bit_Shift_RoPE`.

1 Introduction

Spiking Neural Networks (SNNs) are brain-inspired networks that transmit information through binary events (spikes) of value 0 or 1, replacing the floating-point operations (FLOPs) of conventional Artificial Neural Networks (ANNs) with ultra-low-power conditional synaptic operations (SOPs). Early SNN work was convolutional [6]; recent work uses Transformer architectures [35, 32, 34]. As Transformer-based models become dominant, designing a positional encoding (PE) that resolves the position-invariance of self-attention has emerged as a core problem for SNNs.

Positional encodings are broadly classified as either absolute or relative, and given the spike-generating Leaky Integrate-and-Fire (LIF) neuron at the core of SNN computation, they are applied either *pre-spike* (before LIF) or *post-spike* (after LIF). However, pre-spike application incurs substantial extra FLOPs to apply the encoding, undermining SNN's energy advantage [27, 32], and the

*First author.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

LIF non-linearity breaks the distance-preservation identity of self-attention [20] (formal statement in Section 4).

To overcome this limitation, recent SNN PE methods adopt the post-spike strategy [19, 20], injecting positional information after spike conversion. Yet representative methods such as Gray-PE and Log-PE [20] still require floating-point operations during PE computation and yield multi-bit integer outputs, sacrificing SNN's signature bitwise-AND efficiency.

In this work, we claim **three constraints** that positional encodings for SNN Transformers must satisfy to ensure operational efficiency and mathematical integrity: **C1)** No additional floating-point operations are introduced during positional encoding; **C2)** The output of the positional encoding remains binary, so that attention does not require a multiplier; **C3)** Relative distance in the attention map is expressible as an exact mathematical formula.

Rotary Position Embedding (RoPE) [30], the de facto standard for ANN Transformers, uses a trigonometric rotation matrix R to yield the relative-distance form $QR_{m-n}K^T$. In an SNN, however, pre-spike RoPE collapses through the LIF non-linearity and post-spike RoPE converts spikes back to floating-point reals, so both variants violate the constraints above.

In response, we propose Bit Shift Rotary Position Embedding (simply, BitShift-RoPE), the first SNN positional encoding that replaces the trigonometric, floating-point rotation with a discrete-space permutation. Our method is built on a cyclic shift over the indices of a 0/1 spike array. BitShift-RoPE adapts post-spike to avert the relative-distance collapse. The full feature dimension is partitioned into groups, and RoPE's multi-frequency structure is converted into a per-group fixed shift count k_g. Exploiting the orthogonality of the shift matrix S (i.e., $(S^n)^T = S^{-n}$), the attention dot product produces the scaled relative distance $S^{k_g(m-n)}$ *without any additional arithmetic*. To the best of our knowledge, BitShift-RoPE is the first SNN positional encoding that introduces no floating-point arithmetic on the encoding path (C1), keeps queries and keys binary (C2), and realizes relative distance with a rounding error bounded by one shift step per group independently of absolute position (C3). Section 2.3 provides a detailed comparison with existing methods.

The main contributions of this paper are as follows:

1. **Mathematical preservation of SNN integrity.** BitShift-RoPE injects relative positional information into SNN Transformers without passing through the LIF non-linearity or converting spikes into integer tensors, preserving pure spike binarity throughout positional encoding and attention.
2. **Hardware efficiency in attention.** Floating-point operations are eliminated from the positional-encoding stage, and the attention dot product reduces to ultra-sparse bit-level AND, preserving SNN's low-power advantage.
3. **Multi-scale frequency and state-of-the-art performance.** An analytical mapping projects RoPE's exponential-decay frequencies into discrete integer shift steps, extended to 2D for spatial context. We set state-of-the-art benchmarks across modalities: average R^2 of 0.758 in time-series forecasting, 69.91% accuracy in text classification, and 84.17% accuracy in image classification. A dense prediction task (Oxford-IIIT Pet segmentation) further confirms that the gains stem from the encoded positional content rather than from efficiency alone.

2 Related Work

2.1 SNN Transformers

Spikformer [35] introduced spike-based self-attention by combining LIF neurons with a vanilla Transformer. The Spike-Driven Transformer [32] then exploited spike sparsity through pure-spike matrix multiplication, Spikingformer [33] addressed vanishing gradients via spike-based residual learning, and QKFormer [34] introduced a multi-stage attention that mitigates the information loss of query/key binarization.

None of these architectures applies explicit positional encoding through attention. They instead rely on convolutional layers to learn position implicitly, which limits their ability to capture long-range

dependencies (Section 5.1) and underscores the difficulty of injecting position without distorting spike binarity.

2.2 SNN Positional Encodings

To overcome the limitations of implicit, convolution-based positional learning, several positional encoding methods that respect SNN's binary nature have been proposed. CPG-PE [19] is an absolute PE that converts thresholded sine waves into 0/1 patterns via a Heaviside step and concatenates them onto the spike sequence. Because it is keyed on absolute coordinates, aliasing arises on long sequences from the periodicity of the patterns.

To overcome the limitations of absolute PE, Lv et al. [20] proposed two relative-PE variants. Gray-PE concatenates Gray-coded position indices to the spike vector but inflates the spike dimension $D \to D + k$ and yields a Hamming distance that does not preserve linear relative distance. Log-PE adds log-bucketed distance biases directly to the attention map. The authors themselves describe both as *principled approximations constrained by binary spike dynamics* [20]. Their outputs are multi-bit integer tensors rather than pure binary spikes, so attention can no longer use SNN's signature bitwise-AND operation.

2.3 Positioning of BitShift-RoPE

The methods reviewed above leave three core gaps regarding the constraints stated in Section 1. (1) Pre-spike methods [35, 32, 34] break the relative-distance algebra of attention through the LIF non-linearity, violating **C3**. (2) The post-spike absolute PE, CPG-PE [19], preserves binarity but suffers aliasing on long sequences and cannot express relative distance natively, also violating **C3**. (3) The most recent post-spike relative PE attempts, Gray-PE and Log-PE [20], require floating-point arithmetic during encoding (violating **C1**) and yield multi-bit integer outputs, so attention can no longer be computed as a sparse bitwise AND (violating **C2**). The cost of the resulting XNOR-based similarity lies not in the gate itself, which is as cheap as AND, but in sparsity. $\mathrm{XNOR}(q, k) = 1$ whenever the operands agree, including 0–0, so on sparse spike tensors nearly every term is non-zero and the accumulation that follows becomes dense.

BitShift-RoPE addresses all three gaps. It adds no FLOPs on the positional-encoding path (satisfying **C1**). It preserves pure binarity in queries and keys, so attention is a sparse bitwise AND (satisfying **C2**). Its attention map depends on relative displacement up to a rounding term bounded by one shift step per group, independently of absolute position (satisfying **C3**, Proposition 1 and Corollary 2, Section 4).

Distinction from prior post-spike PEs. Unlike CPG-PE [19], our method directly encodes *relative* position, so it does not depend on absolute coordinates: Corollary 2 shows that the per-group rounding error is bounded independently of the absolute position m and therefore does not accumulate with sequence length. The joint shift code is collision-free only within its period, so the alias-free span must be checked against the sequence length or patch grid in use (Section 6). Unlike Gray-PE [20], our method does not inflate the spike dimension D (Gray-PE expands $D \to D+k$), and its output remains purely binary, so attention stays a sparse bitwise AND rather than a dense XNOR accumulation. Unlike Log-PE [20], our method does not add integer or real-valued biases to the attention map; it injects positional information solely by permuting the query and key tensors. Quantitative support appears in Table 1 (complexity) and Table 5 (energy).

Distinction from continuous-space RoPE. Standard RoPE [30] also provides relative positional encoding, but its $\sin$ / $\cos$-based rotation matrix R converts the data to floating-point tensors, violating **C2** (Section 3.3). BitShift-RoPE replaces R with a discrete shift matrix S, a RoPE-inspired discretization in which an m-step cyclic shift acts as a phase rotation at uniformly spaced DFT modes (Proposition 1, Appendix A.1). The correspondence is structural rather than pointwise: RoPE's exponential frequency schedule and continuous rotation are not preserved, while the shift-difference identity and binarity are.

3 Preliminary

3.1 Leaky Integrate-and-Fire (LIF) Neuron

SNNs use the Leaky Integrate-and-Fire (LIF) neuron as their fundamental unit. At a discrete time step t, the membrane-potential $U[t]$ update and spike-firing process $S[t]$ of an LIF neuron are defined as:

$$U[t] = \tau U[t-1](1 - S[t-1]) + X[t]; S[t] = \mathcal{H}(U[t] - V_{th}), \tag{1}$$

where $X[t]$ is the input synaptic current, τ is the leak constant, V_{th} is the firing threshold, and $\mathcal{H}(\cdot)$ is the Heaviside step function (returning 1 when its input is non-negative, 0 otherwise). The LIF neuron thus converts a real-valued input into a binary spike $S[t] \in \{0, 1\}$.

3.2 Spiking Self-Attention (SSA)

Spiking Self-Attention (SSA) SSA operates on binary spike tensors $S_q, S_k \in \{0, 1\}^{N \times D}$ produced by LIF neurons:

$$SSA = S_q \cdot S_k^T. \tag{2}$$

Since both operands are 0 or 1, the dot product reduces to a bitwise AND that counts the indices where both vectors have a 1. The binary sparsity of S_q and S_k must be preserved throughout to satisfy **C2**.

3.3 Rotary Position Embedding

Rotary Position Embedding (RoPE) [30] encodes the absolute positions m, n of tokens in a sequence through a rotation matrix R based on trigonometric functions. The rotation matrix R_m for position m at feature dimension D is defined as a block-diagonal matrix of 2×2 real-valued rotation blocks, formed by pairing feature dimensions in twos:

$$R_m = \text{diag}(r_1, r_2, \ldots, r_{D/2}) \in \mathbb{R}^{D \times D}, \quad \text{where } r_i = \begin{pmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{pmatrix}. \tag{3}$$

In the ANN setting, applying RoPE to real-valued vectors X_q, X_k yields the attention dot product:

$$(X_q R_m) \cdot (X_k R_n)^T = X_q R_m R_n^T X_k^T = X_q R_{m-n} X_k^T. \tag{4}$$

Each block i is assigned a distinct fundamental frequency $\theta_i = 10000^{-2i/D}$, mapping higher frequencies to lower indices and yielding multi-scale positional awareness. However, applying R to binary spikes converts the result into a floating-point tensor, violating **C2**.

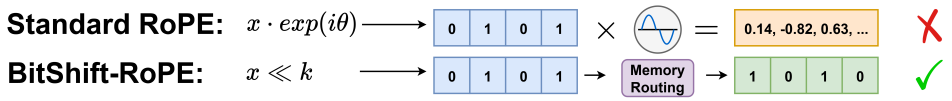
4 Proposed Method: BitShift-RoPE

4.1 Overview

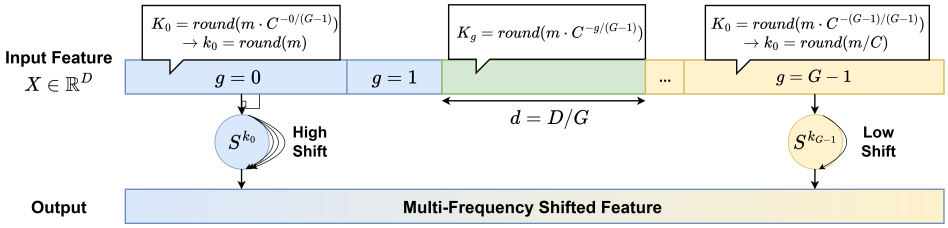
We propose BitShift-RoPE, a zero-FLOP relative positional encoding framework that preserves the pure binary integrity of SNNs. As shown in Figure 1a, our method replaces the trigonometric rotations of standard RoPE with discrete cyclic-shift operations executed via memory pointer routing, thereby preserving the binary property of spike tensors. To capture multi-scale positional contexts, Figure 1b illustrates our multi-frequency shift mapping, which partitions the feature dimension into discrete groups and applies a specific integer shift step derived from an exponential decay formula to each group. Finally, Figure 1c depicts the 2D extension designed for vision tasks, which orthogonally decomposes the channel groups into X-axis and Y-axis halves to independently shift and encode 2D spatial geometries.

Concretely, BitShift-RoPE uses a cyclic shift matrix S satisfying:

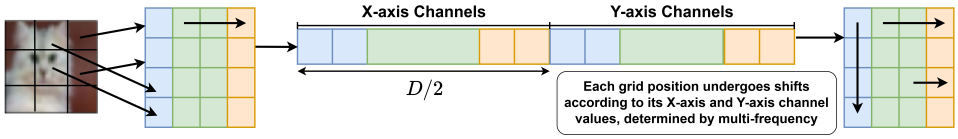
$$S^T = S^{-1}. \tag{5}$$



(a) Standard RoPE vs. BitShift-RoPE.



(b) Multi-frequency shift mapping.



(c) BitShift-RoPE 2D extension.

Figure 1: Overview of BitShift-RoPE. (a) The rotation matrix of standard RoPE is replaced with a discrete cyclic shift matrix S, applied via memory-pointer routing. (b) Multi-frequency context is captured by partitioning the feature dimension D into G groups, each assigned a distinct integer shift count k_g. (c) The 2D extension splits the G groups into halves along the X- and Y-axes for image patches.

Since S is orthogonal ($(S^n)^T = S^{-n}$), we project RoPE's relative form R_{m-n} onto the discrete shift S^{m-n} by manipulating only the spike-array index order. Applying S to the post-spike query $Q \in \{0,1\}^D$ and key $K \in \{0,1\}^D$, the attention dot product between tokens at positions m and n unfolds as:

$$\begin{aligned} Attn(m,n) &= (QS^m) \cdot (KS^n)^T \\ &= QS^m (S^n)^T K^T \\ &= QS^m S^{-n} K^T \\ &= QS^{m-n} K^T. \end{aligned} \tag{6}$$

As in Eq. 6, the orthogonality of the shift matrix replaces the absolute positions m, n with the relative distance $(m-n)$ during the attention computation. The following proposition, a textbook property of circulant matrices, relates the cyclic shift to a phase rotation in the frequency domain.

Proposition 1 (DFT Diagonalization of the Cyclic Shift)**.** *The m-fold application S^m of the discrete cyclic shift matrix S of size D is diagonalized by the Discrete Fourier Transform (DFT) matrix F, and acts as a complex phase rotation in the frequency domain at uniformly spaced modes θ_r:*

$$S^m = F^{-1} \Lambda^m F, \quad \textit{where } \Lambda^m_{r,r} = e^{-im\theta_r} \left(\theta_r = \frac{2\pi r}{D},\ r \in \{0, 1, \ldots, D-1\} \right). \tag{7}$$

By Proposition 1, an m-step cyclic shift in the spatial domain corresponds, in the frequency domain, to a multi-frequency phase rotation of the same form $e^{-im\theta_r}$ as continuous-space RoPE. The correspondence is *structural*, not pointwise. S^m activates uniformly spaced modes $\theta_r = 2\pi r/D$, whereas RoPE uses an exponential-decay schedule $\theta_i = 10000^{-2i/D}$, and the continuous rotation and continuous rescaling of RoPE are not reproduced. What is preserved exactly, independently of the frequency schedule, is the shift-difference identity $QS^a(KS^b)^T = QS^{a-b}K^T$ together with binarity.

Table 1: Computational complexity comparison across positional encoding methods, separated by operator type. N: sequence length, D: feature dimension, K: CPG convolution kernel size, b: Gray-code width. Sparse/dense refers to the density of non-zero terms entering the accumulation.

Method	PE Application (runtime)	Attention Cost ($Q \cdot K^T$)	Positional Arithmetic
ANN RoPE	$\mathcal{O}(N \cdot D)$ FP mul	$\mathcal{O}(N^2 \cdot D)$ FP mul + add	–
CPG-PE	$\mathcal{O}(N \cdot K \cdot D)$ SOPs/FLOPs	$\mathcal{O}((N \cdot K)^2 \cdot D)$ AND + accumulate (sparse)	–
Gray-PE	dimension expansion ($D \to D + b$)	$\mathcal{O}(N^2 \cdot (D + b))$ XNOR + accumulate (dense)	$\mathcal{O}(N^2)$ INT add
Log-PE	LUT memory access	$\mathcal{O}(N^2 \cdot D)$ XNOR + accumulate (dense)	$\mathcal{O}(N^2)$ INT add + $\mathcal{O}(N^2)$ FP add
BitShift-RoPE (Ours)	0 **FLOPs**†, $\mathcal{O}(N \cdot D)$ **gather**	$\mathcal{O}(N^2 \cdot D)$ **AND + accumulate (sparse)**	**none**

†The index table is built once at initialization with $\mathcal{O}(N \cdot D)$ floating-point operations and cached; the runtime path is a single indexed gather.

The per-group shift count k_g in Section 4.2 (Eq. (8)) is therefore a RoPE-inspired discretization of the operator family rather than a reproduction of its frequency schedule (proofs in Appendices A.1 and A.2).

4.2 Multi-Frequency Shift Mapping

A naive cyclic shift aliases once the period is exceeded. To avoid this without learnable parameters, we map RoPE's exponential-decay frequency formula directly to integer shift steps.

When the full feature dimension D is partitioned into G groups, the per-group shift count $k_g(m)$ for position m at group $g \in \{0, 1, \ldots, G-1\}$ is defined using a decay constant C as follows:

$$k_g(m) = \text{round}\left(m \cdot C^{-\frac{g}{G-1}}\right). \tag{8}$$

The first group ($g = 0$) shifts by m to capture fine-grained ordering, while the last group ($g = G-1$) shifts slowly to capture global context. Applying these per-group frequencies to Eq. 6 yields the attention dot product $QS^{k_g(m)-k_g(n)}K^T$ in each channel group, producing a multi-scale relative distance.

The rounding operation introduces a per-group phase quantization error whose bound is constant in m (Corollary 2), so the error does not grow with absolute position even when the sequence length exceeds the training distribution (full proof in Appendix A.3).

4.3 2D Extension

For vision patches at 2D coordinates (m_x, m_y), we split the G channel groups into halves and assign one half to X-axis shifts and the other to Y-axis shifts ($Q = [Q_x \| Q_y]$). The per-axis shift counts $k_{x,g'}(m_x)$ and $k_{y,g'}(m_y)$ for axis subgroup index $g' \in \{0, 1, \ldots, G/2-1\}$ are:

$$k_{x,g'}(m_x) = \text{round}\left(m_x \cdot C^{-\frac{g'}{G/2-1}}\right), \quad k_{y,g'}(m_y) = \text{round}\left(m_y \cdot C^{-\frac{g'}{G/2-1}}\right). \tag{9}$$

Attention between 2D patches $\mathbf{m}$ and $\mathbf{n}$ decomposes orthogonally:

$$SSA(\mathbf{m}, \mathbf{n}) = Q_x S^{k_{x,g'}(m_x)-k_{x,g'}(n_x)} K_x^T + Q_y S^{k_{y,g'}(m_y)-k_{y,g'}(n_y)} K_y^T. \tag{10}$$

4.4 Zero-FLOPs Implementation via Memory Pointer Routing

BitShift-RoPE implements the shift operation through memory routing, manipulating only the start-address offset when reading spike data from hardware memory rather than using any arithmetic unit during positional encoding.

The shift-count computation in Eq. 8 and the resulting routing indices are precomputed once per sequence and cached, using no floating-point arithmetic. The permuted Q and K thus retain pure 0/1 binarity, so attention reduces to multiplier-free bitwise AND (full implementation in Appendix E). Because S is a permutation matrix, surrogate-gradient training of the upstream LIF non-linearity flows through S unchanged. The Jacobian of the permutation is the identity on the support, so no special handling is required during backpropagation.

4.5 Computational Complexity Analysis

Table 1 summarizes the complexity of each PE with sequence length N and feature dimension D, separated by operator type. ANN RoPE is bottlenecked by floating-point multiplications. CPG-PE adds convolutions, and Gray-PE/Log-PE either expand the dimension or require LUT accesses. More critically, the latter two output multi-bit integers, so they compute similarity by XNOR rather than AND. The cost of this change lies not in the gate itself, which is as cheap as AND under the Horowitz model, but in density. $\mathrm{XNOR}(q, k) = 1$ whenever the operands agree, including 0–0, so on sparse spike tensors nearly every term is non-zero, and the subsequent $\mathcal{O}(N^2 \cdot D)$ accumulation becomes dense. Gray-PE further pays for its augmented $(D + b)$-dimensional similarity, and Log-PE for $\mathcal{O}(N^2)$ floating-point bias additions on the attention map.

In contrast, BitShift-RoPE introduces no positional arithmetic on the runtime path. Its only runtime cost is an $\mathcal{O}(N \cdot D)$ indexed gather over a cached table, which is $\mathcal{O}(N)$ cheaper than the $\mathcal{O}(N^2 \cdot D)$ attention it feeds, and by preserving the binary property of Q and K it executes attention with only sparse AND + accumulate (SOPs) (detailed breakdown in Appendix F).

5 Experiments

We compare benchmark accuracy and runtime efficiency across three domains to demonstrate the effectiveness of BitShift-RoPE. For time-series forecasting and image classification, we use Spikformer [35], Spikingformer [33], and QKFormer [34] as backbones. For text classification, we use SpikeBERT [18], which builds on Spikformer-style SSA. The comparison PE methods are convolutional Conv-PE [35], CPG-PE [19], and Gray-PE/Log-PE [20]. The ablation analyzes how the two hyperparameters of BitShift-RoPE, namely the number of groups G over the feature dimension D and the decay constant C, affect performance. Detailed experimental configurations, including common hyperparameters ($T = 4$, $G = 4$, $C = 64.0$, and LIF thresholds V_{th}), are provided in Appendix B. Throughout Section 5 and the appendices, all reported numbers are averaged across 3 random seeds.

5.1 Time-Series Forecasting

Table 2 reports results on four benchmarks: Metr-LA, PEMS-BAY [17], Solar, and Electricity [14], evaluated at prediction lengths $\{6, 24, 48, 96\}$. BitShift-RoPE records the highest average R^2 on every SNN backbone evaluated. Attention over correlated temporal signals is also central to ANN forecasting and anomaly detection on multi-source energy data [24, 25, 12]; here we study its spike-domain counterpart, where position must be injected without floating-point arithmetic.

Comparison with absolute and relative PE methods. On the QKFormer backbone, BitShift-RoPE records an average R^2 of 0.748 against 0.708 for the absolute PE CPG-PE and 0.729 and 0.733 for the relative PEs Gray-PE and Log-PE. The margins are small relative to seed variation, but their direction is consistent. Against the strongest baseline on each backbone, the paired improvement over the 16 dataset $\times$ horizon conditions is $+0.0075$ on Spikformer ($p = 0.025$), $+0.0146$ on Spikingformer ($p = 0.042$), and $+0.0148$ on QKFormer ($p = 0.002$). At the cell level, 44 of 48 conditions favor BitShift-RoPE over Log-PE.

Long-horizon dependency capture and accuracy retention. At the longest horizon (96), BitShift-RoPE records R^2 of 0.314 (Spikformer) and 0.307 (QKFormer) on Metr-LA, the highest among SNN backbones on that cell. This shows that BitShift-RoPE preserves global context and defends against accuracy loss at long forecast horizons.

5.2 Text Classification

To evaluate effectiveness on sentence-classification tasks, we run experiments on four standard benchmarks: MR and Subj [23, 22], and SST-2 / SST-5 [29]. The backbone is SpikeBERT [18], and we report the floating-point reference from fine-tuned BERT [5]. Beyond vision, SNNs have been applied to language tasks with adaptive spike generation [7]; BitShift-RoPE is complementary, encoding token position without altering spike dynamics. Robust ANN text classification under distribution shift has likewise been studied [10, 11], whereas we target the energy-efficient spiking regime.

Table 2: Performance comparison for time series forecasting on 4 benchmarks with prediction lengths of 6, 24, 48, and 96. The best results are shown in **bold**. PE types: A = absolute, R = relative. Metrics: higher R^2 and lower RSE indicate better performance.

Models	PE Type	Metric	Metr-la (L = 12)				Pems-bay (L = 12)				Solar (L = 168)				Electricity (L = 168)				Avg.
			6	24	48	96	6	24	48	96	6	24	48	96	6	24	48	96	
Transformer w/RoPE (floating-point reference)	R	R^2 ↑	.729	.560	.416	.306	.787	.730	.694	.676	.951	.854	.763	.720	.984	.978	.974	.968	.756
		RSE↓	.548	.696	.802	.878	.499	.563	.600	.617	.225	.373	.492	.539	.251	.274	.341	.420	.507
Transformer w/Sin-PE	A	R^2 ↑	.727	.554	.413	.284	.785	.734	.688	.673	.953	.858	.759	.718	.978	.975	.972	.964	.733
		RSE↓	.551	.704	.808	.895	.502	.558	.610	.618	.223	.377	.504	.545	.260	.277	.347	.425	.512
Spikformer w/Conv-PE	A	R^2 ↑	.713	.527	.399	.267	.773	.697	.686	.667	.929	.828	.744	.674	.959	.955	.955	.954	.733
		RSE↓	.565	.725	.818	.903	.514	.594	.606	.621	.272	.426	.519	.586	.373	.371	.379	.382	.541
Spikformer w/RoPE	A	R^2 ↑	.699	.493	.390	.243	.768	.699	.680	.664	.911	.820	.714	.644	.954	.951	.949	.940	.720
		RSE↓	.584	.757	.835	.920	.519	.591	.614	.625	.294	.441	.550	.633	.375	.383	.384	.454	.559
Spikformer w/CPG-PE	A	R^2 ↑	.726	.526	.418	.287	.780	.712	.690	.666	.937	.833	.757	.707	.972	.970	.966	.960	.744
		RSE↓	.553	.720	.806	.890	.508	.580	.602	.622	.257	.420	.506	.555	.299	.310	.314	.355	.519
Spikformer w/Gray-PE	R	R^2 ↑	.728	.544	.405	.269	.782	.724	.694	.673	**.936**	.840	.756	.710	.974	.972	.966	.962	.748
		RSE↓	.546	.706	.813	.910	.506	.578	.597	.618	**.257**	.409	.507	.546	.276	.304	.320	.342	.513
Spikformer w/Log-PE	R	R^2 ↑	.735	.535	.424	.290	.789	.717	.691	.670	.933	.841	**.758**	**.734**	.978	.974	.968	.964	.750
		RSE↓	.543	.719	.799	.876	.496	.575	.601	.620	.265	.408	**.504**	**.525**	.272	.300	.314	.340	.509
Spikformer w/BitShift-RoPE (Ours)	R	R^2 ↑	**.735**	**.561**	**.431**	**.314**	**.788**	**.729**	**.699**	**.689**	**.936**	**.865**	**.758**	.712	**.979**	**.976**	**.976**	**.973**	**.758**
		RSE↓	**.543**	**.698**	**.795**	**.873**	**.499**	**.563**	**.594**	**.603**	.260	**.377**	**.504**	.551	**.256**	**.277**	**.278**	**.293**	**.498**
Spikingformer w/Conv-PE	A	R^2 ↑	.657	.369	.213	.081	.794	.697	.643	.560	.903	.728	.533	.495	.958	.956	.949	.923	.654
		RSE↓	.618	.838	.936	1.01	.490	.594	.657	.735	.311	.522	.685	.720	.365	.374	.405	.495	.610
Spikingformer w/CPG-PE	A	R^2 ↑	.691	.414	.269	.133	.797	.667	.658	.562	.900	.739	.664	**.615**	.962	.958	.949	.935	.682
		RSE↓	.746	.812	.910	.985	.486	.622	.633	.731	.315	.523	.585	**.623**	.361	.366	.405	.477	.599
Spikingformer w/Gray-PE	R	R^2 ↑	.714	.430	.312	.181	.805	**.701**	.680	**.657**	.905	.734	.555	.499	.964	.957	.951	.949	.687
		RSE↓	.564	.797	.874	.954	.478	**.582**	**.611**	**.633**	.309	.516	.668	.709	.358	.369	.397	.406	.577
Spikingformer w/Log-PE	R	R^2 ↑	.720	.390	.294	.199	.803	.696	.673	.654	.908	.751	**.686**	.536	.965	.962	**.959**	.955	.697
		RSE↓	.558	.824	.886	.944	.480	.596	.619	.636	.304	.500	**.561**	.682	.354	.361	**.363**	.378	.565
Spikingformer w/BitShift-RoPE (Ours)	R	R^2 ↑	**.722**	**.444**	**.325**	**.236**	**.806**	.696	**.681**	**.657**	**.924**	**.780**	.653	**.615**	**.967**	**.963**	**.959**	**.957**	**.712**
		RSE↓	**.553**	**.787**	**.867**	**.921**	**.477**	.597	**.611**	**.633**	**.276**	**.469**	.590	**.623**	**.347**	**.359**	**.363**	**.369**	**.553**
QKFormer w/Conv-PE	A	R^2 ↑	.680	.541	.424	.299	.739	.700	.671	.663	.889	.728	.540	.436	.944	.926	.912	.908	.688
		RSE↓	.597	.714	.800	.883	.553	.592	.621	.628	.342	.535	.696	.771	.420	.486	.527	.538	.606
QKFormer w/CPG-PE	A	R^2 ↑	.707	.534	**.426**	.299	.756	.693	.680	.674	.905	.735	.594	.574	.943	.942	.931	.928	.708
		RSE↓	.571	.720	**.799**	.883	.534	.599	.612	.618	.316	.528	.654	.670	.423	.428	.469	.477	.581
QKFormer w/Gray-PE	R	R^2 ↑	.686	.549	.408	.254	.791	.704	.680	.661	.915	.818	.761	.684	.948	**.945**	.931	.925	.729
		RSE↓	.591	.708	.812	.911	.494	.588	.612	.629	.290	.438	.502	.578	.406	**.416**	.469	.487	.558
QKFormer w/Log-PE	R	R^2 ↑	.710	.545	.411	.299	.739	.701	.679	.667	.917	.832	.755	.734	.947	.944	.932	.923	.733
		RSE↓	.566	.713	.807	.882	.552	.592	.614	.624	.297	.421	.508	.530	.410	.419	.462	.493	.556
QKFormer w/BitShift-RoPE (Ours)	R	R^2 ↑	**.717**	**.557**	.425	**.307**	**.795**	**.750**	**.682**	**.675**	**.929**	**.849**	**.776**	**.739**	**.959**	.941	**.940**	**.931**	**.748**
		RSE↓	**.562**	**.699**	.800	**.877**	**.487**	**.538**	**.609**	**.619**	**.273**	**.399**	**.486**	**.524**	**.359**	.432	**.436**	**.459**	**.535**

Table 3: Performance comparison on four text classification tasks using SpikeBERT. Accuracy (%) is mean ± s.d. over three seeds; the fine-tuned BERT row is a cited single-run number. The best results are shown in **bold**. PE types: A = absolute, R = relative.

Model	PE		Param(M)	Length = 128				Avg.
	Spike	Type		MR	SST-2	Subj	SST-5	
Fine-tuned BERT (floating-point reference)	×	A	109.8	86.39	92.01	95.43	49.87	80.93
SpikeBERT w/o PE	-	A	109.8	71.84 ± 0.41	80.17 ± 0.31	86.60 ± 0.26	35.20 ± 0.40	68.45
SpikeBERT w/CPG-PE	✓	A	110.4	72.73 ± 0.40	81.77 ± 0.34	86.76 ± 0.28	35.34 ± 0.41	69.15
SpikeBERT w/Gray-PE	✓	R	109.8	70.71 ± 0.44	81.08 ± 0.36	80.00 ± 0.27	34.39 ± 0.48	66.55
SpikeBERT w/Log-PE	✓	R	109.8	71.09 ± 0.47	81.42 ± 0.35	80.80 ± 0.25	34.43 ± 0.46	66.94
SpikeBERT w/BitShift-RoPE (Ours)	✓	R	109.8	**73.81 ± 0.43**	**82.00 ± 0.36**	**87.90 ± 0.24**	**35.93 ± 0.44**	**69.91**

Text classification. As shown in Table 3, BitShift-RoPE records the highest accuracy on all four benchmarks with an average of 69.91% against 69.15% for CPG-PE. The margin over CPG-PE is resolvable on MR ($p = 0.033$) and Subj ($p = 0.006$) but not on SST-2 ($p = 0.47$) or SST-5 ($p = 0.17$). BitShift-RoPE achieves this without adding parameters and maintains the base model size of 109.8M, whereas CPG-PE adds 0.6M.

Degradation under prior relative PEs. Gray-PE and Log-PE reduce accuracy below the SpikeBERT baseline without any explicit positional encoding. The baseline averages 68.45% whereas Gray-PE and Log-PE average 66.55% and 66.94%. The drop is resolvable on MR, Subj, and SST-5 for both methods ($p \leq 0.015$) and on SST-2 for Gray-PE ($p = 0.035$).

5.3 Image Classification

To examine the 2D-extension effect across multiple settings, we evaluate on static-image benchmarks (CIFAR10, CIFAR100 [13], Tiny-ImageNet) and on neuromorphic event benchmarks (CIFAR10-DVS [16], DVS128-Gesture [1]) that include spatio-temporal dynamics.

Image classification. As Table 4 shows, BitShift-RoPE 2D (BSR 2D) records the highest accuracy on every benchmark and backbone. With the QKFormer backbone, BSR 2D reaches 96.36% on CIFAR10, 81.62% on CIFAR100, and 59.05% on Tiny-ImageNet. The margins are resolvable on CIFAR100 and Tiny-ImageNet for both backbones and on CIFAR10 and DVS128-Gesture for Spikformer. On QKFormer CIFAR10 ($p = 0.75$ against Conv-PE) and on CIFAR10-DVS, where

Table 4: Performance comparison on image classification benchmarks. Accuracy (%) is the mean $\pm$ s.d. over three seeds; the ViT row is a cited single-run number. †CIFAR10-DVS has 1,000 test samples, so accuracy is quantized to 0.1%, and several s.d. are 0. The best results are shown in **bold**. PE types: A = absolute, R = relative.

Model	PE		Standard Benchmarks			Neuromorphic Benchmarks		Avg.
	Spike	Type	CIFAR10	CIFAR100	Tiny-ImageNet	CIFAR10-DVS†	DVS128-Gesture	
Vision-Transformer (floating-point reference)	×	A	96.73	81.02	62.18	-	-	-
Spikformer w/Conv-PE	✓	A	94.61 ± 0.14	77.55 ± 0.15	55.28 ± 0.11	80.3 ± 0.0	87.59 ± 0.39	79.01
Spikformer w/CPG-PE	✓	A	94.94 ± 0.12	77.51 ± 0.18	54.76 ± 0.14	80.8 ± 0.0	88.97 ± 0.36	79.40
Spikformer w/Gray-PE 2D	✓	R	93.74 ± 0.28	77.53 ± 0.19	55.45 ± 0.13	80.1 ± 0.06	88.36 ± 0.37	79.04
Spikformer w/BitShift-RoPE 2D (Ours)	✓	R	$\mathbf{95.13 \pm 0.15}$	$\mathbf{77.68 \pm 0.16}$	$\mathbf{56.31 \pm 0.15}$	$\mathbf{81.2 \pm 0.0}$	$\mathbf{93.79 \pm 0.40}$	**80.82**
QKFormer w/Conv-PE	✓	A	96.32 ± 0.15	80.90 ± 0.14	58.64 ± 0.12	85.1 ± 0.06	98.26 ± 0.24	83.84
QKFormer w/CPG-PE	✓	A	96.21 ± 0.17	80.65 ± 0.19	56.75 ± 0.16	82.0 ± 0.12	97.26 ± 0.21	82.57
QKFormer w/Gray-PE 2D	✓	R	96.03 ± 0.22	80.70 ± 0.18	57.48 ± 0.14	81.4 ± 0.2	97.91 ± 0.28	82.70
QKFormer w/BitShift-RoPE 2D (Ours)	✓	R	$\mathbf{96.36 \pm 0.13}$	$\mathbf{81.62 \pm 0.18}$	$\mathbf{59.05 \pm 0.11}$	$\mathbf{85.2 \pm 0.06}$	$\mathbf{98.61 \pm 0.22}$	**84.17**

Table 5: Efficiency comparison on the Spikformer backbone. $\downarrow$ indicates lower is better and $\uparrow$ indicates higher is better.

Methods	Params (M) ($\downarrow$)	FLOPs (M) ($\downarrow$)	SOPs (M) ($\downarrow$)	XNOROPs (M) ($\downarrow$)	Energy (mJ/sample) ($\downarrow$)	Latency (ms/sample) ($\downarrow$)	Throughput (samples/s) ($\uparrow$)
Vanilla Spikformer	1.67	24.70	176.80	0	0.216	4.03 ± 0.02	248.0
CPG-PE	1.71	26.59	180.41	0	0.224	4.41 ± 0.81	226.5
XNOR Gray-PE	1.98	31.54	207.56	75.87	0.260	10.92 ± 0.92	91.6
XNOR Log-PE	1.98	31.54	207.35	57.80	0.260	10.96 ± 0.72	91.3
BitShift-RoPE	**1.67**	**24.70**	**176.80**	**0**	**0.216**	$\mathbf{4.06 \pm 0.02}$	**246.3**

accuracy is quantized to 0.1%, we claim no difference. All margins are obtained at zero additional parameters.

Spatio-temporal benchmarks. On DVS128-Gesture, a neuromorphic benchmark with interleaved spatio-temporal dynamics, BSR 2D reaches 98.61% on QKFormer and 93.79% on Spikformer. The Spikformer margin over the strongest baseline is $+4.8$ points ($p < 0.001$). We attribute this to the orthogonal split, which keeps the two spatial coordinates separable in the attention map (Section 4.3) rather than collapsing them when patches are flattened into a 1D sequence.

Qualitative analysis via Grad-CAM. Beyond quantitative accuracy, we qualitatively evaluate the spatial recognition capability of each positional encoding using Grad-CAM. The visualizations show that BitShift-RoPE 2D localizes salient objects and preserves spatial coherence. The attention maps of existing relative SNN PEs are more scattered. One plausible contributing factor is that their multi-bit outputs densify the attention map. We do not isolate the cause here. Detailed visualizations and further qualitative discussions are provided in Appendix G.

5.4 Efficiency Comparison

To quantitatively assess the operation-level efficiency of BitShift-RoPE under neuromorphic-style cost accounting, we compute the cost of one forward pass under identical experimental conditions (sequence length $L = 168$, feature dimension $D = 256$, time steps $T = 4$, on the Electricity benchmark) for vanilla SSA and four PE methods. We do not deploy on a dedicated neuromorphic chip; latency is wall-clock on a GPU under PyTorch, and energy is estimated from operation counts as described below.

Hardware energy consumption is estimated using the Horowitz energy model [8] under 45nm CMOS, with per-operation costs $E_{\text{FLOPs}} = 9.2$ pJ, $E_{\text{SOP}} = 0.9$ pJ, and $E_{\text{XNOR}} = 0.013$ pJ; the total energy E_{total} is the weighted sum of these. The reported energy is therefore an operation-count estimate, not a measurement on a dedicated neuromorphic device, and the absolute scale would shift under a different process node. Table 5 reports the aggregate efficiency results.

Zero parameter and FLOP overhead. The most striking observation is that BitShift-RoPE matches the baseline in both parameter count (1.67 M) and FLOPs (24.70 M) exactly. This contrasts with the CPG-PE, which incurs additional overhead due to its extra linear layer. The result validates the methodology's claim that our approach pre-computes positional indices at compile time, caches them in a hardware buffer, and at runtime performs only memory-pointer routing, thereby completely suppressing arithmetic-unit usage.

Energy consumption. Under the Horowitz operation-count model, BitShift-RoPE consumes 0.2159 mJ per sample, matching the baseline's ultra-low-power characteristic. In contrast to existing PE methods, BitShift-RoPE adds relative-position awareness without an estimated hardware power or memory penalty.

Inference latency and throughput. On GPU under PyTorch, BitShift-RoPE achieves a latency of 4.06 ± 0.02 ms and a throughput of 246.3 samples/s, within 1% of the vanilla Spikformer baseline (4.03 ms and 248.0 samples/s). Gray-PE and Log-PE slow substantially. Their latency nearly triples (over 10.9 ms) and their throughput drops by more than 60% ($\sim$91 samples/s). The cause is loss of sparsity rather than the XNOR gate itself. $\mathrm{XNOR}(q, k) = 1$ whenever the operands agree, including 0–0, so nearly every term of their similarity is non-zero, and the accumulation becomes dense. Gray-PE compounds this with its dimension expansion. BitShift-RoPE instead keeps the operands sparse and binary, so the attention block retains the baseline's cost on this hardware. Absolute latencies on a dedicated neuromorphic chip would differ in scale.

6 Limitations

Pre-computed shift-map memory footprint. To eliminate runtime arithmetic, BitShift-RoPE pre-computes the per-group shift counts $k_g(m) = \text{round}\left(m \cdot C^{-\frac{g}{G-1}}\right)$ for every position m in the sequence and caches the resulting routing indices in a hardware buffer. The cache size scales as $\mathcal{O}(N \cdot G)$ in sequence length N and group count G (see Table 1). For ML-scale sequence lengths, this overhead is negligible (e.g., $N = 168$, $G = 4$ requires only a few hundred integers), but for extremely long-context inference – such as a foundation-scale SNN with $N \gg 10^4$ – the per-token routing-address storage may strain neuromorphic memory budgets.

Fixed analytical decay constant C. To minimize the number of trainable parameters, we map the multi-frequency distribution via a single fixed decay constant C rather than learning per-group frequencies from data. This design preserves the parameter-free property of the positional encoder and ensures the rotation mechanism is identical across data domains (a desirable inductive bias for neuromorphic deployment). The trade-off is that C cannot adapt to domain-specific frequency distributions. For example, time series with strong daily periodicity may benefit from a different C than text or image inputs. Our experiments use $C = 64.0$ across all three domains (Appendix B.2), which works well in practice but is unlikely to be Pareto-optimal for any single domain. We discuss adaptive variants in Appendix H (Future work).

Hardware-energy estimate vs. measured energy. The energy comparison in Section 5.4 and Table 5 uses the Horowitz energy model [8], which estimates per-operation energy from algorithmic FLOP / SOP / XNOR counts. This is the standard methodology for SNN-energy comparisons, but it assumes the underlying neuromorphic hardware actually achieves the per-operation costs reported in 45 nm CMOS. Wall-clock energy on real silicon (e.g., Loihi-2 [3], TrueNorth [21]) may diverge due to memory-access patterns, sparsity exploitation, or accelerator-specific constraints. Validating our zero-FLOP claim on real neuromorphic hardware is a target for follow-up work.

7 Conclusion

We have proposed BitShift-RoPE, the first method to inject relative positional information into the attention mechanism of Transformers while preserving the binary property of Spiking Neural Networks. Whereas existing positional encoding methods either force floating-point operations or convert spikes into multi-bit integer tensors, eroding SNNs' signature sparsity and computational efficiency, our method addresses this by replacing continuous-space trigonometric rotation with a discrete-space cyclic shift. From a hardware implementation perspective, BitShift-RoPE reduces positional-encoding overhead to 0 FLOPs through memory-pointer routing. Furthermore, by preserving the pure binary integrity of the query and key tensors, the attention dot product reduces to multiplier-free bitwise AND operations. Across time-series forecasting, text classification, and standard and neuromorphic vision benchmarks, BitShift-RoPE is consistently at least as good as prior SNN positional encodings on every backbone and benchmark, at zero additional positional arithmetic.

Acknowledgments and Disclosure of Funding

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-00521452), and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-15002982). The authors declare no competing interests.

References

[1] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7397, 2017.

[2] Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*, 2018.

[3] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018.

[4] 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*, pages 248–255. IEEE, 2009.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019.

[6] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL `https://openreview.net/forum?id=6OoCDvFV4m`.

[7] Seung-Kyu Hong and Hyuk-Yoon Kwon. Adaptive spike neural networks for natural language inference tasks with dynamic spike predictor. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 4797–4801. ACM, 2025. doi: 10.1145/3746252.3760917.

[8] Mark Horowitz. 1.1 Computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10–14, 2014. doi: 10.1109/ISSCC.2014.6757323.

[9] Hosagrahar V Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

[10] Min-Seon Kim, Ling Liu, and Hyuk-Yoon Kwon. OL4TeX: Adaptive online learning for text classification under distribution shifts. In *Proceedings of the IEEE International Conference on Big Data (BigData)*, pages 1340–1345, 2024. doi: 10.1109/BigData62323.2024.10826003.

[11] Min-Seon Kim, Ling Liu, and Hyuk-Yoon Kwon. Modular model adaptation for online learning in streaming text classification. *IEEE Transactions on Knowledge and Data Engineering*, 38(3): 1843–1856, 2026. doi: 10.1109/TKDE.2026.3651427.

[12] Taehee Kim, Jae-Seok Jang, and Hyuk-Yoon Kwon. Correlation-driven multi-level learning for anomaly detection on multiple energy sources. *Applied Soft Computing*, 2024.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[14] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 95–104, 2018.

[15] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:244131, 2017.

[16] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11:309, 2017. doi: 10.3389/fnins.2017.00309.

[17] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

[18] Changze Lv, Tianlong Li, Jianhan Xu, Chenxi Gu, Zixuan Ling, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. SpikeBERT: A language Spikformer learned from BERT with knowledge distillation. *arXiv preprint arXiv:2308.15122*, 2023.

[19] Changze Lv, Yansen Wang, Dongqi Han, Xiaoqing Zheng, Xuanjing Huang, and Dongsheng Li. Advancing spiking neural networks for sequential modeling with central pattern generators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/hash/2f55a8b7b1c2c6312eb86557bb9a2bd5-Abstract-Conference.html`.

[20] Changze Lv, Yansen Wang, Dongqi Han, Yifei Shen, Xiaoqing Zheng, Xuanjing Huang, and Dongsheng Li. Toward relative positional encoding in spiking transformers. *arXiv preprint arXiv:2501.16745*, 2025.

[21] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345 (6197):668–673, 2014.

[22] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 271–278, 2004.

[23] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer, editors, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219855. URL `https://aclanthology.org/P05-1015/`.

[24] Jong Seong Park, Jeong-Ha Park, Jihyeok Choi, and Hyuk-Yoon Kwon. Learning with correlation-guided attention for multi-energy consumption forecasting. *IEEE Transactions on Industrial Informatics*, 2024.

[25] Jong Seong Park, Jeong-Ha Park, Ji-Hyeok Choi, and Hyuk-Yoon Kwon. Self-supervised inference of missing energy sources for enhancing energy consumption forecasting. *Applied Soft Computing*, 192, 2026. doi: 10.1016/j.asoc.2026.114732.

[26] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[27] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[29] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, 2013.

[30] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[32] Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=9FmolyOHi5.

[33] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954*, 2023.

[34] Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. QKFormer: Hierarchical spiking transformer using Q-K attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://openreview.net/forum?id=AVd7DpiooC.

[35] Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=frE4fUwz_h.

A Proofs

A.1 Proof of Proposition 1: DFT Diagonalization of the Cyclic Shift

This section proves Proposition 1 from Section 4. The result is a textbook property of circulant matrices. We include it to fix notation and to make explicit which features of RoPE the construction retains.

Proof. The cyclic shift matrix $S \in \mathbb{R}^{d\times d}$ over a d-dimensional group is a circulant matrix. Every circulant matrix is diagonalized by the Discrete Fourier Transform (DFT) matrix F as

$$S = F^{-1}\Lambda F,$$

where Λ is a diagonal matrix whose entries are the eigenvalues of S. The r-th eigenvalue λ_r is a root of unity

$$\lambda_r = e^{-i\frac{2\pi r}{d}}, \qquad r \in \{0, 1, \ldots, d-1\}.$$

Applying the shift matrix m times gives

$$S^m = (F^{-1}\Lambda F)^m = F^{-1}\Lambda^m F,$$

and each entry of Λ^m is

$$(\lambda_r)^m = \left(e^{-i\frac{2\pi r}{d}}\right)^m = e^{-im\theta_r}, \qquad \theta_r = \frac{2\pi r}{d}.$$

Hence an m-step cyclic shift acts in the frequency domain as a phase rotation linear in m at every mode r. Standard RoPE applies a phase rotation $\tilde{q}_m = q \cdot e^{im\theta}$ of the same form. The correspondence is structural rather than pointwise. The frequency schedule differs: S^m activates uniformly spaced modes $\theta_r = 2\pi r/d$, whereas RoPE uses the exponential schedule $\theta_i \propto 10000^{-2i/D}$, and RoPE does not reproduce continuous rotation and continuous rescaling. What is preserved exactly and independently of the schedule is that S is orthogonal with $(S^n)^\top = S^{-n}$, that shift composition is additive, and that the attention product admits the shift-difference form $QS^a(KS^b)^\top = QS^{a-b}K^\top$ (Section 4). □

A.2 Corollary: Extension to Real 2D Rotation Blocks

Corollary 1 (Extension to Real 2D Rotation Blocks)**.** *When the input feature vector lies in real space $\mathbb{R}^d$, the complex diagonal matrix Λ^m from Proposition 1 is similar to a block-diagonal matrix of 2×2 real rotation blocks of the same form as standard RoPE.*

Proof. Since the input feature vector lies in $\mathbb{R}^d$, the Fourier-domain eigenvalues form complex-conjugate pairs λ_r and λ_{d-r}. Restructuring these conjugate pairs through a real basis-change matrix T makes Λ^m similar to a block-diagonal matrix $R \in \mathbb{R}^{d\times d}$ composed of 2×2 real rotation blocks

$$R(\theta_r) = \begin{pmatrix} \cos(m\theta_r) & -\sin(m\theta_r) \\ \sin(m\theta_r) & \cos(m\theta_r) \end{pmatrix}.$$

The permutation of a 1D spike array therefore corresponds to a combination of independent 2D rotation planes in $\mathbb{R}^d$. □

A.3 Corollary: Bounded Rounding Error of the Multi-Frequency Shift

Corollary 2 (Bounded, Non-Accumulating Rounding Error)**.** *Let $x = m \cdot C^{-g/(G-1)}$ and $k_g(m) = \mathrm{round}(x)$ as in Section 4.2. At DFT mode r over a group of width d, the phase error ϵ_r between the applied rotation $e^{-ik_g(m)\theta_r}$ and the continuous target $e^{-ix\theta_r}$ satisfies, independently of m,*

$$\epsilon_r \leq \frac{\pi r}{d}.$$

The bound is 0 for $g = 0$, reaches $\pi/2$ per position at the Nyquist mode $r = d/2$, and π for a query–key pair at that mode.

Proof. By Proposition 1, S^x acts at mode r as $e^{-ix\theta_r}$, so replacing x by $\mathrm{round}(x)$ quantizes the phase angle at that mode. By the rounding property

$$|x - \mathrm{round}(x)| \leq \tfrac{1}{2} \implies \epsilon_r \leq \tfrac{1}{2}\,\theta_r = \frac{\pi r}{d}.$$

The bound depends on r and d only and not on m. For $g = 0$ the scale is $C^0 = 1$, so x is an integer and $\epsilon_r = 0$. At $r = d/2$ the bound equals $\pi/2$, and the pairwise phase error between a query at m and a key at n is at most twice that, namely π. The bound is attained and is not small at the Nyquist mode. What it establishes is non-accumulation. The error does not grow with the absolute position m or with the sequence length. Aggregation across G groups in the attention dot product follows by linearity. □

B Experimental Settings

This appendix provides comprehensive experimental setup details for all experiments reported in Section 5, including datasets, hyperparameters, and evaluation metrics.

B.1 Datasets

B.1.1 Time Series Forecasting

We evaluate on four standard time-series benchmarks covering traffic, solar generation, and electricity consumption. **Metr-la** [9] contains traffic speed measurements collected from Los Angeles County highways, and **Pems-bay** [17] provides analogous traffic speed measurements from the California Bay Area. Together, these test the model on volatile spatially correlated traffic flows. **Solar** [14] records solar power generation from the 2006 US National Solar Radiation Database and captures strong daily-period structure. **Electricity** [14] records power consumption across 321 customers, providing a high-dimensional multivariate setting.

B.1.2 Text Classification

We evaluate on four text classification benchmarks covering sentiment analysis, subjectivity, and semantic similarity. **MR** [23] is a binary sentiment classification dataset of positive/negative movie reviews. **SST-2** and **SST-5** [29] are derived from the same Stanford Sentiment Treebank. SST-2 provides binary sentiment labels at the sentence level, while SST-5 provides finer-grained five-class labels ranging from very positive to very negative. **Subj** [2] requires classifying sentences as subjective or objective, testing register sensitivity rather than polarity.

B.1.3 Image Classification

We evaluate on five image classification benchmarks spanning two physically distinct modalities. The neuromorphic benchmarks consist of **DVS128-Gesture** [1], a hand-gesture event-stream dataset captured with a DVS128 silicon retina, and **CIFAR10-DVS** [16], an event-stream version of CIFAR-10 obtained by recording images shown on a monitor. Both benchmarks emphasize the temporal causality of spike events while maintaining low spatial resolution. The standard benchmarks consist of **CIFAR-10** and **CIFAR-100** [13], two static-image classification datasets with 32×32 RGB images and 10/100 classes respectively, and **Tiny-ImageNet**, a 64×64-resolution 200-class subset of ImageNet [4].

B.1.4 Dense Prediction

We evaluate dense prediction on the **Oxford-IIIT Pet** [26] trimap segmentation task, a three-class task (foreground, background, and boundary) over 37 cat and dog breeds. Images are resized to 128×128 for both training and evaluation. The boundary class is the one most sensitive to fine spatial position, which is why we report boundary IoU alongside mIoU. The decoder is a per-token linear projection with no convolution and no skip connection, so any spatial coherence in the prediction must originate in the encoder.

B.2 Hyperparameter Settings

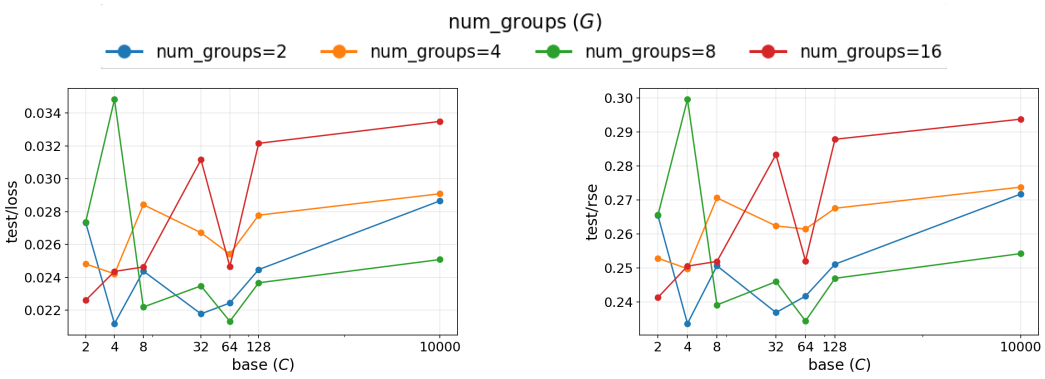
We run all experiments under three independent seeds and report the average performance. Table 6 lists the LIF neuron parameters and other hyperparameters used across all experiments.

Table 6: Experiment configuration and hyperparameter settings for time series forecasting, text classification, image classification, and dense prediction, including LIF neuron parameters, training settings, and positional-encoding-specific parameters.

Parameter	Time Series Forecasting	Text Classification	Image Classification	Dense Prediction
Common LIF Neuron Parameters				
Membrane leak time constant (τ)	2.0	2.0	2.0	2.0
Common firing threshold (V_{th})	0.8	0.8	1.0	1.0
Training & Model Parameters				
Batch Size	64	64	64	64
Learning Rate	1e-3	5e-4	5e-4	5e-4
Optimizer	Adam	Adam	AdamW	AdamW
Embedding Dimension (D)	256	768	384	256
Layer Depth	2	12	4	4
Time Steps (T)	4	4	4	4
Attention Heads	8	8	12	12
Max Sequence Length	-	256	-	-
Input Resolution	-	-	32×32 / 64×64	128×128
Positional Encoding Parameters				
CPG Neurons (N_{CPG})	40	20	20	20
BitShift-RoPE Base (C)	64.0	64.0	64.0	64.0
BitShift-RoPE Groups (G)	4	4	4	4

C Hyperparameter Selection

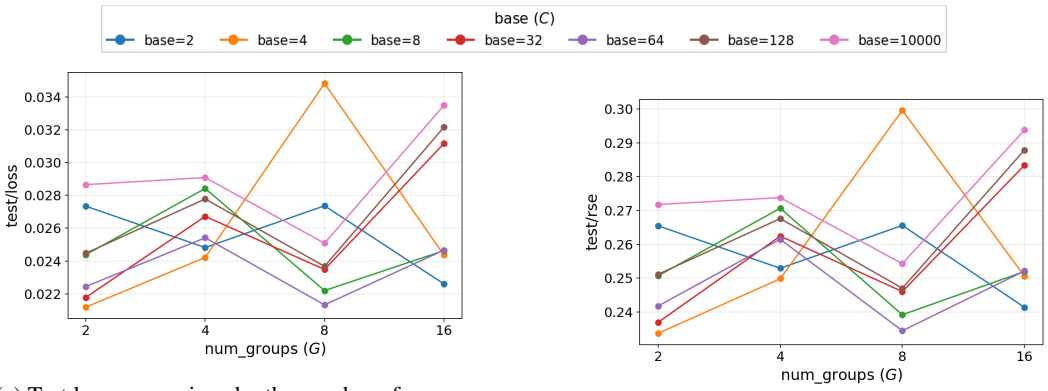
To analyze the impact of the core hyperparameters of the proposed BitShift-RoPE, namely the decay constant C (base) and the number of channel groups G (num_groups), on the model's predictive



(a) Test loss comparison by base (C).

(b) RSE comparison by base (C).

Figure 2: Performance variation according to the base (C).



(a) Test loss comparison by the number of groups (G).

(b) RSE comparison by the number of groups (G).

Figure 3: Performance variation according to the number of groups (G).

performance and training stability, we conducted an ablation study. Figure 2 and Figure 3 show trends in Test Loss and RSE across combinations of these two parameters.

Performance and stability analysis regarding the decay constant (C, base). Examining the results in Figure 2, when the decay constant is excessively small (e.g., 2 or 4) or extremely large (e.g., 10000), the Loss and RSE tend to fluctuate unstably or show performance degradation across most group settings. This indicates that when the shift-frequency spectrum skews toward either the high- or low-frequency bands, the model fails to fully capture multi-scale patterns in the time-series data. In contrast, around a base of 64, each group captures short-term volatility and long-term dependencies in a balanced manner, minimizing performance deviation and converging to the most stable lower bound.

Representation power optimization regarding the number of channel groups (G, num_groups). Analyzing the graphs showing variations by the number of groups in Figure 3, the setting of $G = 2$ exhibited somewhat limited representation power due to a lack of diversity in the frequency bands. Conversely, when the channel groups were excessively partitioned into $G = 8$ or $G = 16$, the Loss and RSE spiked sharply around certain base values, significantly increasing training instability. When the number of channels per partition becomes too small, the representational power of the information responsible for a specific frequency band inherently degrades. In contrast, the $G = 4$ setting maintained the lowest error rate and remained stable without abrupt spikes, even across various base values.

Selection of optimal hyperparameters. In conclusion, synthesizing the above experimental results, the optimal regime where the model converged most stably with minimal performance deviation was the condition of $G = 4$ and $C = 64$. Therefore, to robustly adapt to variations across diverse feature

dimensions and to most effectively learn multi-scale contexts, we conducted the main experiments with the final hyperparameters set to *num_groups=4* and *base=64.0*.

D Dense Prediction

Classification alone cannot establish that positional content rather than efficiency drives the gains. We therefore evaluate on Oxford-IIIT Pet trimap segmentation with three classes, including boundary. The backbone is Spikformer at embedding dimension 256 and depth 4. We train five PE arms over three seeds on an identical 80-epoch budget at the training resolution of 128×128. The decoder is a per-token linear projection with no convolution and no skip connection. Any spatial coherence must therefore originate in the encoder. The No-PE arm quantifies the position that Spikformer's convolutional stem supplies by itself.

Table 7: Oxford-IIIT Pet trimap segmentation at 128 px (mean $\pm$ s.d., 3 seeds). †Gray-PE 2D requires token-mean normalization of the attention map because its non-negative Hamming similarity saturates the attention LIF at 256 tokens. Without it, the arm reaches 0.366 mIoU.

PE	Params	mIoU	Boundary IoU
No PE	3.61 M	0.638 ± 0.007	0.335 ± 0.010
Conv-PE	4.20 M	0.665 ± 0.006	0.376 ± 0.009
CPG-PE	3.67 M	0.660 ± 0.005	0.363 ± 0.006
Gray-PE 2D†	3.61 M	0.502 ± 0.051	0.261 ± 0.043
BitShift-RoPE 2D (Ours)	**3.61 M**	$\mathbf{0.673 \pm 0.003}$	$\mathbf{0.381 \pm 0.003}$

Against the No-PE floor, BitShift-RoPE gains $+0.035$ mIoU ($p = 0.006$) and $+0.046$ boundary IoU ($p = 0.011$). Boundary is the class most sensitive to fine spatial position. Since the decoder cannot generate spatial structure, both gains originate in the encoding. BitShift-RoPE also exceeds CPG-PE by $+0.013$ mIoU ($p = 0.026$) and $+0.018$ boundary IoU ($p = 0.020$). Gray-PE 2D falls below the floor by -0.136 mIoU ($p = 0.042$). Against Conv-PE the margins of $+0.008$ ($p = 0.133$) and $+0.005$ ($p = 0.442$) are not resolvable at $n = 3$. We therefore call the two comparable and claim no win. What differs is cost. Conv-PE adds 0.59 M parameters whereas BitShift-RoPE adds none and performs no forward-path positional arithmetic.

E Implementation Details

This appendix describes the implementation of BitShift-RoPE referred to in Section 4.4. The core principle relies on pre-computing the multi-frequency shift indices and applying them at runtime using a single, highly optimized memory "gather" operation.

E.1 Pre-computation of the Routing Index Map

As formalized in Eq. 8, the exponential-decay-based shift count for position m and group g is defined as $k_g(m) = \text{round}\left(m \cdot C^{-\frac{g}{G-1}}\right)$. Because $k_g(m)$ depends only on the sequence length N, the decay constant C (`base`), and the group configuration G (`num_groups`), the module can precompute the entire coordinate mapping before the forward pass begins.

During initialization (or when the sequence length N changes), the module computes the necessary cyclic shifts. For a given sub-dimension d (`group_dim`), the shifted index for the original coordinate is evaluated using modulo arithmetic as $(\text{idx}_{base} - k_g(m)) \bmod d$. These shifted relative indices are then offset by $(g \times d)$ to map them back to the global feature dimension D. The resulting index tensor is registered as a non-trainable buffer (`_cached_idx`), completely eliminating the need for floating-point arithmetic (FLOPs) during inference.

E.2 Runtime Memory Pointer Routing (Zero-FLOPs)

During the actual forward pass, the input spike tensor $X \in \{0,1\}^{T\times B\times H\times N\times D}$ does not undergo any matrix multiplication or addition. Instead, the cached index tensor is expanded to match the input shape, and a single `torch.gather` operation is executed along the feature dimension (dim = -1).

In PyTorch, this is an indexed gather over a pre-built table rather than a literal pointer-routing primitive. At the hardware level, memory-pointer routing is a plausible realization of the same deterministic permutation, in which the memory controller reads the spike values using the shifted address offsets provided by the cached index map. In either realization, the data values are merely relocated, so the permuted Q and K tensors strictly retain their pure binary properties.

F Detailed Computational Complexity Analysis

This section provides a detailed breakdown of the computational overhead for various Positional Encoding (PE) methods, expanding upon the summary presented in Section 4.5 and Table 1. With sequence length N and feature dimension D, we analyze the detailed complexity as follows.

ANN RoPE. In the conventional ANN setting [31], RoPE performs $\sin$ / $\cos$ matrix multiplications during PE application, requiring $\mathcal{O}(N \cdot D)$ FLOPs. Furthermore, since Q and K are floating-point tensors, the subsequent attention-score computation also incurs $\mathcal{O}(N^2 \cdot D)$ FLOPs, which dominates the energy budget.

CPG-PE. This method generates dynamic positional spikes using convolution and LIF neurons, then adds them to the original data. This requires an additional $\mathcal{O}(N \cdot K \cdot D)$ operations and parameters proportional to the convolution kernel size K.

Gray-PE and Log-PE. Gray-PE concatenates a Gray code onto the end of the feature vector, inflating the tensor dimension to $D + k$ and increasing both memory usage and computational complexity. Log-PE requires lookup-table memory accesses to add bias values to the attention map. The most critical issue is that both methods produce integer-type encoding outputs, making bitwise AND attention impossible. As a result, they cannot exploit sparsity and instead force a dense XNOR computation, incurring $\mathcal{O}(N^2 \cdot D)$ integer-operation overhead.

BitShift-RoPE (Ours). Our proposed method requires $\mathcal{O}(N \cdot D)$ floating-point operations only to derive the exponential-decay-based shift indices. However, we perform this computation once per sequence length N and cache the resulting integer shift map in a hardware buffer. As a result, no arithmetic logic unit is involved during inference; the positional encoding is applied via a memory-pointer routing scheme that merely manipulates the start-address offset of the spike data. The dynamic computation cost of the network thus converges to zero FLOPs. Furthermore, the permuted Q and K tensors retain their pure binary property, so the attention dot product requires neither floating-point nor XNOR operations, but only $\mathcal{O}(N^2 \cdot D)$ SOPs, preserving the full SNN advantage.

G Qualitative Analysis with Grad-CAM

To qualitatively analyze how BitShift-RoPE captures spatial features in vision tasks, we visualize attention maps using Grad-CAM (Gradient-weighted Class Activation Mapping) [28].

G.1 Gradient-based Attention Contribution in SNNs

To determine how each spiking attention map A_l in the SNN-based Transformer influences the final class prediction y^c, we compute the contribution of each attention element via $A_l \odot \nabla A_l$.

The contribution of a particular element $A_l[i,j]$ to the class score is approximated as the product of its activation magnitude (how strongly it attends) and the gradient of the class score with respect to the element (how much y^c changes per unit change in the element):

$$\text{Contribution}_{i,j} \approx A_l[i,j] \times \frac{\partial y^c}{\partial A_l[i,j]}.$$

This can be interpreted as the first-order Taylor expansion of the objective $f(A) = y^c$:

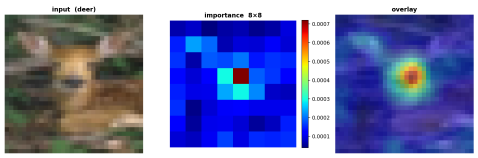
$$y^c(A_l + \Delta A_l) \approx y^c(A_l) + \sum_{i,j} \nabla A_l[i,j] \cdot \Delta A_l[i,j].$$

By setting the perturbation $\Delta A_l[i,j]$ to the element value $A_l[i,j]$, the expression captures the change in score from removing and re-enabling that attention connection. In effect, this measures how critical a specific bitwise-AND firing in SNN attention is to the model's final decision.

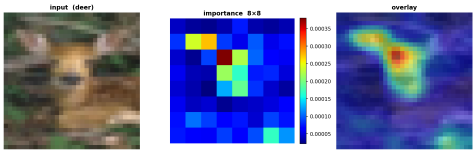
G.2 Visual Analysis on Object Localization

Figure 4 compares Grad-CAM visualizations for different positional encoding methods using the Spikformer backbone on CIFAR-10. By inspecting how accurately the model focuses on the salient object in the input image, we can assess each PE's spatial-recognition fidelity.

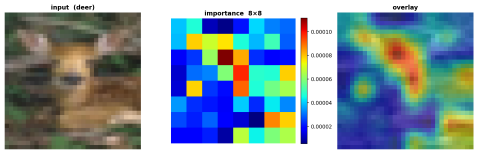
- **Vanilla SSA & CPG-PE.** The Vanilla SSA (Spikformer) model without explicit positional encoding (Figure 4a) fails to capture object features clearly and responds only to limited regions. CPG-PE (Figure 4b) focuses on the object's central region to some extent, but because it depends on absolute coordinates, it does not fully cover the object's flexible spatial extent.
- **Gray-PE 2D.** The contribution map of Gray-PE 2D (Figure 4c) is scattered across the image. One plausible contributing factor is that its multi-bit outputs densify the attention map, though we do not isolate the cause here.
- **BitShift-RoPE 2D (Ours).** In contrast, BitShift-RoPE 2D (Figure 4d) shows gradient activations concentrated along the object's face and outline. Decomposing the 2D plane through discrete cyclic shifts, without floating-point arithmetic on the encoding path, provides a visual inductive bias while keeping the queries and keys binary.



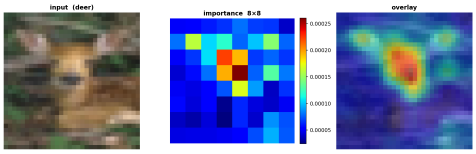
(a) Vanilla SSA (Spikformer)



(b) CPG-PE



(c) Gray-PE 2D



(d) BitShift-RoPE 2D

Figure 4: Grad-CAM comparison across PE methods on Spikformer + CIFAR-10.

H Discussion

This appendix discusses the broader significance of the work and outlines future research directions. Section 6 treats limitations separately, and Appendix I provides the broader impact statement.

Preserving spike-domain attention. Recent SNN Transformer research has injected positional information by directly adding real or integer values to the attention map. This converts the spike tensor into multi-bit data and makes SNN's bitwise AND attention unavailable, so the accumulation that follows becomes dense. The main contribution of this work is to inject relative position through a permutation of the query and key tensors. The permutation adds no floating-point arithmetic on the encoding path and keeps queries and keys binary, so attention remains a sparse bitwise AND. This is an architectural arithmetic property pending silicon validation (Section 6).

Future work. Future work will explore an Adaptive BitShift-RoPE that dynamically optimizes the decay constant and shift frequencies during training, tailored to the data domain. We further plan to scale our method to large-parameter SNN-based foundation models and rigorously validate its long-context handling.

I Broader Impact Statement

Positive impacts. BitShift-RoPE addresses a foundational efficiency bottleneck in SNN Transformers by eliminating runtime floating-point arithmetic and preserving bitwise-AND attention. The downstream impact is energy efficiency across the configurations we evaluate; our method matches the vanilla SNN backbone's per-sample energy (0.216 mJ/sample on Spikformer + Electricity), consuming roughly $20\times$ less energy than the equivalent floating-point Transformer. At the deployment scale, this could enable Transformer-quality inference on edge devices and battery-powered sensors that today rely on architectures with weaker accuracy. Concrete positive applications include low-power health-monitoring devices, on-board satellite forecasting, distributed sensor networks for environmental monitoring, and brain-computer interfaces, all of which directly benefit from SNN energy efficiency. By making relative positional encoding tractable for SNN Transformers, our work helps close the accuracy gap between neuromorphic and conventional architectures, enabling broader adoption of energy-efficient ML.

Potential negative impacts and mitigations. The proposed method is a foundational module (a positional-encoding mechanism), not an end-application system, so direct misuse pathways are limited. However, the broader category of more-efficient Transformers does have indirect risks worth flagging:

- **Surveillance and tracking.** More energy-efficient on-device transformers lower the cost of deploying always-on inference (e.g., in vision-equipped sensors), which can amplify surveillance applications. We do not see any property of our method that uniquely enables surveillance beyond what existing efficient architectures already permit, but the cumulative effect of efficiency improvements warrants ongoing scrutiny.
- **Reduced auditability of low-power inference.** Edge deployment can reduce the visibility of model behavior compared to cloud inference. As mitigation, our method does not affect the interpretability of the underlying model; standard SNN interpretability techniques (such as the Grad-CAM analysis in Appendix G) continue to apply.
- **Foundation-model scale-up.** Our work suggests SNN Transformers are a viable substrate for larger models. As with any capability-scaling research, this carries the dual-use risks common to large-model research. Downstream applications, not the encoder itself, would determine the magnitude of any harm.

No human subjects, no scraped data. Our experiments use only public benchmark datasets (Section B.1). We conducted no human-subject experiments or data collection. No new datasets are released. The method is a small architectural change (a positional encoding) and does not introduce risks of bias amplification beyond those inherent to the underlying backbone networks and datasets.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state three contributions (binary-preserving relative PE, zero added FLOPs with bitwise-AND attention), each backed by Proposition 1 (Section 4, full proof in Appendix A.1), Table 5 for energy parity (0.216 mJ/sample identical to vanilla Spikformer), and Tables 2, 3, 4 for the three-domain accuracy numbers (avg R^2 0.758, 69.91%, 84.17%) reported in the abstract.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 provides a dedicated Limitations section covering pre-computed shift-map memory footprint, the fixed analytical decay constant C, and the gap between Horowitz-model energy estimates and measured silicon energy. Broader discussion of significance and future work is provided in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results are formally stated in numbered environments and proven in the appendix. Proposition 1 (DFT diagonalization of the cyclic shift, Section 4) is proved in Appendix A.1. Corollary 1 (extension to real 2D rotation blocks) is proved in Appendix A.2. Corollary 2 (bounded, non-accumulating rounding error) is stated with its constants and proved in Appendix A.3. The scope of each result is stated explicitly. The correspondence to RoPE is structural rather than pointwise, and the rounding bound establishes non-accumulation with position rather than uniformly small error. Propositions and Corollaries cross-reference each other consistently.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The proposed method is fully described algorithmically in Section 4 (Eq. (6), the multi-frequency mapping in Eq. (8), and the 2D extension in Section 4.3). The backbones are public (Spikformer, Spikingformer, QKFormer, SpikeBERT), the datasets are public (Appendix B.1), and all training/optimization hyperparameters — LIF time constant τ, firing threshold V_{th}, batch size, learning rate, optimizer, embedding dimension, depth, time steps, and PE-specific parameters G and C — are listed in Table 6 (Appendix B.2). Reproduction requires only the published code of each backbone plus the per-group shift-count formula in Eq. (8).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is released to Github. All datasets used (Appendix B.1) are public benchmarks accessible from their original sources (Metr-la [9], Pems-bay [17], Solar / Electricity [14], MR / Subj [23, 2], SST-2 / SST-5 [29], CIFAR-10/100 [13], Tiny-ImageNet [4], CIFAR10-DVS [15], DVS128-Gesture [1]). The method (Section 4) and all hyperparameters (Appendix B.2) are fully specified, so the results can be reproduced by re-implementing the per-group cyclic shift on top of any of the listed open-source SNN-Transformer backbones.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 gives the per-task experimental framing (which backbones, which PE baselines, which datasets, prediction lengths for time-series). Appendix B provides full configuration: dataset descriptions (Appendix B.1) and Table 6 (Appendix B.2) listing common LIF parameters, training settings (batch size, learning rate, optimizer, embedding dimension, depth, time steps, attention heads, max sequence length), and PE-specific parameters (CPG-PE neuron count, BitShift-RoPE base C and group count G). All experiments use 3-seed averaging.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experimental results are reported as mean $\pm$ standard deviation over three independent seeds for every method we trained (Tables 3, 4, and 7).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide an exhaustive per-method energy breakdown in Section 5 (Efficiency Comparison) and Table 5 for the central efficiency claim: parameter count, FLOPs, SOPs, XNOR-OPs, and Horowitz-model energy in mJ/sample, all under identical conditions ($L = 168$, $D = 256$, $T = 4$, Spikformer + Electricity). For training, Appendix B.2 (Table 6) specifies model size (embedding dimension, depth, attention heads), batch size, and time-step count, from which compute requirements can be derived. Since the proposed method introduces no additional FLOPs or parameters at inference (Table 5), training compute is dominated by the backbone, not by the positional encoder.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is a foundational architectural contribution (a positional-encoding mechanism for SNN Transformers). It does not involve human subjects, scraped or sensitive data, or new dataset releases. All datasets used are public ML benchmarks, and all baselines and backbones are properly cited. We have reviewed the NeurIPS Code of Ethics and the work conforms to its requirements.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix I provides a Broader Impact Statement covering both positive impacts (energy-efficient on-device inference enabling low-power health monitoring, edge sensing, satellite forecasting, and brain-computer interfaces) and potential negative impacts (lower deployment cost can amplify always-on surveillance; reduced auditability of low-power edge inference; foundation-model-scale dual-use risks). The statement also notes that no human subjects or scraped data are involved, and that mitigation rests on standard SNN interpretability tooling (e.g., the Grad-CAM analysis in Appendix G).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work introduces an architectural module (a positional encoding for SNN Transformers), not a generative model, scraped dataset, or pre-trained large model with high misuse risk. No new model checkpoints or datasets are released. The contribution is a method that can be re-implemented from the formula in Eq. (8). Safeguards in the sense of release controls do not apply.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All baseline backbones (Spikformer [35], Spike-Driven Transformer [32], Spikingformer [33], QKFormer [34], SpikeBERT [18]) and PE baselines (CPG-PE [19], Gray-PE / Log-PE [20]) are cited at every point of use. All datasets are public benchmarks distributed under their own licenses (Appendix B.1). We cite the original publication of each. No proprietary assets are used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

Justification: This work does not release new datasets, model checkpoints, or other assets. The contribution is a positional-encoding mechanism specified algorithmically in Section 4 (Eq. (6), Eq. (8)) and Appendix B.2 (parameter values G, C).

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or research with human subjects. All experiments use only publicly available ML benchmark datasets (Appendix B.1).

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human subjects are involved in this work. IRB approval is therefore not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The proposed method (BitShift-RoPE) is a positional-encoding mechanism for SNN Transformers and does not use Large Language Models as any part of its core methodology. LLMs were used only for writing, editing, and translation support during manuscript preparation, which the policy notes does not require declaration.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.