SPIKEHENA: SPIKING HYBRID ENCODING AND NONLINEAR APPROXIMATION FOR TRAINING-FREE LLM-TO-SNN CONVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs) are emerging as a promising alternative to conventional Transformer-based large language models (LLMs) because of their spike-driven computation and sparse communication. However, existing ANN-to-SNN conversion methods face a difficult trade-off among representation fidelity, spike activity, and the required number of time steps, while key nonlinear operations such as Softmax, RMSNorm, and SiLU remain difficult to realize with signed binary spikes. These limitations hinder high-fidelity conversion of pretrained LLMs. In this work, we address the conversion problem from both spike encoding and nonlinear computation perspectives. For spike encoding, we propose a Two-Stage Hybrid Encoding Neuron that maintains a single membrane state across both stages, using population-activity-constrained BinarySpike encoding for large-magnitude activation components and SingleSpike encoding for the remaining residuals. For nonlinear computation, we design dedicated spiking units for square root, exponentiation, and division, and compose them into spike-driven implementations of Softmax, RMSNorm, and SiLU. Based on these designs, we develop SpikeHENA, a training-free conversion framework that directly reuses pretrained ANN parameters. Experiments on BERT and four LLMs show that, at $T = 44$, the converted LLMs incur only 0.12–0.64 percentage-point drops in average zero-shot accuracy and small perplexity increases on WikiText-2 and C4, while their theoretical synaptic-computation energy is estimated at 31.2%–32.5% of the corresponding ANN estimates.

1 INTRODUCTION

Large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; BigScience Workshop, 2024) based on Transformers (Vaswani et al., 2017) achieve strong performance across natural language processing tasks, but rely on costly dense computation and frequent memory access. Spiking neural networks (SNNs) (Maass, 1997; Gerstner et al., 2014) offer an alternative through spike-driven computation and sparse communication (Schuman et al., 2022). Their efficiency depends on spike representation, activity, time steps, and nonlinear-operation implementations.

Existing approaches to constructing SNNs can generally be categorized into direct training (Neftci et al., 2019; Wu et al., 2018) and ANN-to-SNN conversion (Bu et al., 2022; Li et al., 2021; Han et al., 2020; Hao et al., 2023b). Direct training of large-scale SNNs is costly because temporal unfolding increases training complexity, while discrete spike generation complicates optimization. In contrast, ANN-to-SNN conversion reuses pretrained ANN parameters and avoids retraining the entire model. However, converting LLMs remains challenging because their activations span wide dynamic ranges. Conventional spike encoding therefore involves a trade-off among representation fidelity, the required number of time steps, and spike activity. For example, SingleSpike encoding produces sparse spikes but may require many time steps to represent a sufficiently wide activation range (Hwang et al., 2024), whereas schemes with higher spike activity can improve representation fidelity at the cost of increased synaptic computation.

Another major challenge arises from nonlinear Transformer operations. Softmax, RMSNorm, and SiLU involve exponentiation, square root, and division, which are difficult to implement using

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

accumulation-based SNN computation. Existing conversion methods provide only partial support for these operations. SpikedAttention (Hwang et al., 2024) implements spike-based Softmax but retains conventional numerical computation for RMSNorm and SiLU, whereas SpikeZIP-TF (You et al., 2024) provides no dedicated spiking implementations of these nonlinearities. LAS (Chen et al., 2026b) approximates nonlinear functions using FewSpike neurons (Stöckl & Maass, 2021), which generate multi-valued rather than signed binary spikes. Wang et al. (2026) extend FewSpike neurons with multi-basis exponential decay (MBE) to approximate nonlinear operations without modifying pretrained ANN weights, and evaluate conversion on ViT, RoBERTa, and GPT-2. Consequently, existing methods do not jointly provide high-fidelity encoding over wide activation ranges, controlled spike activity within a moderate number of time steps, and unified support for key Transformer nonlinearities using signed binary spikes. Further related work is discussed in Appendix D.

To address these challenges, we propose SpikeHENA, a training-free framework for converting pretrained Transformer-based models into spike-driven SNNs. SpikeHENA introduces a Two-Stage Hybrid Encoding Neuron that continuously evolves one membrane state across two consecutive stages. Stage I employs population-activity-constrained BinarySpike encoding to represent large-magnitude activation components, whereas Stage II uses SingleSpike encoding to represent the remaining residuals. This hybrid design balances representation fidelity, spike activity, and the required number of time steps. We further design dedicated spiking arithmetic units for square root, exponentiation, and division and compose them into spike-driven implementations of Softmax, RMSNorm, and SiLU while preserving signed binary spike outputs. The resulting framework directly reuses pretrained ANN parameters without retraining, fine-tuning, or post-conversion optimization.

We evaluate SpikeHENA on BERT and four LLMs spanning different architectures and model scales. At $T = 16$, the converted BERT incurs an average GLUE (Wang et al., 2018) score drop of only 0.23 percentage points. At $T = 44$, the converted LLMs exhibit average zero-shot accuracy drops of 0.12–0.64 percentage points. Their corpus-level perplexity increases by only 0.0303–0.2192 on WikiText-2 (Merity et al., 2017) and 0.1271–0.3757 on C4 (Raffel et al., 2020). Meanwhile, their normalized theoretical synaptic-computation energy is estimated to be 31.2%–32.5% of the corresponding ANN estimates. These results demonstrate high conversion fidelity across model architectures and scales.

Our main contributions are summarized as follows:

- We propose a Two-Stage Hybrid Encoding Neuron that continuously evolves one membrane state across two stages. It combines population-activity-constrained BinarySpike encoding for large-magnitude components with SingleSpike encoding for residuals, balancing representation fidelity, spike activity, and time steps over the wide activation ranges of LLMs.
- We design dedicated spiking arithmetic units for square root, exponentiation, and division and compose them into spike-driven implementations of Softmax, RMSNorm, and SiLU while preserving signed binary spike outputs.
- We integrate the proposed encoding and nonlinear units into a training-free conversion framework that directly reuses pretrained ANN parameters, and validate it on BERT and four LLMs with scales of up to 72B parameters. The converted models preserve near-ANN task performance while substantially reducing the estimated synaptic-computation energy.

2 METHOD

2.1 SIGNED INTEGRATE-AND-FIRE DYNAMICS

In SNNs, integrate-and-fire neurons (Liu & Wang, 2001) are commonly used to convert real-valued signals into temporal spike sequences. In this work, we adopt a signed integrate-and-fire neuron, formulated as follows:

$$\widetilde{u}_i^t = u_i^{t-1} + \sum_j w_{ij} s_j^t, \tag{1}$$

$$s_i^t = H\big(\widetilde{u}_i^t - \vartheta^t\big) - H\big(-\widetilde{u}_i^t - \vartheta^t\big), s_i^t \in \{-1, 0, +1\}, \tag{2}$$

$$u_i^t = \widetilde{u}_i^t - s_i^t \vartheta^t. \tag{3}$$

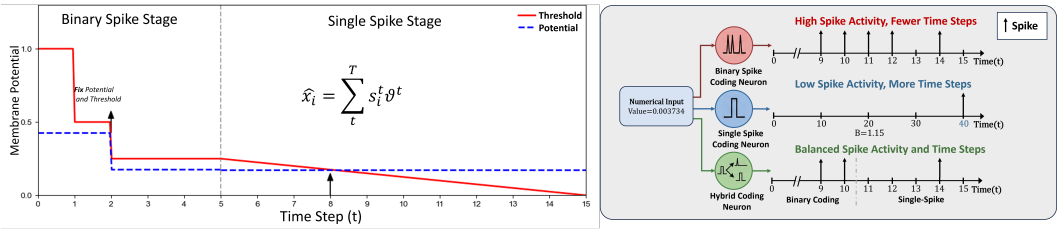


Figure 1: Illustration of the Two-Stage Hybrid Encoding Neuron and different spike encoding schemes. Left: Membrane-potential dynamics of the proposed neuron. Stage I performs population-activity-constrained BinarySpike encoding with exponentially decreasing thresholds. When accepting the candidate spikes would cause the cumulative population spike activity to exceed α, Stage I terminates, the residual membrane potential is retained, and the last threshold associated with accepted spikes is fixed as ϑ_f. Stage II continues from the same membrane state with linearly decreasing thresholds and allows each neuron to generate at most one signed residual spike. Right: Spike trains generated by BinarySpike, SingleSpike, and the proposed hybrid encoding to represent the same floating-point value.

Here, $u_i^t, \widetilde{u}_i^t$ denote the membrane potentials after and before firing at time step t, while w_{ij} and s_j^t denote the synaptic weight and presynaptic spike, respectively. The positive time-varying threshold $\vartheta^t > 0$ determines spike generation, $H(\cdot)$ denotes the Heaviside step function, and $s_i^t \in \{-1, 0, +1\}$ denotes the output spike. At each time step, synaptic inputs are first accumulated into the membrane potential, followed by spike generation and subtractive reset. Over T time steps, the signed spike sequence reconstructs the real-valued activation $\hat{x}_i$ through threshold-weighted accumulation:

$$\hat{x}_i = \sum_{t=1}^{T} s_i^t \vartheta^t. \tag{4}$$

The subtractive reset removes only the transmitted threshold-weighted spike contribution, thereby preserving residual information for subsequent time steps. The hybrid encoding neuron we introduce next retains these signed dynamics and the continuously evolving membrane state, while adopting different threshold schedules in two consecutive stages.

2.2 Two-Stage Hybrid Encoding Neuron

Existing studies have explored various spike encoding schemes for ANN-to-SNN conversion. However, as LLMs scale up, their activation ranges expand substantially, making a single threshold schedule inadequate for representing both large values and small residuals. BinarySpike covers a wide range with relatively few time steps, but its exponentially decreasing thresholds may cause high spike activity near the end. SingleSpike (Hwang et al., 2024) generates sparser spikes, yet fine-grained representation over a wide dynamic range generally requires substantially more time steps. IntegerSpike (Luo et al., 2024) and the FewSpike neurons (Stöckl & Maass, 2021) adopted by LAS (Chen et al., 2026b) represent large numerical ranges within a few time steps, but produce multi-valued spikes rather than the signed binary spikes considered here. These complementary trade-offs motivate our hybrid design. To combine the strengths of BinarySpike and SingleSpike while preserving signed binary outputs, we propose the Two-Stage Hybrid Encoding Neuron.

As illustrated in Figure 1, the two stages are connected through a continuously evolving membrane state: Stage I performs population-activity-constrained BinarySpike encoding, while Stage II encodes the remaining residual with at most one signed spike.

$$u_i^0 = x_i. \tag{5}$$

Throughout both stages, we follow the signed integrate-and-fire dynamics in Eqs. (2)–(3) for signed spike generation and subtractive soft reset. Information not transmitted by spikes remains in the residual membrane potential and is carried continuously across the two stages, which adopt different threshold schedules and spike constraints.

Stage I: Population-activity-constrained BinarySpike encoding. During the first N time steps, the firing threshold decreases exponentially:

$$\vartheta_{\mathrm{I}}^{t} = \vartheta_{\mathrm{I}}^{1} 2^{-(t-1)}, \qquad t = 1, \ldots, N. \tag{6}$$

This threshold schedule progressively discharges the large-magnitude component of the input. At each time step, candidate spikes $\hat{s}_i^t$ are evaluated before updating the membrane state. Let $\hat{\boldsymbol{s}}^t$ collect the candidate spikes across the N_{neu} neurons in the encoded population. We define the candidate cumulative population spike activity ρ^t as

$$\rho^t = \rho^{t-1} + \frac{1}{N_{\mathrm{neu}}} \left\| \hat{\boldsymbol{s}}^t \right\|_0, \qquad \rho^0 = 0. \tag{7}$$

The actual output s_i^t is determined by

$$s_i^t = \begin{cases} \hat{s}_i^t, & \rho^t \le \alpha, \\ 0, & \rho^t > \alpha. \end{cases} \tag{8}$$

If the candidate spikes satisfy the population-activity constraint, they are generated, and the membrane potential undergoes subtractive soft reset. If accepting them would cause the cumulative population spike activity to exceed the preset threshold α, they are rejected, and the neuron exits Stage I. The current residual membrane potential is retained without reset, and the last threshold associated with an accepted spike is fixed as ϑ_f. If the constraint is not violated within N time steps, ϑ_f is set to the final Stage I threshold $\vartheta_f = \vartheta_{\mathrm{I}}^{N}$. Any remaining positions in the N-step Stage I interval output zero spikes, preserving a fixed temporal layout.

Stage II: SingleSpike encoding. The same neuron continues with the residual membrane potential preserved at the stage transition; neither a new membrane state nor a separate residual encoder is introduced. Over the subsequent M time steps, the threshold decreases linearly from $\frac{M}{M+1}\vartheta_f$ to $\frac{\vartheta_f}{(M+1)}$:

$$\vartheta_{\mathrm{II}}^{m} = \frac{M+1-m}{M+1}\vartheta_f, \qquad m = 1, \ldots, M. \tag{9}$$

The neuron follows the same signed threshold comparison and subtractive soft-reset rule, while allowing each neuron to generate at most one residual spike. Once a nonzero spike is generated, the neuron outputs zero spikes for the remaining Stage II time steps. Consequently, the spike sign indicates the direction of the residual, while the spike time selects a level from the linearly spaced threshold sequence. If no threshold is crossed, the residual magnitude remains below the final threshold.

The proposed neuron therefore generates a single $(N+M)$-step spike sequence through the continuous evolution of one membrane potential:

$$\mathbf{s}_i = \left(s_i^1, \ldots, s_i^{N+M}\right), \qquad s_i^t \in \{-1, 0, +1\}, \tag{10}$$

We reconstruct the represented value by accumulating all generated spikes weighted by their corresponding thresholds:

$$\hat{x}_i = \sum_{t=1}^{N} s_i^t \vartheta_{\mathrm{I}}^t + \sum_{m=1}^{M} s_i^{N+m} \vartheta_{\mathrm{II}}^m. \tag{11}$$

2.3 Nonlinear Function Approximation

Nonlinear operations such as division, square root, and exponentiation are difficult to implement using the simple accumulation operations inherent to SNNs, posing a key challenge to constructing spike-driven Transformer models. To address this challenge, we propose dedicated spiking units that approximate these operations through computation schemes tailored to spiking dynamics.

Division Neuron. We propose a division neuron based on a spike-triggered mechanism, where a positive denominator membrane potential B_i determines the firing threshold. The neuron accepts either a spike-coded numerator or an accumulated numerical numerator. For a spike-coded numerator, the membrane state is initialized from the first nonzero spike's weighted contribution and integrates

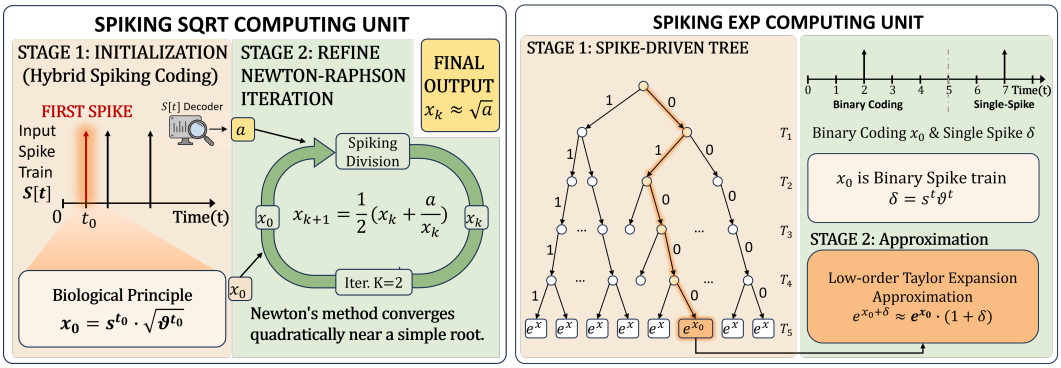


Figure 2: Overview of the proposed square-root unit (left) and exponentiation unit (right). The five-level lookup tree is a truncated illustration of the temporal traversal rather than the complete lookup structure.

subsequent weighted spike inputs. For a numerical numerator A_i accumulated from spikes, the membrane state is initialized directly as $u_i^0 = A_i$. The denominator is scaled by an integer shift to define the time-dependent threshold:

$$\vartheta_{\mathrm{div},i}^t = B_i \ll (r - t + 1), \qquad t = 1, \ldots, T. \tag{12}$$

where the integer r aligns the threshold scale with the numerator and denominator magnitudes. For a spike-coded numerator, the scale is determined from the first nonzero spike's temporal weight and the denominator magnitude; for a numerical numerator, it is determined directly from the operand magnitudes. The result sign is represented by the signed spike.

At each time step, we compare u_i^t with $\vartheta_{\mathrm{div},i}^t$ to generate $s_i^t \in \{-1, 0, +1\}$, while following the subtractive soft-reset rule of the signed integrate-and-fire dynamics. Each spike subtracts the corresponding signed threshold from the numerator state, and the quotient is decoded by accumulating the associated binary weights:

$$\hat{x}_i = \sum_{t=1}^{T} s_i^t 2^{r-t+1}. \tag{13}$$

For notational simplicity, division is written in conventional floating-point form in other equations, with accumulated floating-point values as numerators. Both input forms are processed by the same threshold-comparison and subtractive-reset mechanism, and the quotient is represented by signed binary spikes.

Multiplication and Square. We implement multiplication using the differential recurrence of SpikeZIP-TF (You et al., 2024), which updates the product state from increments in the accumulated operands. Squaring is obtained by using the same input for both operands.

Square Root Function. Inspired by first-spike coding in biological neurons (Gerstner & Kistler, 2002), the square-root unit uses the first nonzero spike to initialize the numerical iteration. Given a signed spike sequence s^t whose accumulation represents a nonnegative operand a, we define the first spike time and initial estimate as

$$t_0 = \min\{t \mid s^t \neq 0\}, \qquad x_0 = s^{t_0}\sqrt{\vartheta^{t_0}}. \tag{14}$$

Since the square-root operand is nonnegative, the first nonzero spike used for initialization is positive. We obtain the initial estimate through a lookup table. If no spike is generated over the entire sequence, we treat the input as zero and set $x_0 = 0$. We then apply the Newton–Raphson method to update the estimate:

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right), \qquad k = 0, 1, \tag{15}$$

where a denotes the accumulated nonnegative operand, and each quotient a/x_k is computed by the division neuron. We perform two Newton–Raphson updates to refine the first-spike estimate. Thus, the square-root unit combines temporal initialization with a short numerical refinement.

Exponentiation Function. For exponentiation, we clip the input to $x \in [-10, 10]$. The two consecutive stages of our neuron naturally decompose the represented value into

$$x = x_0 + \delta, \tag{16}$$

where the Stage I output forms the coarse component x_0, while the Stage II output represents the residual component δ. We retrieve the stored value of e^{x_0} using a temporal tree. The Stage I spike sequence guides the temporal traversal, and the lookup value is obtained once the coarse component is resolved. We then incorporate the residual contribution using the first-order approximation

$$e^{x_0+\delta} \approx e^{x_0}(1 + \delta). \tag{17}$$

Since δ is the residual retained after Stage I, we use it as a local correction to the coarse lookup value. Among the N conceptual threshold planes, spike planes associated with larger thresholds remain zero under the restricted input range and are therefore skipped during lookup. Consequently, we traverse only the remaining spike planes with smaller thresholds. We then use the Stage II output to compute the correction term $e^{x_0}\delta$.

Figure 2 summarizes the proposed square-root and exponentiation units. The former combines first-spike initialization with Newton–Raphson refinement, while the latter employs temporal-tree lookup for the coarse component and a first-order residual correction. Appendix C provides the encoding error bound for the Two-Stage Hybrid Encoding Neuron and approximation error analyses for the nonlinear units.

2.4 Spike-Driven Nonlinear Layers

Using spiking arithmetic units for division, exponentiation, and square root, we construct spike-driven nonlinear layers that efficiently implement Softmax, RMSNorm, and SiLU, which are commonly used in Transformer-based LLMs.

Spiking Softmax. Direct exponentiation of large logits may amplify approximation errors and expand the numerical range. We therefore center each input by subtracting the maximum logit $m = \max_j x_j$ before applying the exponentiation unit. We compute the normalized output as

$$\mathrm{Softmax}(x_i) = \frac{\exp(x_i - m)}{\sum_j \exp(x_j - m)}. \tag{18}$$

We first process the centered logits with the exponentiation unit and sum the resulting values across channels to obtain the denominator. The division neuron then normalizes each channel.

Spiking RMSNorm. For a real-valued input vector $\boldsymbol{x}$ of length n, we use the multiplication unit to square each element and accumulate the results to obtain the mean-square value. The square-root unit then approximates the denominator, and the division neuron performs normalization:

$$\mathrm{RMSNorm}(\boldsymbol{x}) = \frac{\boldsymbol{\gamma} \odot \boldsymbol{x}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2 + \epsilon}}, \tag{19}$$

where ϵ is a small constant that ensures numerical stability, and $\boldsymbol{\gamma}$ is the element-wise scale parameter inherited from the pretrained model.

Spiking SiLU. We evaluate SiLU using the identity

$$\mathrm{SiLU}(x) = \frac{x}{1 + \exp(-x)}. \tag{20}$$

We first pass the negated input to the exponentiation unit to approximate $\exp(-x)$, add one to the result, and then use the division neuron to divide x by the resulting denominator.

3 Experiments

3.1 Experimental Setup

We evaluate SpikeHENA on BERT (Devlin et al., 2019), Qwen1.5-MoE-A2.7B (Team, 2024), DeepSeek-V2-Lite (Liu et al., 2024), Qwen3-30B-A3B (Yang et al., 2025), and Qwen2.5-72B-Instruct-AWQ (Ahmed et al., 2025). Using OpenCompass (Contributors, 2023), we directly convert pretrained models without retraining, fine-tuning, or post-conversion optimization. We evaluate

Table 1: Comparison of BERT-based ANN and SNN models on the GLUE benchmark. MRPC reports accuracy/F1, and STS-B reports Pearson/Spearman correlations. Avg. is computed by first averaging the two metrics for MRPC and STS-B and then averaging across the seven tasks. Δ denotes the difference in average score relative to the ANN baseline. † indicates additional time steps used to collect prior information.

Method	SNN	T	QQP	MNLI-m	SST-2	QNLI	RTE	MRPC	STS-B	Avg.	Δ
BERT (Devlin et al., 2019)	×	1	90.83	83.94	92.66	90.99	64.98	84.56/89.19	88.70/88.48	85.55	–
SPR (Hao et al., 2023a)	✓	$8(16^{\dagger})$	87.48	77.56	90.48	87.75	64.98	78.68/85.76	86.71/86.50	82.44	-3.11
QCFS (Bu et al., 2022)	✓	8	88.42	79.57	89.91	86.80	56.68	78.92/85.37	86.18/85.82	81.36	-4.19
COS (Hao et al., 2023b)	✓	$8(8^{\dagger})$	88.85	79.91	89.79	87.37	63.18	79.66/86.33	86.49/86.23	82.64	-2.91
FAS (Chen et al., 2026a)	✓	4	90.38	82.77	91.17	90.13	66.06	86.02/90.22	87.46/87.26	85.14	-0.41
AIF (Cao et al., 2026)	✓	16	–	83.7	92.1	89.8	63.2	–/90.3	–	–	–
LAS (Chen et al., 2026b)	✓	16	90.63	83.58	92.09	90.65	63.90	83.82/88.70	88.64/88.43	85.09	-0.46
Ours$_{N=10,M=6}$	✓	16	90.52	83.88	**92.78**	90.10	65.34	84.07/88.07	88.68/88.48	**85.32**	**-0.23**

Table 2: Comparison of zero-shot accuracy and normalized theoretical synaptic-computation energy between ANN baselines and converted SNNs across four LLMs. Avg. is the mean over six tasks, Δ is the difference from the corresponding ANN, and $E_{\mathrm{syn}}/E_{\mathrm{ANN}}$ is the 45-nm theoretical synaptic-computation energy normalized to that of the corresponding ANN.

Model	T	HellaSwag	PIQA	Wino.	ARC-c	ARC-e	BoolQ	Avg.	Δ	$E_{\mathrm{syn}}/E_{\mathrm{ANN}}$
Qwen1.5-MoE-A2.7B (Team, 2024)	1	71.52	80.25	63.77	37.63	58.02	70.67	63.64	–	1.000
Ours$_{N=24,M=20}$	44	70.53	79.05	63.14	37.97	56.44	70.86	63.00	-0.64	0.312
DeepSeek-V2-Lite (Liu et al., 2024)	1	72.90	80.03	67.40	37.97	66.31	57.95	63.76	–	1.000
Ours$_{N=24,M=20}$	44	72.64	79.98	67.64	37.29	67.02	56.97	63.59	-0.17	0.312
Qwen3-30B-A3B (Yang et al., 2025)	1	70.92	79.98	67.64	42.37	55.56	64.31	63.46	–	1.000
Ours$_{N=24,M=20}$	44	70.31	80.30	67.17	44.07	53.09	65.08	63.34	-0.12	0.325
Qwen2.5-72B-Instruct-AWQ (Ahmed et al., 2025)	1	80.58	82.86	74.59	48.47	67.90	89.17	73.93	–	1.000
Ours$_{N=24,M=20}$	44	78.38	82.59	76.16	48.47	67.72	88.35	73.61	-0.32	0.325

BERT on the GLUE benchmark (Wang et al., 2018), and the LLMs are evaluated zero-shot on HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2021), ARC-Challenge (Clark et al., 2018), ARC-Easy (Clark et al., 2018), and BoolQ (Clark et al., 2019), with corpus-level perplexity reported on WikiText-2 (Merity et al., 2017) and C4 (Raffel et al., 2020). Here, N and M denote the numbers of BinarySpike and SingleSpike time steps, respectively, with $T = N + M$. The total T denotes the number of SNN encoding time steps rather than measured end-to-end inference latency. Unless otherwise specified, the LLM experiments use $N = 24$, $M = 20$, and $\alpha = 0.9$. Detailed parameter sweeps are provided in Appendix B.

3.2 Main Results

Experiments on GLUE Tasks. With $T = 16$, SpikeHENA achieves an average GLUE score of 85.32%, corresponding to $\Delta = -0.23$ relative to the ANN and outperforming LAS (Chen et al., 2026b) under the same time-step setting (85.09%). SpikeHENA also achieves 92.78% on SST-2, slightly exceeding the ANN score of 92.66%. The small average gap reported in Table 1 shows that our conversion largely preserves BERT's language-understanding capability without task-specific adaptation.

Experiments on Zero-Shot Tasks. Table 2 evaluates conversion fidelity against each source ANN on four LLMs at $T = 44$. Cross-method comparisons use BERT/GLUE because existing LLM conversion studies differ in architectures, implementations, and tasks. Average zero-shot accuracy drops by 0.12–0.64 percentage points, with theoretical synaptic-computation energy ratios of 0.312–0.325. Qwen3-30B-A3B (Yang et al., 2025) improves ARC-Challenge accuracy by 1.70 percentage points. These results indicate high conversion fidelity across architectures and scales; the energy ratios are theoretical, not measured.

Experiments on Language Modeling. We evaluate corpus-level PPL on WikiText-2 (Merity et al., 2017) and C4 (Raffel et al., 2020). As shown in Table 3, across the four models, conversion with

Table 3: Comparison of corpus-level perplexity between ANN baselines and converted SNNs on WikiText-2 (W2) and C4. Qwen2.5-72B denotes Qwen2.5-72B-Instruct-AWQ.

Model	T	W2	C4
Qwen1.5-MoE-A2.7B (Team, 2024)	1	6.3944	10.0494
Ours$_{N=24,M=20}$	44	6.6136	10.4251
DeepSeek-V2-Lite (Liu et al., 2024)	1	5.4964	9.3362
Ours$_{N=24,M=20}$	44	5.5925	9.5147
Qwen3-30B-A3B (Yang et al., 2025)	1	7.6736	14.0369
Ours$_{N=24,M=20}$	44	7.7039	14.1640
Qwen2.5-72B (Ahmed et al., 2025)	1	3.8936	10.0903
Ours$_{N=24,M=20}$	44	4.0624	10.3971

Table 4: Theoretical computation-energy breakdown for BERT (mJ/sample). Syn. includes synaptic MAC/AC computation and FP32 boundary operations; N&E, neuron updates, encoding, and candidate spike-activity control; LN, LayerNorm; Sq., square; and Oth., auxiliary attention and Softmax operations, attention-state and mask control, pooler TANH, and residual additions. Rel. and Red. denote total energy normalized to the ANN and the corresponding reduction, respectively. LN excludes the separately reported Sq. and SQRT terms.

	Theoretical computation energy (mJ/sample)										*Normalized energy*	
Model	Syn.	N&E	GELU	LN	Sq.	SQRT	EXP	DIV	Oth.	Total	Rel.	Red. (%)
BERT (Devlin et al., 2019)	51.4014	–	1.5194	0.0201	0.0091	0.0002	0.0181	0.0109	0.0233	53.0024	1.000	–
Ours$_{N=10,M=6}$	15.1406	1.7000	0.0783	0.0184	0.0278	0.0001	0.1508	0.4396	0.0402	**17.5960**	**0.3320**	**66.80**

$T = 44$ increases PPL by only 0.0303–0.2192 on WikiText-2 and 0.1271–0.3757 on C4. These consistently small increases indicate that SpikeHENA largely preserves the language-modeling capability of the corresponding ANNs across different model architectures and scales.

3.3 BERT Theoretical Computation-Energy Analysis

Following prior conversion studies (Hwang et al., 2024; Wang et al., 2026), we multiply trace-derived operation counts by 45-nm unit-energy costs (Horowitz, 2014). Traces use batch size 32 and sequence length 128, micro-averaged across GLUE tasks. Unlike the LLM synaptic-only estimates in Table 2, the BERT breakdown in Table 4 includes synaptic computation, FP32 boundary operations, neuron updates, hybrid encoding and activity control, nonlinear operations, and specified local-access and LUT-read costs. Appendix A details the counting rules and unit-energy assumptions.

With $N = 10$, $M = 6$, and $\alpha = 0.9$, the theoretical computation energy of SpikeHENA is $17.5960\,\mathrm{mJ/sample}$, compared with $53.0024\,\mathrm{mJ/sample}$ for the ANN, corresponding to 33.20% of the ANN estimate and a reduction of 66.80%. General memory transfers, routing, synchronization, clock-tree power, and leakage are excluded from this computation-energy estimate. Appendix A separately reports a memory-augmented theoretical estimate for BERT under explicit SRAM, DRAM, routing, and synchronization assumptions. All reported energy values are theoretical estimates under the stated models, rather than hardware measurements.

3.4 Ablation Studies

Ablation Study on Hybrid Encoding. We compare hybrid encoding with BinarySpike and SingleSpike on DeepSeek-V2-Lite (Liu et al., 2024) and Qwen1.5-MoE-A2.7B (Team, 2024). Under the same 44-step setting, hybrid encoding maintains low diagnostic PPL while requiring substantially lower normalized theoretical synaptic-computation energy than BinarySpike. SingleSpike produces lower spike activity but exhibits severe PPL degradation at 44 time steps and requires 120 time steps to approach the ANN PPL. The results in Table 5 show that combining BinarySpike and SingleSpike

Table 5: Ablation of spike encoding schemes on a balanced subset sampled equally from the six zero-shot datasets in Table 2. Gold PPL denotes gold-choice diagnostic perplexity, B denotes the base of SingleSpike encoding, and AC Act denotes the number of spike-triggered accumulation operations per encoded tensor element. $E_{\text{syn}}/E_{\text{ANN}}$ denotes the theoretical synaptic-computation energy normalized to that of the corresponding ANN.

Encoding	T	Parameters	DeepSeek-V2-Lite (Liu et al., 2024)			Qwen1.5-MoE-A2.7B (Team, 2024)		
			Gold PPL ↓	AC Act. ↓	$E_{\text{syn}}/E_{\text{ANN}}$ ↓	Gold PPL ↓	AC Act. ↓	$E_{\text{syn}}/E_{\text{ANN}}$ ↓
ANN	1	–	3.032510	–	1.000	3.338293	–	1.000
BinarySpike	16	$N = 16$	3.231372	2.779	0.544	5.630653	2.910	0.569
BinarySpike	24	$N = 24$	3.032100	4.701	0.920	3.346200	4.066	0.796
BinarySpike	44	$N = 44$	3.032996	5.821	1.139	3.346382	4.332	0.847
SingleSpike	44	$B = 1.10,\ M = 44$	11.170068	0.186	0.036	13.257139	0.325	0.064
SingleSpike	80	$B = 1.10,\ M = 80$	8.694300	0.749	0.147	5.951800	0.869	0.170
SingleSpike	120	$B = 1.10,\ M = 120$	3.098265	0.993	0.194	3.390174	0.995	0.195
Ours$_{N=24,M=20}$	44	$N = 24,\ M = 20,\ \alpha = 0.9$	**3.027800**	1.604	0.314	**3.400300**	1.616	0.316

Table 6: Ablation of spike-based nonlinear approximations on a balanced subset sampled equally from the six zero-shot datasets in Table 2. Each (+) configuration adds one spike-based nonlinear unit to Encoding-only, whereas each w/o configuration restores the corresponding ANN operation in Full. Δ is measured relative to Encoding-only for (+) and Full, and relative to Full for w/o; positive values indicate higher PPL.

Configuration	DeepSeek-V2-Lite		Qwen1.5-MoE-A2.7B	
	PPL ↓	Δ (%)	PPL ↓	Δ (%)
Encoding-only	3.025884	–	3.373119	–
+Softmax	3.025311	-0.019	3.434351	+1.815
+RMSNorm	3.026024	+0.005	3.356426	-0.495
+SiLU	3.026492	+0.020	3.372138	-0.029
Full	**3.021393**	**-0.148**	3.406865	+1.000
w/o Softmax	3.026493	+0.169	3.379800	-0.794
w/o RMSNorm	3.023713	+0.077	3.427952	+0.619
w/o SiLU	3.026670	+0.175	3.410058	+0.094

stages achieves a better balance among representation fidelity, spike activity, and the number of time steps than either scheme alone.

Ablation Study on Nonlinear Approximation. We evaluate the spike-based approximations of Softmax, RMSNorm, and SiLU. Compared with Encoding-only, which retains the ANN nonlinear operations, Full changes the diagnostic PPL by -0.148% on DeepSeek-V2-Lite and $+1.000\%$ on Qwen1.5-MoE-A2.7B, as shown in Table 6. The leave-one-out results in Table 6 show architecture-dependent changes: restoring ANN Softmax or SiLU increases diagnostic PPL on DeepSeek-V2-Lite, while restoring ANN RMSNorm increases it on Qwen1.5-MoE-A2.7B. Overall, replacing all three nonlinear units preserves near-ANN behavior, although their effects vary across architectures.

4 Conclusion

This paper proposes SpikeHENA, a training-free framework for converting pretrained Transformers into spike-driven SNNs. Our Two-Stage Hybrid Encoding Neuron combines population-activity-constrained BinarySpike encoding with SingleSpike residual encoding, balancing representation fidelity, spike activity, and time steps. Dedicated spiking arithmetic units further enable spike-driven implementations of Softmax, RMSNorm, and SiLU with signed binary spikes. Experiments on BERT and four LLMs up to 72B demonstrate high conversion fidelity. At $T = 44$, the converted LLMs show only 0.12–0.64 percentage-point drops in average zero-shot accuracy and small PPL increases, while requiring 31.2%–32.5% of the corresponding ANN theoretical synaptic-computation energy.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and LaTeX format conversion of this paper. In particular, OpenAI Codex assisted with converting the existing manuscript and technical supplement from the AAAI format to the ICLR 2027 format, including document structure, table formatting, citations, and cross-references. The authors are responsible for the final content of the paper, including its text, claims, and artifacts.

REFERENCES

Imtiaz Ahmed, Sadman Islam, Partha Protim Datta, Imran Kabir, Naseef Ur Rahman Chowdhury, and Ahshanul Haque. Qwen 2.5: A comprehensive review of the leading resource-efficient LLM with potentioal to surpass all competitors. *Authorea Preprints*, 2025.

BigScience Workshop. BLOOM: A 176B-parameter open-access multilingual language model. *Journal of Machine Learning Research*, 25(422):1–74, 2024. URL https://jmlr.org/papers/v25/23-0581.html.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7B3IJMM1k_M.

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

Long Chen, Xiaotian Song, Andy Song, and Yanan Sun. FAS: Fast ANN-SNN conversion for spiking multimodal large language models. In *2026 IEEE International Conference on Multimedia and Expo*, 2026a.

Long Chen, Xiaotian Song, and Yanan Sun. Las: Loss-less ANN-SNN conversion for fully spike-driven large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1730–1738, 2026b.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=FZ1oTwcXchK.

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, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. Rmp-SNN: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13558–13567, 2020.

Zecheng Hao, Tong Bu, Jianhao Ding, Tiejun Huang, and Zhaofei Yu. Reducing ANN-SNN conversion error through residual membrane potential. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11–21, 2023a. doi: 10.1609/aaai.v37i1.25071.

Zecheng Hao, Jianhao Ding, Tong Bu, Tiejun Huang, and Zhaofei Yu. Bridging the gap between ANNs and SNNs by calibrating offset spikes. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=PFbzoWZyZRX.

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)*, pp. 10–14. IEEE, 2014. doi: 10.1109/ISSCC.2014.6757323.

Sangwoo Hwang and Jaeha Kung. One-spike SNN: Single-spike phase coding with base manipulation for ANN-to-SNN conversion loss minimization. *IEEE Transactions on Emerging Topics in Computing*, 13(1):162–172, 2025. doi: 10.1109/TETC.2024.3386893.

Sangwoo Hwang, Seunghyun Lee, Dahoon Park, Donghun Lee, and Jaeha Kung. Spikedattention: Training-free and fully spike-driven transformer-to-SNN conversion with winner-oriented spike shift for softmax operation. *Advances in Neural Information Processing Systems*, 37:67422–67445, 2024.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration. In *International conference on machine learning*, pp. 6316–6325. PMLR, 2021.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024.

Ying-Hui Liu and Xiao-Jing Wang. Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron. *Journal of computational neuroscience*, 10(1):25–45, 2001.

Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *European Conference on Computer Vision*, pp. 253–272. Springer, 2024.

Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

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

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Bodo Rueckauer and Shih-Chii Liu. Conversion of analog to spiking neural networks using sparse temporal coding. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2018. doi: 10.1109/ISCAS.2018.8351295.

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

Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Prasanna Date, and Bill Kay. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022.

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

Qwen Team. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters, February 2024. URL https://qwenlm.github.io/blog/qwen-moe/.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Jingya Wang, Xin Deng, Wenjie Wei, Dehao Zhang, Shuai Wang, Qian Sun, Jieyuan Zhang, Hanwen Liu, Ning Xie, and Malu Zhang. Training-free ANN-to-SNN conversion for high-performance spiking transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2128–2136, 2026. doi: 10.1609/aaai.v40i3.37195. URL https://ojs.aaai.org/index.php/AAAI/article/view/37195.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

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

Kang You, Zekai Xu, Chen Nie, Zhijie Deng, Qinghai Guo, Xiang Wang, and Zhezhi He. SpikeZIP-TF: Conversion is all you need for transformer-based SNN. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57367–57383. PMLR, 2024. URL https://proceedings.mlr.press/v235/you24b.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

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

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 QK attention. *Advances in Neural Information Processing Systems*, 37:13074–13098, 2024.

Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Han Zhang, Jiaqi Wang, Huihui Zhou, Zhengyu Ma, and Yonghong Tian. Spikingformer: A key foundation model for spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2236–2244, 2026.

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

Table A: Unit energies used in the theoretical BERT analysis. Derived values follow the published 45-nm FP32 and SRAM costs. Hardware-dependent COMPARE, SHIFT, LUT, NoC, and synchronization costs are conservatively modeled.

Operation	Unit energy	Basis
FP32 MAC	4.6 pJ	ADD + MUL
FP32 MUL	3.7 pJ	Published
FP32 ADD/SUB	0.9 pJ	Published
INT32 ADD	0.1 pJ	Published
Synaptic AC	0.9 pJ	Derived
COMPARE/SHIFT	0.9 pJ	Conservative
EXP/TANH	92 pJ	20 MACs
SQRT/DIV	55.2 pJ	12 MACs
GELU	322 pJ	70 MACs
32-bit SRAM access	5 pJ	Published
1-bit SRAM access	0.15625 pJ	Derived
2-bit signed-spike access	0.3125 pJ	Derived
32-bit DRAM access	640 pJ	Published
32-bit LUT read	5 pJ	SRAM-mapped
32-bit NoC flit-hop	3.5 pJ	Mapping parameter
Synchronization barrier	1000 pJ	Mapping parameter

This supplementary material provides a detailed theoretical energy analysis, complete ablation experiments across encoding configurations and time-step allocations, and deterministic local error bounds for SpikeHENA's hybrid encoding and spike-based nonlinear units.

A Energy Analysis

A.1 Evaluation Setup and Energy Model

We provide a component-level theoretical computation-energy analysis for BERT inference. Operation counts are collected from actual inference with a batch size of 32 and a sequence length of 128. We run one batch for each of the seven GLUE tasks and report the sample-level micro-average. The SNN uses $N = 10$ BinarySpike steps, $M = 6$ SingleSpike residual steps, $T = 16$, and $\alpha = 0.9$. The reported energy is computed as

$$E_{\text{total}} = \sum_i N_i e_i, \tag{A.1}$$

where N_i denotes the trace-derived count of type-(i) operations, and e_i denotes the corresponding unit energy. We distinguish LLM synaptic-computation energy, BERT component-level computation energy, and a memory-augmented BERT estimate under explicit mapping assumptions. The candidate-control term uses the analytical bound derived below. All reported energies are theoretical estimates under the stated operation and access models.

Table A summarizes the adopted 45-nm energy model. Following prior hardware-aware analyses (Hwang et al., 2024; Horowitz, 2014), FP32 addition and multiplication consume 0.9 and 3.7 pJ, respectively. Accordingly, the energy of an FP32 MAC is 4.6 pJ. A spike-triggered accumulation consumes 0.9 pJ. Comparisons and 32-bit shifts are conservatively assigned the same energy as an addition. The energy of a 1-bit access is obtained by linearly scaling that of a 32-bit SRAM access, and a signed spike is represented by one spike bit and one sign bit.

A.2 LLM Synaptic-Computation Energy

For the LLM results in Table 2, we estimate synaptic-computation energy using the same 45-nm MAC/AC unit costs. The normalized ratio is

$$\frac{E_{\text{syn}}^{\text{SNN}}}{E_{\text{syn}}^{\text{ANN}}} = \frac{N_{\text{AC}}^{\text{SNN}} e_{\text{AC}}}{N_{\text{MAC}}^{\text{ANN}} e_{\text{MAC}}}, \qquad e_{\text{AC}} = 0.9\,\text{pJ}, \quad e_{\text{MAC}} = 4.6\,\text{pJ}. \tag{A.2}$$

With $\alpha = 0.9$, the measured AC activity ranges from 1.59 to 1.66 spike-triggered accumulations per encoded tensor element, giving normalized theoretical synaptic-computation energies of 0.312–0.325. These ratios cover synaptic MAC/AC computation; neuron updates, encoding and activity control, nonlinear operations, and general memory and communication costs are outside this scope. The following component-level and memory-augmented estimates are specific to BERT and are reported separately from the LLM synaptic-computation results.

A.3 BERT Synaptic Computation

The ANN performs

$$N_{\mathrm{MAC}}^{\mathrm{ANN}} = 11.174217216\ \mathrm{B} \tag{A.3}$$

MAC operations per sample. Accordingly, its theoretical synaptic-computation energy is

$$\begin{aligned} E_{\mathrm{syn}}^{\mathrm{ANN}} &= 11.174217216\mathrm{B} \times 4.6\ \mathrm{pJ} \\ &= 51.401399\ \mathrm{mJ/sample}. \end{aligned} \tag{A.4}$$

For SpikeHENA, the measured AC activity of the hybrid encoding is 1.509 spike-triggered accumulations per encoded tensor element. The linear layers involve 10.871635968 billion dense MAC operations, while the spike-driven QK/AV computation involves 0.414600981 billion SAA operations. Thus,

$$\begin{aligned} N_{\mathrm{SOP}}^{\mathrm{SNN}} &= 1.509 \times 10.871635968\mathrm{B} + 0.414600981\mathrm{B} \\ &= 16.819899656\mathrm{B}. \end{aligned} \tag{A.5}$$

The SAA count includes the three spike-driven attention terms $\Delta A \Delta B$, $\Delta A S_B$, and $S_A \Delta B$. The pooler and classifier additionally perform 591,360 FP32 boundary MAC operations. Consequently,

$$\begin{aligned} E_{\mathrm{syn}}^{\mathrm{SNN}} &= 16.819899656\mathrm{B} \times 0.9\ \mathrm{pJ} \\ &\quad + 591{,}360 \times 4.6\ \mathrm{pJ} \\ &= 15.140630\ \mathrm{mJ/sample}. \end{aligned} \tag{A.6}$$

The resulting reduction in theoretical MAC/AC synaptic-computation energy is

$$1 - \frac{E_{\mathrm{syn}}^{\mathrm{SNN}}}{E_{\mathrm{syn}}^{\mathrm{ANN}}} = 70.54\%. \tag{A.7}$$

A.4 Neuron and Hybrid-Encoding Arithmetic

The first $N = 10$ temporal coefficients are powers of two and are implemented using bit shifts, whereas only the $M = 6$ residual coefficients require FP32 multiplications. These coefficients form a short fixed sequence shared across each tensor and can be stored in registers. Thus, the synaptic weights require neither runtime modification nor T temporally pre-folded copies.

The measured counts of neuron slots, temporal shifts, and residual multiplications are

$$\begin{aligned} N_{\mathrm{slot}} &= 566.329344\ \mathrm{M}, \\ N_{\mathrm{shift}} &= 137.379154\ \mathrm{M}, \\ N_{\text{res-mul}} &= 11.367887\ \mathrm{M}. \end{aligned} \tag{A.8}$$

Each neuron slot performs one membrane accumulation, one threshold comparison, and one conditional subtractive reset. The resulting theoretical arithmetic energy is

$$\begin{aligned} E_{\mathrm{neu/enc}} &= 566.329344\mathrm{M} \times (0.9 + 0.9 + 0.9)\ \mathrm{pJ} \\ &\quad + 137.379154\mathrm{M} \times 0.9\ \mathrm{pJ} \\ &\quad + 11.367887\mathrm{M} \times 3.7\ \mathrm{pJ} \\ &= 1.694792\ \mathrm{mJ/sample}. \end{aligned} \tag{A.9}$$

A.5 Candidate Spike-Activity Control Overhead

We account for the overhead of evaluating the cumulative candidate population spike activity. Each signed spike is represented by one spike bit, $b_i^t \in \{0,1\}$, and one sign bit, $q_i^t \in \{0,1\}$:

$$s_i^t = (-1)^{q_i^t} b_i^t. \tag{A.10}$$

Its ℓ_0 norm depends only on the spike bits:

$$\left\|\boldsymbol{s}^t\right\|_0 = \sum_{i=1}^{D_l} \mathbf{1}\left[s_i^t \neq 0\right] = \sum_{i=1}^{D_l} b_i^t, \tag{A.11}$$

where D_l is the number of neurons in layer l. Therefore, computing the population spike activity does not require accessing the sign bits.

Rather than explicitly updating the normalized FP32 activity,

$$\rho^t = \frac{1}{D_l} \sum_{\tau=1}^{t} \left\|\boldsymbol{s}^\tau\right\|_0, \tag{A.12}$$

we maintain an INT32 counter C^{t-1} for the cumulative number of accepted spikes and compute the candidate count as

$$\widetilde{C}^t = C^{t-1} + \left\|\boldsymbol{s}^t\right\|_0. \tag{A.13}$$

The corresponding integer activity budget is precomputed as

$$B_l = \lfloor \alpha D_l \rfloor. \tag{A.14}$$

The candidate spikes are accepted if

$$\widetilde{C}^t \leq B_l, \tag{A.15}$$

in which case $C^t = \widetilde{C}^t$. Otherwise, the candidate spikes are rejected and the encoding proceeds to Stage II. This implementation avoids per-step FP32 activity updates and FP32 divisions. With a streaming counter that terminates as soon as the budget is exceeded, it requires one INT32 counter increment per counted nonzero candidate spike and at most one budget comparison per Stage I time step.

For BERT, the trace contains

$$N_{\text{MAC-slot}} = 226{,}492{,}416 \tag{A.16}$$

Stage I neuron–time slots. Because Stage I uses $N = 10$ time steps, the sample-level number of encoded neuron elements is

$$\begin{aligned} D_{\text{enc}} &= \frac{N_{\text{MAC-slot}}}{N} \\ &= \frac{226{,}492{,}416}{10} \\ &= 22{,}649{,}241.6. \end{aligned} \tag{A.17}$$

With $\alpha = 0.9$, the cumulative number of accepted nonzero spikes is bounded by

$$\begin{aligned} K_{\max} &= \alpha D_{\text{enc}} \\ &= 0.9 \times 22{,}649{,}241.6 \\ &= 20{,}384{,}317.44. \end{aligned} \tag{A.18}$$

BERT contains $L_{\text{spk}} = 72$ spiking linear layers. Under streaming early termination, each layer may additionally count at most one spike that first exceeds its activity budget. The total number of counted candidate spikes is therefore bounded by

$$\begin{aligned} K_{\text{count}} &\leq K_{\max} + L_{\text{spk}} \\ &= 20{,}384{,}389.44. \end{aligned} \tag{A.19}$$

Under a conservative local-memory model, each counted spike incurs one 1-bit spike read at $0.15625\,\text{pJ}$ and one INT32 counter increment at $0.1\,\text{pJ}$. The sign bit is not accessed. Each of

Table B: Theoretical computation-energy breakdown of nonlinear and auxiliary operations. All energy values are reported in mJ/sample and are computed from trace-derived operation counts using the unit energies in Table A.

Component	ANN	SpikeHENA	ANN implementation	SpikeHENA implementation
GELU	1.519387	0.078329	Native GELU	Polynomial MUL/ADD
LayerNorm arithmetic	0.020147	0.018387	ADD and scaling MUL	ADD and scaling MUL
Square	0.009093	0.027839	FP32 MUL	Temporal SHIFT/ADD
SQRT/RSQRT	0.000177	0.000122	Native RSQRT	Search, Newton iteration, and LUT
EXP/Exp-tree	0.018088	0.150847	Native EXP	Binary-tree traversal
Division	0.010853	0.439624	Native DIV	SHIFT, COMPARE, and SUB
Attention/Softmax auxiliary	0.009227	0.015066	Scaling, reduction, and masking	Scaling, reduction, and masking
SAA state/mask control	0	0.011094	–	State ADD and COMPARE
Pooler TANH	0.000071	0.000071	Native TANH	FP32 boundary TANH
Residual/structural ADD	0.013980	0.013979	FP32 ADD	FP32 ADD
Nonlinear/auxiliary total	1.601021	0.755358		

the 72 spiking linear layers requires at most $N = 10$ budget comparisons. Conservatively assigning each comparison the energy of an FP32 addition, $0.9\,\mathrm{pJ}$, gives

$$\begin{aligned} E_{\text{candidate}} \le\ & K_{\text{count}} e_{\text{1b-read}} \\ & + K_{\text{count}} e_{\text{INT32-ADD}} \\ & + L_{\text{spk}} N e_{\text{compare}} \\ =\ & 20{,}384{,}389.44 \times 0.15625\,\mathrm{pJ} \\ & + 20{,}384{,}389.44 \times 0.1\,\mathrm{pJ} \\ & + 72 \times 10 \times 0.9\,\mathrm{pJ} \\ =\ & 0.003185061 + 0.002038439 \\ & + 0.000000648 \\ =\ & 0.005224148\,\mathrm{mJ/sample}. \end{aligned} \tag{A.20}$$

The three terms in Eq. (A.20) correspond to the 1-bit spike reads, INT32 counter increments, and budget comparisons, respectively. This estimate is conservative because a population controller may count incoming spikes directly without an additional SRAM read. These operations are categorized as encoding and control overhead rather than synaptic MAC/AC computation.

We use the bound in Eq. (A.20) as the candidate-control term in the theoretical energy accounting below. Its local spike reads are included in this term and excluded from the general SRAM term to avoid double counting.

A.6 Nonlinear and Auxiliary Operations

Table B reports the energy of each nonlinear and auxiliary component separately. LayerNorm excludes the independently reported square and RSQRT operations. Likewise, the remaining Softmax arithmetic excludes EXP and DIV to avoid double counting.

For completeness, we detail the energy calculations for the main nonlinear components below. The ANN GELU energy is

$$E_{\text{GELU}}^{\text{ANN}} = 4.718592\text{M} \times 322\ \text{pJ} = 1.519387\ \text{mJ}, \tag{A.21}$$

whereas the polynomial GELU in SpikeHENA requires 18.874368 million multiplications and 9.437184 million additions:

$$\begin{aligned} E_{\text{GELU}}^{\text{SNN}} =\ & 18.874368\text{M} \times 3.7\ \text{pJ} \\ & + 9.437184\text{M} \times 0.9\ \text{pJ} \\ =\ & 0.078329\ \text{mJ}. \end{aligned} \tag{A.22}$$

The BinarySpike-based square operation in SpikeHENA performs 15.466301 million temporal shifts and an equal number of accumulations:

$$E_{\text{square}}^{\text{SNN}} = 15.466301\text{M} \times (0.9 + 0.9)\ \text{pJ} = 0.027839\ \text{mJ}. \tag{A.23}$$

The spike-based division approximation forms $2^k d$ by shifting the denominator d, rather than multiplying it by a temporal coefficient. It performs 151.093248 million shifts, 302.186496 million comparisons, and 35.191244 million conditional subtractions:

$$\begin{aligned} E_{\mathrm{DIV}}^{\mathrm{SNN}} &= \big(151.093248 + 302.186496 \\ &\qquad + 35.191244\big)\mathrm{M} \times 0.9\ \mathrm{pJ} \\ &= 0.439624\ \mathrm{mJ}. \end{aligned} \tag{A.24}$$

A.6.1 EXP-TREE

The spike-based exponential approximation in SpikeHENA traverses a depth-15 binary tree without explicitly constructing a numerical LUT index. At each level, a spike bit directly selects the left or right child. The sign associated with the first left-child selection is stored in a one-bit state and subsequently used to select the leaf value. Consequently, the traversal requires no per-level ADD or MUL operations.

For 7.077888 million exponential elements, the tree traverses 106.168320 million internal nodes. Conservatively, each node is assigned the energy of one 1-bit spike read and one comparison-equivalent branch operation. We additionally count one sign-bit read, one state write, one state read, and one 32-bit leaf-LUT read per element:

$$\begin{aligned} E_{\mathrm{ExpTree}} &= 106.168320\mathrm{M} \times (0.15625 + 0.9)\ \mathrm{pJ} \\ &\quad + 3 \times 7.077888\mathrm{M} \times 0.15625\ \mathrm{pJ} \\ &\quad + 7.077888\mathrm{M} \times 5\ \mathrm{pJ} \\ &= 0.150847\ \mathrm{mJ/sample}. \end{aligned} \tag{A.25}$$

This estimate is conservative because, in hardware, a spike bit may directly control a multiplexer rather than require a full FP32 comparison.

A.7 COMPUTE-RELATED ENERGY

Combining the synaptic-computation, neuron/encoding-arithmetic, candidate-control, and nonlinear/auxiliary components yields the total theoretical computation energy:

$$\begin{aligned} E_{\mathrm{comp}}^{\mathrm{ANN}} &= 51.401399 + 1.601021 \\ &= 53.002420\ \mathrm{mJ/sample}, \\ E_{\mathrm{comp}}^{\mathrm{SNN}} &= 15.140630 + 1.694792 \\ &\quad + 0.755358 + 0.005224148 \\ &= 17.596004\ \mathrm{mJ/sample}. \end{aligned} \tag{A.26}$$

Therefore,

$$\frac{E_{\mathrm{comp}}^{\mathrm{SNN}}}{E_{\mathrm{comp}}^{\mathrm{ANN}}} = 0.3320, \qquad \Delta E_{\mathrm{comp}} = 66.80\%. \tag{A.27}$$

A.8 MEMORY ACCESSES

We next adopt a logical-access model consistent with SpikedAttention. Static weights are loaded once and reused across all $T = 16$ time steps. SpikeHENA does not store 16 temporally pre-folded copies of the weight matrices. Each signed spike is represented by two bits: one spike bit and one sign bit.

For the ANN, each MAC operation reads two 32-bit operands. The resulting synaptic-operand memory energy is

$$\begin{aligned} E_{\mathrm{rd}}^{\mathrm{ANN}} &= 2N_{\mathrm{MAC}}^{\mathrm{ANN}} e_{\mathrm{SRAM32}} \\ &= 22.348434432\mathrm{B} \times 5\ \mathrm{pJ} \\ &= 111.742172\ \mathrm{mJ/sample}. \end{aligned} \tag{A.28}$$

Table C: Theoretical computation-energy breakdown for BERT inference. All energy values are reported in mJ/sample. The normalized total is reported relative to the ANN baseline. General memory, routing, and synchronization costs are excluded.

Component	ANN	SpikeHENA
MAC/AC synaptic computation	51.401399	15.140630
Neuron/encoding arithmetic	0	1.694792
Candidate spike-activity control	0	0.005224
Nonlinear and auxiliary operations	1.601021	0.755358
Total theoretical computation energy	53.002420	17.596004
Normalized total	1.0000	0.3320

For SpikeHENA, each static weight or corresponding QK/AV state operand is read once and reused across all time steps. The resulting number of 32-bit synaptic-operand accesses is 11.173625856 billion:

$$\begin{aligned} E^{\mathrm{SNN}}_{\mathrm{weight/state}} &= 11.173625856\mathrm{B} \times 5\ \mathrm{pJ} \\ &= 55.868129\ \mathrm{mJ/sample}. \end{aligned} \tag{A.29}$$

Each spike-triggered synaptic accumulation reads one 2-bit signed spike:

$$\begin{aligned} E^{\mathrm{SNN}}_{\mathrm{event-read}} &= N^{\mathrm{SNN}}_{\mathrm{SOP}} \times 0.3125\ \mathrm{pJ} \\ &= 16.819899656\mathrm{B} \times 0.3125\ \mathrm{pJ} \\ &= 5.256219\ \mathrm{mJ/sample}. \end{aligned} \tag{A.30}$$

The membrane state is represented using 32 bits. Assuming one state read and one state write per neuron slot, the resulting memory-access energy is

$$\begin{aligned} E^{\mathrm{SNN}}_{\mathrm{state}} &= N_{\mathrm{slot}} \times (5+5)\ \mathrm{pJ} \\ &= 566.329344\mathrm{M} \times 10\ \mathrm{pJ} \\ &= 5.663293\ \mathrm{mJ/sample}. \end{aligned} \tag{A.31}$$

Writing one 2-bit signed output spike per neuron slot incurs

$$\begin{aligned} E^{\mathrm{SNN}}_{\mathrm{event-write}} &= 566.329344\mathrm{M} \times 0.3125\ \mathrm{pJ} \\ &= 0.176978\ \mathrm{mJ/sample}. \end{aligned} \tag{A.32}$$

Embedding accesses, final output writes, and FP32 boundary-layer accesses are also included. The algorithm-specific Exp-Tree bit accesses and leaf-LUT reads are accounted for in Eq. (A.25), and candidate-control spike reads are accounted for in Eq. (A.20); all are excluded here to avoid double counting. The resulting general SRAM-access energies are 111.814921 and 66.972503 mJ/sample for the ANN and SpikeHENA, respectively.

A.9 DRAM, Routing, and Synchronization

Across the seven GLUE tasks, the checkpoints contain an average of 108.479234 million 32-bit parameter words. Assuming that all parameters are loaded from DRAM once per batch and shared across its 32 samples, both methods incur

$$\begin{aligned} E_{\mathrm{DRAM}} &= \frac{108.479234\mathrm{M} \times 640\ \mathrm{pJ}}{32} \\ &= 2.169585\ \mathrm{mJ/sample}. \end{aligned} \tag{A.33}$$

SpikeHENA requires no additional temporally pre-folded weight copies.

For tensor transport, the ANN generates 42.860544 million 32-bit logical flits per sample. Assuming an average of two hops, its routing energy is

$$\begin{aligned} E^{\mathrm{ANN}}_{\mathrm{route}} &= 42.860544\mathrm{M} \times 2 \times 3.5\ \mathrm{pJ} \\ &= 0.300024\ \mathrm{mJ/sample}. \end{aligned} \tag{A.34}$$

Table D: Memory-augmented theoretical energy for BERT under the baseline hardware mapping. All energy values are reported in mJ/sample. Exp-Tree bit accesses, leaf-LUT reads, and candidate-control spike reads are included in the theoretical-computation term and excluded from the general SRAM term to avoid double counting.

Component	ANN	SpikeHENA
Theoretical computation	53.002420	17.596004
General SRAM accesses	111.814921	66.972503
Parameter DRAM loading	2.169585	2.169585
NoC routing	0.300024	0.049797
Synchronization	0.000005	0.000048
Memory-augmented total	167.286955	86.787936
Normalized total	1.0000	0.5188
Reduction	–	48.12%

SpikeHENA generates 113.822007 million signed spike packets. Since each signed spike is represented by two bits, 16 signed spikes can be packed into one 32-bit flit. The resulting routing energy is

$$\begin{aligned} E_{\text{route}}^{\text{SpikeHENA}} &= \frac{113.822007\text{M}}{16} \times 2 \times 3.5 \text{ pJ} \\ &= 0.049797 \text{ mJ/sample}. \end{aligned} \tag{A.35}$$

As a conservative upper bound, assigning one full 32-bit flit to each signed spike yields 0.796754 mJ/sample.

The ANN and SpikeHENA execute 5.40625 and 48 logical barriers per sample, respectively. Under the assumed cost of 1000 pJ per barrier, their synchronization energies are negligible:

$$\begin{aligned} E_{\text{sync}}^{\text{ANN}} &= 0.000005 \text{ mJ/sample}, \\ E_{\text{sync}}^{\text{SpikeHENA}} &= 0.000048 \text{ mJ/sample}. \end{aligned} \tag{A.36}$$

A.10 Memory-Augmented Theoretical Energy

Table D combines the theoretical computation energy with the estimated logical SRAM, DRAM, routing, and synchronization costs. The estimate assumes the weight reuse, access pattern, spike packing, and communication costs specified above; clock-tree power and leakage are not included.

B Complete Ablation Experiments

B.1 Scope and Evaluation Protocol

This appendix presents the parameter scans that complement Tables 5 and 6 in the main paper. Because Table 6 already reports the add/remove ablations for spike-based Softmax, RMSNorm, and SiLU, we do not repeat the component-level results here. Instead, we report the encoding, time-step, stage-allocation, and spike-activity-budget scans used to determine the default hybrid configuration.

All configurations are evaluated on the same fixed balanced diagnostic subset, sampled equally from the six zero-shot datasets evaluated in the main paper: HellaSwag, PIQA, WinoGrande, ARC-Challenge, ARC-Easy, and BoolQ. For an example with gold-option negative log-likelihood ℓ_{gold}, the pooled gold-choice diagnostic perplexity is

$$\text{PPL}_{\text{gold}} = \exp\left(\frac{1}{K}\sum_{k=1}^{K} \ell_{\text{gold}}^{(k)}\right). \tag{B.1}$$

We also report the mean NLL margin $\ell_{\text{wrong}}^{\min} - \ell_{\text{gold}}$, for which larger positive values are preferable. "AC activity" denotes the cumulative number of nonzero spikes entering AC-based synaptic operations per encoded tensor element. It is a spike-count proxy rather than a measured energy or

Table E: Encoding-scheme scan on the balanced six-dataset subset described in Appendix B.1. ANN nonlinear operations are retained to isolate the encoding scheme. Gold PPL is the gold-choice diagnostic perplexity defined in Eq. (B.1); AC Act. is the cumulative number of nonzero spikes entering AC-based synaptic operations per encoded tensor element.

Encoding	Setting	T	DeepSeek-V2-Lite		Qwen1.5-MoE-A2.7B	
			Gold PPL ↓	AC Act. ↓	Gold PPL ↓	AC Act. ↓
BinarySpike	$N = 16$	16	3.231372	2.779	5.630653	2.910
	$N = 18$	18	3.040140	3.267	3.525302	3.344
	$N = 20$	20	3.029151	3.773	3.350786	3.662
	$N = 24$	24	3.032051	4.701	3.346208	4.066
	$N = 32$	32	3.033043	5.640	3.345455	4.300
	$N = 44$	44	3.032996	5.821	3.346382	4.332
SingleSpike	$B = 1.10$	44	11.170068	0.186	13.257139	0.325
	$B = 1.10$	80	8.694300	0.749	5.951771	0.869
	$B = 1.10$	120	3.098265	0.993	3.390174	0.995
	$B = 1.15$	44	12.963885	0.561	15.607148	0.620
	$B = 1.15$	80	3.139789	0.991	3.427004	0.993
	$B = 1.15$	120	3.122865	1.000	3.398916	1.000
Ours$_{N=24,M=20}$	$\alpha = 0.9$	44	**3.027800**	1.604	**3.400300**	1.616

power quantity and differs from the slot-level spike rate, which is obtained by dividing the number of nonzero spikes by the total number of neuron–time slots.

The encoding scan in Appendix B.2 retains the ANN nonlinear operations to isolate the effect of the encoder. The time-step, N/M-allocation, and α scans in Appendix B.3, B.4, B.5 use the complete SpikeHENA stack, including the spike-based Softmax, RMSNorm, and SiLU units. The time-step and stage-allocation scans fix $\alpha = 0.7$, whereas the spike-activity-budget scan varies α. These fixed-subset results are intended for controlled comparisons among configurations.

B.2 Dense Encoding-Scheme Scan

Table E reports the intermediate BinarySpike time-step settings and the two SingleSpike bases omitted from the compact table in the main paper. BinarySpike approaches the ANN diagnostic PPL when T reaches approximately 20–24, but its spike activity increases as more binary time steps are used. SingleSpike exhibits substantially lower spike activity; however, at $T = 44$, underflow over the activation range remains pronounced. For $B = 1.10$, the measured underflow rates are 21.8% and 32.1% for DeepSeek-V2-Lite and Qwen1.5-MoE, respectively; for $B = 1.15$, they decrease to 11.5% and 12.6%. Extending the SingleSpike window to 80–120 time steps reduces underflow and recovers diagnostic PPL, but requires substantially more time steps. These results motivate the hybrid encoder, which uses BinarySpike to represent large-magnitude activation components and SingleSpike to encode the remaining residuals.

B.3 Sensitivity to the Total Number of Time Steps

Table F varies the total number of time steps while retaining the complete SpikeHENA stack. For DeepSeek-V2-Lite, diagnostic PPL improves steadily from $T = 24$ to $T = 44$ and then saturates. Increasing T from 44 to 56 raises AC activity by only 0.032 spikes per encoded tensor element and yields no consistent PPL improvement. Qwen1.5-MoE exhibits a sharper transition: configurations with $T \leq 40$ fail to preserve the fixed-subset likelihoods, whereas performance recovers once Stage I reaches $N = 24$. These shorter-window results therefore indicate insufficient temporal coverage for this architecture rather than a smooth trade-off between representation quality and the number of time steps. Across both models, $T = 44$ is the smallest shared setting in this scan that avoids degradation at fewer time steps.

Table F: Sensitivity to the total number of time steps for the complete SpikeHENA stack with $\alpha = 0.7$, evaluated on the balanced six-dataset subset described in Appendix B.1. The $N = 24$, $M = 20$, and $T = 44$ configuration is used as the reference for the controlled scans in this appendix. AC Act. denotes AC activity.

N	M	T	DeepSeek-V2-Lite		Qwen1.5-MoE-A2.7B	
			Gold PPL ↓	AC Act. ↓	Gold PPL ↓	AC Act. ↓
14	10	24	3.815167	1.280	34.596985	1.306
16	16	32	3.674781	1.395	48.329752	1.352
20	20	40	3.245039	1.387	54.291175	1.388
24	**20**	**44**	**3.021393**	**1.386**	**3.733682**	**1.397**
24	24	48	3.025153	1.406	3.588837	1.414
28	28	56	3.027397	1.418	3.654166	1.416

Table G: Sensitivity to the allocation of time steps between Stage I and Stage II under a fixed total of $T = 44$. Results are obtained using the complete SpikeHENA stack with $\alpha = 0.7$ on the balanced six-dataset subset described in Appendix B.1. Bold denotes the reference configuration $N = 24$, $M = 20$. AC Act. denotes AC activity.

N	M	T	DeepSeek-V2-Lite		Qwen1.5-MoE-A2.7B	
			Gold PPL ↓	AC Act. ↓	Gold PPL ↓	AC Act. ↓
12	32	44	5.859227	1.440	25.062445	1.430
16	28	44	3.657783	1.425	46.493893	1.391
20	24	44	3.301963	1.407	49.156324	1.408
24	**20**	**44**	**3.021393**	**1.386**	**3.733682**	**1.397**
28	16	44	3.021899	1.388	3.881229	1.394
32	12	44	3.017501	1.319	4.025241	1.354
36	8	44	3.012521	1.287	4.015886	1.306

B.4 Allocation Between BinarySpike and SingleSpike Stages

Tables G and H evaluate two complementary time-step allocation settings. Table G fixes the total number of time steps at $T = N + M = 44$ and reallocates them between Stage I and Stage II, whereas Table H varies one stage while holding the other fixed.

Under the fixed-T setting, allocating too few time steps to Stage I causes substantial Gold PPL degradation, particularly for Qwen1.5-MoE-A2.7B. DeepSeek-V2-Lite becomes relatively insensitive once $N \geq 24$, while its spike activity generally decreases as time steps are shifted from Stage II to Stage I. For Qwen1.5-MoE-A2.7B, the best tested hybrid allocation is $N = 24$, $M = 20$. Further increasing N reduces spike activity but slightly worsens Gold PPL. These results indicate that the preferred allocation between the two stages is architecture-dependent.

The one-stage-at-a-time controls in Table H support the same conclusion without fixing T. With $M = 20$, DeepSeek-V2-Lite improves as N increases from 12 to 24, whereas Qwen1.5-MoE-A2.7B does not recover stable representation quality until $N = 24$. With $N = 24$, reducing M from 16 to 8 has little effect on DeepSeek-V2-Lite but substantially degrades Qwen1.5-MoE-A2.7B at $M = 8$. Stage II is therefore necessary for some architectures even when its contribution to total spike activity is small.

B.5 Spike-Rate Threshold

The spike-rate threshold α limits the Stage I spike activity before the encoding proceeds to Stage II. Table I shows that AC activity increases nearly monotonically with α. DeepSeek-V2-Lite is relatively insensitive once $\alpha \geq 0.5$: its diagnostic PPL remains within a narrow range, whereas AC activity nearly doubles between $\alpha = 0.5$ and $\alpha = 1.5$. Qwen1.5-MoE benefits more from increased Stage I spike activity. Raising α from 0.5 to 0.9 reduces its diagnostic PPL from 3.835344 to 3.449994, while increasing AC activity from 1.218 to 1.597. Larger values further improve Qwen1.5-MoE up to $\alpha = 1.3$, but with diminishing PPL gains and steadily increasing spike ac-

Table H: One-stage-at-a-time sensitivity of the Stage I and Stage II time-step allocations. The complete SpikeHENA stack uses $\alpha = 0.7$ and is evaluated on the balanced six-dataset subset described in Appendix B.1. The N-control rows fix $M = 20$, whereas the M-control rows fix $N = 24$. Bold denotes the reference configuration $N = 24$, $M = 20$, and $T = 44$. AC Act. denotes AC activity.

Control	N	M	T	DeepSeek-V2-Lite		Qwen1.5-MoE-A2.7B	
				Gold PPL ↓	AC Act. ↓	Gold PPL ↓	AC Act. ↓
Vary N	12	20	32	5.536133	1.384	25.678355	1.388
Vary N	16	20	36	3.637118	1.392	48.170164	1.367
Vary N	20	20	40	3.245039	1.387	54.291175	1.388
Reference	**24**	**20**	**44**	**3.021393**	**1.386**	**3.733682**	**1.397**
Vary M	24	8	32	3.011884	1.288	17.847877	1.300
Vary M	24	12	36	3.017129	1.319	3.934997	1.356
Vary M	24	16	40	3.021693	1.387	3.890598	1.396
Reference	**24**	**20**	**44**	**3.021393**	**1.386**	**3.733682**	**1.397**

Table I: Sensitivity to the spike-rate threshold α for the complete SpikeHENA stack with $N = 24$, $M = 20$, and $T = 44$, evaluated on the balanced six-dataset subset described in Appendix B.1. NLL Mar. is the mean margin between the lowest incorrect-option NLL and the gold-option NLL, with positive values indicating preference for the gold option. AC Act. denotes AC activity. Bold indicates the cross-model default used in the main experiments.

α	DeepSeek-V2-Lite			Qwen1.5-MoE-A2.7B		
	Gold PPL ↓	NLL Mar. ↑	AC Act. ↓	Gold PPL ↓	NLL Mar. ↑	AC Act. ↓
0.3	3.098913	-0.0001	0.987	3.995242	0.0077	1.033
0.5	3.025861	0.0049	1.196	3.835344	0.0091	1.218
0.7	3.021393	0.0048	1.386	3.733682	0.0126	1.397
0.9	**3.027186**	**0.0041**	**1.593**	**3.449994**	**0.0139**	**1.597**
1.1	3.027525	0.0049	1.786	3.406865	0.0173	1.780
1.2	3.028670	0.0050	1.881	3.398478	0.0147	1.874
1.3	3.027323	0.0045	1.983	3.393843	0.0115	1.997
1.5	3.028752	0.0046	2.186	3.402271	0.0127	2.163

tivity. Thus, $\alpha = 0.9$ provides a cross-model compromise rather than the independently optimal setting for either architecture.

B.6 Summary and Selection Rule

The scans reveal three practical constraints. First, BinarySpike alone can recover low diagnostic PPL with a moderate number of time steps, but incurs substantially higher spike activity than the hybrid encoder. Second, SingleSpike alone requires substantially more time steps to cover the activation range. Third, the preferred allocation between Stage I magnitude representation and Stage II residual encoding varies across architectures. We therefore adopt $N = 24$, $M = 20$, and $\alpha = 0.9$ as a common cross-model operating point. This configuration avoids degradation at fewer time steps, retains 20 time steps for Stage II, and limits the increase in spike activity associated with larger α.

B.7 Additional Results on GSM8K

We further evaluate SpikeHENA on GSM8K (Cobbe et al., 2021) to examine its performance on mathematical reasoning tasks involving quantitative relationships and multi-step calculations. Both models use $N = 24$ BinarySpike steps and $M = 20$ SingleSpike residual steps, giving a total of $T = N + M = 44$ encoding time steps. We set $\alpha = 1.1$ for Qwen1.5-MoE-A2.7B and $\alpha = 0.9$ for DeepSeek-V2-Lite.

Table J reports the accuracy results alongside the official ANN reference results (Team, 2024; Liu et al., 2024). SpikeHENA achieves 60.3% accuracy on Qwen1.5-MoE-A2.7B and 39.0% on DeepSeek-V2-Lite, respectively 1.2 and 2.1 percentage points below their official ANN reference

Table J: GSM8K accuracy. ANN values are official reference results, and SpikeHENA values are obtained from our evaluations. Both models use $N = 24$, $M = 20$, and $T = 44$, with the indicated spike-activity thresholds.

Model	Official ANN accuracy (%)	SpikeHENA accuracy (%)	α
Qwen1.5-MoE-A2.7B	61.5	60.3	1.1
DeepSeek-V2-Lite	41.1	39.0	0.9

results. These additional results support the applicability of SpikeHENA to mathematical reasoning tasks beyond the benchmarks evaluated in the main text.

The spike-activity threshold α provides a practical way to balance accuracy and computational energy. Increasing α relaxes the Stage I population-activity constraint, allowing additional BinarySpike contributions before the neuron transitions to residual encoding. This larger spike budget can improve representation fidelity and help recover task accuracy, at the cost of increased spike activity and the associated synaptic-computation energy. This behavior is consistent with the spike-activity trends reported in Appendix B.5. For Qwen1.5-MoE-A2.7B, we use $\alpha = 1.1$ for GSM8K, moderately increasing the activity budget relative to the default setting of $\alpha = 0.9$, while keeping N, M, and T unchanged. DeepSeek-V2-Lite retains $\alpha = 0.9$. These settings illustrate how the activity budget can be adjusted to accommodate the accuracy requirements of different models and tasks.

Qualitative example. We additionally present an example involving the time required to travel halfway around the Earth. The question specifies a circumference of 40,000 kilometers and a daily travel distance of 400 kilometers, giving the correct answer $40{,}000/2/400 = 50$ days. In the DeepSeek-V2-Lite example, the ANN response overlooks the halfway condition and answers 100, whereas SpikeHENA correctly uses a distance of 20,000 kilometers and answers 50. For Qwen1.5-MoE-A2.7B, both the ANN and SpikeHENA with $\alpha = 1.1$ correctly answer 50. At $\alpha = 0.9$, Qwen1.5-MoE-A2.7B instead produces an incorrect answer of 1 for this example, illustrating how a modest increase in the spike-activity budget can recover a correct response. Together with the aggregate accuracy results, these examples demonstrate that SpikeHENA can retain the ability to interpret quantitative constraints and carry out multi-step calculations after conversion.

Question.

> Alfie, the albatross, flies 400 kilometers every day. If the circumference of the earth is 40,000 kilometers, how many days will it take Alfie to fly a distance equal to half of the way around the earth?

Ground-truth answer: 50.

DeepSeek-V2-Lite — ANN.

```
Alfie flies 400 kilometers every day, so he flies 400
kilometers in one day.
The circumference of the earth is 40,000 kilometers, so it
takes Alfie 40,000/400 = 100 days to fly a distance equal to
half of the way around the earth.
The answer is 100
```

DeepSeek-V2-Lite — SpikeHENA ($\alpha = 0.9$).

```
Alfie flies 400 kilometers every day, so he flies 400
kilometers in one day.
Alfie flies 40,000 kilometers in 40,000/400 = 100 days.
Alfie flies 20,000 kilometers in 20,000/400 = 50 days.
The answer is 50
```

Qwen1.5-MoE-A2.7B — ANN.

```
The circumference of the earth is 40,000 kilometers, so half
of the way around the earth is 40,000/2 = 20,000 kilometers.
Alfie flies 400 kilometers every day, so it will take him
20,000/400 = 50 days to fly a distance equal to half of the
way around the earth.
The answer is:  50.
```

Qwen1.5-MoE-A2.7B — SpikeHENA ($\alpha = 0.9$).

```
Alfie flies 400 kilometers every day, so it will take him
400 kilometers / 400 kilometers per day = 1 day to fly half
way around the earth.
The answer is 1
```

Qwen1.5-MoE-A2.7B — SpikeHENA ($\alpha = 1.1$).

```
Alfie flies 400 kilometers every day, so it will take him
40,000 kilometers / 400 kilometers per day = 100 days to fly
around the earth.
Half of the way around the earth is 40,000 kilometers / 2 =
20,000 kilometers.
It will take Alfie 20,000 kilometers / 400 kilometers per
day = 50 days to fly half the way around the earth.
The answer is 50
```

Display note. DeepSeek-V2-Lite responses are shown through the first final answer. For Qwen1.5-MoE-A2.7B, the ANN response and the SpikeHENA responses at $\alpha = 0.9$ and $\alpha = 1.1$ are shown in full. The quoted responses preserve the original wording and calculations.

C Error Analysis

C.1 Notation and Scope

We analyze the deterministic local approximation errors introduced by the Two-Stage Hybrid Encoding Neuron and the proposed spike-based nonlinear units.

C.2 Hybrid-Encoding Error Bound

For a scalar activation x, let

$$x_{\mathrm{I}} = \sum_{t=1}^{N} s^t \vartheta_{\mathrm{I}}^t, \qquad r_{\mathrm{I}} = x - x_{\mathrm{I}}, \tag{C.1}$$

denote the Stage I reconstruction and the corresponding residual. For activations within the range covered by BinarySpike, the exponentially decreasing thresholds and subtractive resets ensure that

$$|r_{\mathrm{I}}| < \vartheta_f. \tag{C.2}$$

To see this, before processing a binary threshold ϑ, the residual magnitude is smaller than 2ϑ. If a spike is generated, one signed ϑ is subtracted from the residual; otherwise, the residual magnitude is already smaller than ϑ. The invariant therefore follows by induction. If the spike-rate constraint terminates Stage I, the rejected threshold is not subtracted from the membrane state, and the last accepted threshold is retained as ϑ_f. The same invariant consequently remains valid.

Under the Stage II schedule, adjacent effective SingleSpike thresholds are separated by

$$\Delta_{\mathrm{II}} = \frac{\vartheta_f}{M+1}. \tag{C.3}$$

Suppose that the first nonzero SingleSpike occurs at refinement step m. Because the preceding threshold is not crossed,

$$0 \le |r_{\mathrm{I}}| - \vartheta_{\mathrm{II}}^{m} < \Delta_{\mathrm{II}}. \tag{C.4}$$

If no SingleSpike is generated, then $|r_{\mathrm{I}}| < \vartheta_{\mathrm{II}}^{M} = \Delta_{\mathrm{II}}$. Therefore, for the reconstruction

$$\hat{x} = x_{\mathrm{I}} + \sum_{m=1}^{M} s^{N+m} \vartheta_{\mathrm{II}}^{m}, \tag{C.5}$$

the approximation error satisfies

$$\boxed{|x - \hat{x}| < \frac{\vartheta_f}{M+1}}. \tag{C.6}$$

The same construction also provides a bound on spike activity. The accepted Stage I spike activity is at most α, and Stage II generates at most one nonzero spike per neuron. Consequently,

$$\frac{1}{N_{\mathrm{neu}}} \sum_{i=1}^{N_{\mathrm{neu}}} \sum_{t} \mathbf{1}[s_i^t \neq 0] \le \alpha + 1. \tag{C.7}$$

C.3 Signed-Binary Division Error

The numerical numerator input A is itself obtained by accumulating spikes from preceding layers.

Let $q = A/B$, where $B > 0$, and consider a T-step divider with binary weights

$$\beta_t = 2^{r-t+1}, \qquad t = 1, \dots, T. \tag{C.8}$$

The decoded quotient is

$$\hat{q}_T = \sum_{t=1}^{T} s^t \beta_t, \qquad s^t \in \{-1, 0, +1\}. \tag{C.9}$$

Define the quotient residual after step t as

$$e_t = q - \sum_{\tau=1}^{t} s^\tau \beta_\tau. \tag{C.10}$$

Assume that r is chosen to cover the quotient range:

$$|q| < 2^{r+1}. \tag{C.11}$$

Before processing step t, suppose that $|e_{t-1}| < 2\beta_t$. If $|e_{t-1}| < \beta_t$, no spike is generated. Otherwise, subtracting the signed weight β_t yields a residual with magnitude smaller than β_t. Because $\beta_{t+1} = \beta_t/2$, this establishes by induction that

$$|e_t| < \beta_t. \tag{C.12}$$

Therefore, the final division error satisfies

$$\boxed{\left| \frac{A}{B} - \hat{q}_T \right| < 2^{r-T+1}}. \tag{C.13}$$

C.4 Square-Root Approximation Error

Let $a \ge 0$, and let $\hat{a}$ denote its hybrid-encoded reconstruction. For $a, \hat{a} > 0$, the error due to input encoding satisfies

$$|\sqrt{a} - \sqrt{\hat{a}}| = \frac{|a - \hat{a}|}{\sqrt{a} + \sqrt{\hat{a}}}. \tag{C.14}$$

Let ϑ_0 be the threshold corresponding to the first nonzero input spike, and initialize $x_0 = \sqrt{\vartheta_0}$. Because consecutive tested thresholds differ by at most a factor of two,

$$\vartheta_0 \le \hat{a} < 2\vartheta_0, \qquad \frac{1}{\sqrt{2}} < \frac{x_0}{\sqrt{\hat{a}}} \le 1. \tag{C.15}$$

Under exact division, Newton's iteration $F(x) = \frac{1}{2}\left(x + \frac{\hat{a}}{x}\right)$ satisfies

$$F(x) - \sqrt{\hat{a}} = \frac{(x - \sqrt{\hat{a}})^2}{2x}. \tag{C.16}$$

Suppose that the divider at iteration k introduces an error d_k, where $|d_k| < 2^{r-T+1}$ by Eq. (C.13). The implemented update is then

$$x_{k+1} = F(x_k) + \frac{d_k}{2}, \tag{C.17}$$

which yields

$$|x_{k+1} - \sqrt{\hat{a}}| \le \frac{|x_k - \sqrt{\hat{a}}|^2}{2x_k} + 2^{r-T}. \tag{C.18}$$

In the idealized case of exact division, applying the two Newton iterations used by the square-root unit to the initialization in Eq. (C.15) gives the relative-error bound

$$0 \le \frac{x_2}{\sqrt{\hat{a}}} - 1 \le \frac{17}{12\sqrt{2}} - 1. \tag{C.19}$$

Therefore, the complete square-root approximation error is bounded by the sum of the input-encoding error in Eq. (C.14), the perturbed Newton error governed by Eq. (C.18), and the final hybrid-output encoding error in Eq. (C.6).

C.5 Exponentiation Approximation Error

Let $\bar{x} = \mathrm{clip}(x, -10, 10)$. We distinguish the true Stage I residual from its Stage II reconstruction:

$$\bar{x} = x_0 + \delta, \qquad \delta = \hat{\delta} + e_{\mathrm{enc}}, \qquad |e_{\mathrm{enc}}| < \frac{\vartheta_f}{M+1}. \tag{C.20}$$

Here, x_0 is the coarse component represented by BinarySpike, δ is the true residual after Stage I, and $\hat{\delta}$ is its SingleSpike reconstruction. The term e_{enc} explicitly accounts for the residual not represented by the final SingleSpike refinement level.

First, assume that the temporal-tree lookup returns the exact value e^{x_0}. Using the first-order approximation $\tilde{y} = e^{x_0}(1 + \hat{\delta})$, we obtain

$$\begin{aligned} |e^{\bar{x}} - \tilde{y}| &= e^{x_0}\left|e^{\delta} - 1 - \delta + e_{\mathrm{enc}}\right| \\ &\le e^{x_0}\left(\frac{1}{2}e^{|\delta|}\delta^2 + \frac{\vartheta_f}{M+1}\right), \end{aligned} \tag{C.21}$$

where the inequality follows from Taylor's theorem and the hybrid-encoding error bound.

Thus, the local Taylor remainder is second order in the true Stage I residual δ, whereas the error due to the unrepresented SingleSpike residual decreases linearly with the Stage II refinement resolution.

C.6 Propagation to Transformer Nonlinear Layers

We finally describe how the primitive error bounds propagate through Softmax, RMSNorm, and SiLU. We first establish a common perturbation bound for division. Let $y = a/b$, where $b > 0$, and suppose that

$$|\hat{a} - a| \le \varepsilon_a, \qquad |\hat{b} - b| \le \varepsilon_b < b. \tag{C.22}$$

If the final divider introduces an additional error $\eta_{\mathrm{div}} < 2^{r-T+1}$, then

$$|\hat{y} - y| \le \frac{\varepsilon_a}{b - \varepsilon_b} + \frac{|a|\varepsilon_b}{b(b - \varepsilon_b)} + \eta_{\mathrm{div}}. \tag{C.23}$$

For RMSNorm, ε_a includes the numerator-encoding error, whereas ε_b includes the errors introduced by squaring, reduction, and square-root approximation. For SiLU, the denominator error ε_b is induced by the exponentiation approximation in Eq. (C.21). The corresponding output-error bounds for both layers then follow from Eq. (C.23), together with their final output-encoding errors.

For Softmax, let $z_i = e^{x_i - \max_j x_j}$ and $Z = \sum_i z_i$, and define the aggregate exponentiation error as

$$E = \sum_i |\hat{z}_i - z_i|. \tag{C.24}$$

When $E < Z$, the normalization perturbation satisfies

$$\|\hat{\boldsymbol{p}} - \boldsymbol{p}\|_1 \leq \frac{2E}{Z - E} + d\,\eta_{\mathrm{div}}, \tag{C.25}$$

where d is the dimension over which Softmax is normalized, and η_{div} bounds the division error for each component. Subtracting the maximum logit ensures that at least one $z_i = 1$, and hence $Z \geq 1$, which limits the amplification of exponentiation errors during normalization. Together, Eqs. (C.6), (C.13), (C.18), and (C.21) provide deterministic local error bounds for all three spike-based nonlinear layers.

D Related Work

D.1 Spike Encoding for ANN-to-SNN Conversion

ANN-to-SNN conversion represents real-valued activations using temporal spike sequences. Rate coding encodes values through spike counts and improves representation resolution with more time steps, but generally incurs high spike activity (Bu et al., 2022; Li et al., 2021; Deng & Gu, 2021). Time-to-first-spike (TTFS) and other SingleSpike encoding schemes generate sparse spike activity by representing values through spike timing, but require more time steps to achieve fine-grained representations over wide dynamic ranges (Rueckauer & Liu, 2018; Zhao et al., 2025). Phase coding assigns time-dependent weights to spikes and achieves high resolution with fewer time steps, at the cost of increased spike activity (Hwang et al., 2024; Hwang & Kung, 2025). Existing encoding schemes therefore involve trade-offs among representation fidelity, spike activity, and the number of time steps. In contrast, SpikeHENA combines BinarySpike and SingleSpike encoding through a continuously evolving membrane state to represent both large-magnitude activation components and the remaining residuals.

D.2 Spiking Transformers and LLM Conversion

Early spiking Transformers primarily redesign attention mechanisms and directly train the resulting models for vision tasks (Zhou et al., 2023; 2026; 2024), whereas ANN-to-SNN conversion directly reuses pretrained parameters without training large-scale SNNs from scratch. SpikeZIP-TF (You et al., 2024) extends ANN-to-SNN conversion to Transformer architectures, while SpikedAttention (Hwang et al., 2024) provides a spike-based implementation of Softmax but retains conventional computations for RMSNorm and SiLU. LAS (Chen et al., 2026b) approximates LLM nonlinearities using FewSpike neurons (Stöckl & Maass, 2021), whose multi-valued outputs differ from the signed binary spikes considered in this work. LAS introduces outlier-aware threshold neurons and hierarchically gated nonlinear approximations for LLMs, whereas Wang et al. (2026) extend FS neurons with multi-basis exponential decay (MBE) and evaluate conversion on ViT, RoBERTa, and GPT-2. Therefore, jointly achieving efficient temporal encoding and unified spike-driven implementations of Softmax, RMSNorm, and SiLU under a signed binary spike representation remains insufficiently explored.