2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

SpikeRWKV:Energy-efficient Large Language Model with Spiking Neural Network

Yulu Zhang[1&], Qianzi Shen[2&] and Zijian Wang[1(✉)]

[1] Donghua University, Shanghai 201620, China
wang.zijian@dhu.edu.cn
[2] China Mobile Shanghai Industry Research Institude, Shanghai 200031, China
[&]These authors have contributed equally to this work and share first authorship

Abstract. Spiking Neural Networks (SNNs), as the third generation of neural networks, hold great promise for enhancing the energy efficiency of large language models (LLMs) due to their event-driven computation. However, their naive application in large-scale models typically depends on binary spike simulations over long time steps, making it challenging to balance performance and energy consumption. To address this issue, we propose a Multi-head Spike Encoding scheme with three advantages. First, it enables parallel spike processing to accelerate computation; Second, it supports precise representation of positive and negative spikes; Third, it mitigates energy surges caused by high-frequency spikes through hierarchical spike decomposition. To demonstrate the effectiveness of our encoding scheme, we introduce SpikeRWKV, an SNN-based adaptation of the RWKV language model. Experimental results demonstrate that SpikeRWKV significantly enhances performance on natural language understanding (NLU) tasks, achieving a 3.15× reduction in energy consumption compared to the baseline, along with an 8.3% lower perplexity and 5.7% improvement in bits-per-character (BPC). Furthermore, SpikeRWKV is 3.88× more energy-efficient than its non-spiking counterpart.

Keywords: Spiking neural networks · Energy efficiency · Spike encoding scheme.

1 Introduction

The rapid advancement of deep learning has driven remarkable progress in Artificial Neural Networks (ANNs) [1], particularly in the development of Large Language Models (LLMs) [2]. However, as task complexity grows and datasets scale up [3], ANNs face increasing challenges in terms of energy consumption [4]. Biologically inspired Spiking Neural Networks (SNNs) have emerged as a promising alternative, offering greater energy efficiency [5]. In contrast to ANNs that rely on continuous signal transmission [6], SNNs communicate via discrete binary spikes, substantially reducing both computational and memory overhead [7]. Moreover, spiking neurons inherently exhibit

spatiotemporal dynamics, making SNNs well-suited for LLM tasks that involve complex sequential dependencies.

Despite the potential of SNNs for energy-efficient computation, their application to LLM tasks remains limited by fundamental representational challenges. A widely used approach, repetitive spike coding, encodes inputs over multiple time steps using identical spike patterns. However, this method often fails to preserve fine-grained semantic distinctions, particularly when high-frequency spike activity dominates, leading to reduced model accuracy [8]. This limitation becomes especially pronounced in deep LLM architectures, where such representational errors can propagate through multiple layers and amplify downstream inaccuracies [9]. Moreover, the long latency associated with encoding high-precision input values into spike trains introduces both computational overhead and energy inefficiency [10].

To address the aforementioned challenges, we propose a novel Multi-head Spike Encoding scheme tailored for SNN-based large language models. This scheme introduces several key innovations to enhance both computational efficiency and representational fidelity.

First, the proposed encoding scheme supports parallel spike processing by enabling the simultaneous expression of multiple spike streams across different layers. This parallelism allows for concurrent spike computations, significantly reducing inference latency.

Second, the encoding framework introduces a flexible mechanism for representing both positive and negative spikes. By incorporating a spike polarity flag, the scheme effectively distinguishes excitatory and inhibitory signals. Furthermore, it allows for fine-grained control over spike conversion accuracy by adjusting key parameters such as the number of spike layers and time steps during the encoding process.

Third, the scheme mitigates energy surges associated with high-frequency spike events by decomposing spike activity hierarchically across multiple layers, thereby enhancing signal stability and efficiency. To demonstrate the effectiveness of our encoding scheme, we integrate the Multi-head Spike Encoding mechanism into the RWKV architecture. Experimental results demonstrate that our method achieves both high performance and low energy consumption. The key contributions of this work are summarized as follows:

- We propose a Multi-head Spike Encoding scheme that effectively improves the representation capability and efficiency over repetitive spike coding.
- We introduce SpikeRWKV, an SNN-based variant of the RWKV model, which achieves low energy consumption while preserving high performance.
- Experiments on both natural language generation (NLG) and natural language understanding (NLU) tasks demonstrate the effectiveness of SpikeRWKV. Remarkably, SpikeRWKV surpasses the baseline with repetition coding in terms of both bits-per-character (BPC) and perplexity (PPL). Moreover, SpikeRWKV reduces energy consumption by a factor of 3.88× compared to the non-spiking RWKV model.

2 Related Work

2.1 Spiking for LLM

Currently, ANN-SNN conversion and direct training are the main methods for training SNNs [11]. The ANN-SNN conversion technique facilitates the transition from ANNs to SNNs by substituting the traditional ReLU activation function [12] with the average firing rate of spikes. However, this approach typically necessitates hundreds or even thousands of time steps, resulting in increased computational costs. To address this issue, SpikingBERT [13] has optimized the conversion process and successfully applied it to the BERT model. Nevertheless, these methods still encounter challenges related to resource inefficiency due to the prolonged time steps required during the input value transformation process.

Aiming at the semantic loss problem caused by the discreteness of the temporal encoding in the above encoding methods and the inefficient computational resources caused by the long time steps in the conversion process, this paper proposes an improved ANN-SNN conversion method——the multi-head spike encoding model. The expression accuracy per unit time is enhanced by expressing multiple bits simultaneously, which effectively improves the accuracy while reducing resource waste, providing a more efficient and accurate solution for SNN-based LLM tasks.

2.2 Spiking Encoding

Spiking neurons have the capacity to embed and process information within the spatiotemporal domain, enabling them to theoretically encode data across multiple levels and spatiotemporal scales. Temporal encoding represents information based on the timing of the first spike's arrival [14][15]. While this method relies on precise spike timing, it is highly sensitive to temporal errors and noise [16].

To address these limitations, [17] introduced a direct encoding method that bypasses additional transformations by directly feeding the input signal into the network. While direct encoding has demonstrated effectiveness in computer vision applications, its application to LLM incurs significant computational overhead [18].

As can be seen, too short a time step leads to poor accuracy, while too long a time step leads to excessive energy consumption. Therefore, this study developes a new encoding scheme based on repetition coding that was tailored for the LLM tasks. This new method improves the expressive energy per unit time by controlling the expression precision during the transformation process, while also reducing the large computational overhead caused by the long steps in the precise expression process.

3 Method

3.1 SpikeRWKV

SpikeRWKV is RWKV6 [19] converted spiking neural networks. In SpikeRWKV, we add a spiking neural network to save the computing resources of RWKV. Fig. 1 shows

the architecture of the SpikeRWKV model. First, a spiking flag is introduced to record the positive and negative values of the conversion value. Then, the pre-trained input in RWKV is directly converted into spikes through a multi-head spike encoding scheme, and a spiking flag is added during the conversion process to increase the accuracy of the expression. Next, the converted spikes are weighted to complete the Spiking Time Mixing operation.

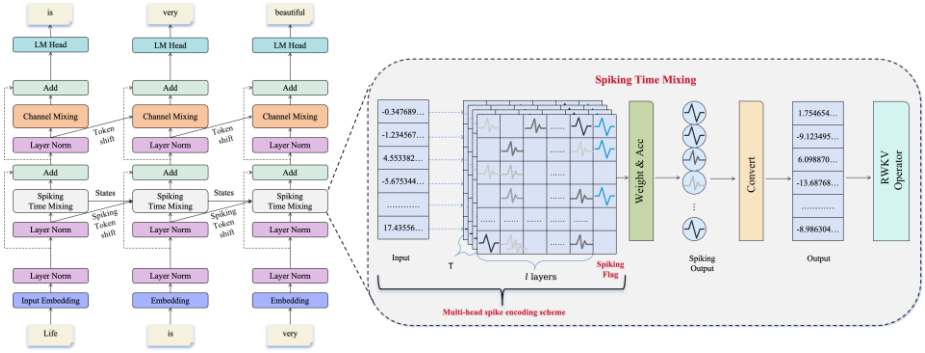


Fig. 1. SpikeRWKV architecture: First, a spiking flag is introduced to record the positive and negative values of the conversion value. Then, the pre-trained input in RWKV is directly converted into spikes through a multi-head spike encoding scheme, and a spiking flag is added during the conversion process to increase the accuracy of the expression. Next, the converted spikes are weighted to complete the Spiking Time Mixing operation.

3.2 Spiking Neural Network

In SNN, the Integrate-and-Fire (IF) spike neuron is often used to transform ANN to SNN. The expression of the IF spike neuron is:

$$v_i^j = v_i^{j-1} + W_i\theta_{i-1}d_{i-1}^j - \theta_i d_i^j \tag{1}$$

v_i^j represents the neural membrane potential of the i-th layer neuron at time step j, θ_{i-1} is the threshold of the i-1 layer neuron, W_i is the linear transformation matrix of the i-th layer, and d_{i-1}^j is the output value of the spike neuron of the i-1 layer. It is defined as:

$$d_i^j = H(u_i^j - \theta_i) \tag{2}$$

And here $u_i^j = v_i^{j-1} + W_i\theta_{i-1}d_{i-1}^j$ represents the spike neuron membrane potential when the spike neuron has not been stimulated at the j time step. H() represents the step function. When the membrane potential u_i^j exceeds the threshold θ_i, the neuron generates an output spike and resets the membrane potential by subtracting the threshold to reduce information loss.

3.3 Spiking Flag

In the spiking neural network, we set a flag called "spiking flag" to record the positive and negative conditions of the spiking neurons when converting the input signal. The main function of this flag is to determine the nature of the input value by controlling the emission of spikes. Specifically, when the input value is positive, the spiking flag will indicate that the spike is not output, indicating that the input signal is a positive-going spike. In contrast, when the input value is negative, the spiking flag will trigger the output of the spike, indicating that this is a negative spike.

In conventional SNNs, positive and negative signals are typically distinguished via excitatory and inhibitory synapses, which necessitates maintaining two distinct synapse types and complicates hardware implementation. Moreover, inhibitory and excitatory spikes lack explicit differentiation and rely on predefined synaptic configurations. In contrast, the "spiking flag" serves as a binary marker that directly encodes the polarity of input signals within a single spike train. This design eliminates the need for separate synapse types, circumvents dual synaptic mechanisms, and unifies all synaptic weights as positive values, thereby significantly simplifying circuit design.

3.4 Multi-head Spike Encoding scheme

Adding a time window to the text introduces the time dimension. As shown in Fig.2, by setting a fixed time window, the input is converted into a spike signal within this window, and then the relevant spike calculation is performed.

In this process, the average spike intensity of T time steps represents the value of the input value. As shown in the Fig.2, the input is converted into the average spike intensity of l layers of neurons within T time steps, and the number of neurons in each layer of the neuron model corresponds to the characteristic dimension of the input vector. During the conversion process, the spiking flag is also initialized to ensure that its state can accurately reflect the characteristics of the input signal.

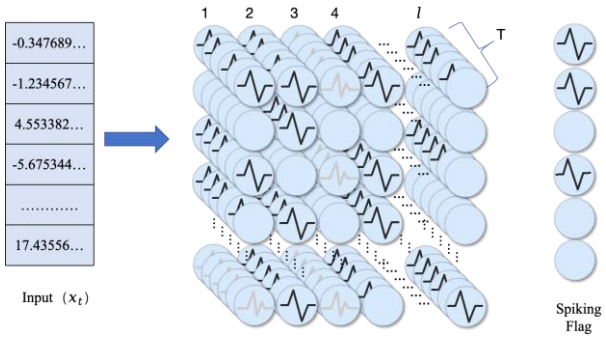


Fig. 2. spiking conversion framework.

The spike intensity of each layer per time step in T time steps is different, which represents different spike intensities and enhances the robustness of the model. The equivalent relationship is:

$$x_t = \frac{\sum_{j=1}^{T}\sum_{i=1}^{l} dw_i'^{j}(x_t) \odot d'_f(x_t)}{T} \tag{3}$$

$$dw_i'^{j}(x_t) = d_i'^{j}(x_t) * W_i \tag{4}$$

Where, $dw_i'^{j}(x_t)$ represents the weighted spike output intensity of the i-th layer j-th time step for the input x_t at time t. $d'_f(x_t)$ represents the spiking flag bit for the input x_t at time t. T is the set spike statistical time, the value of the time window. $d_i'^{j}(x_t)$ is the spike output of the i-th layer j-th time step for the input x_t at time t, and the W_i is the numerical compression weight in the expression process.

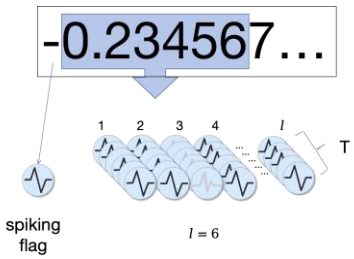


Fig. 3. Multi-head spike conversion example.

During the input conversion process, the transformation is performed according to Eq.(3) and Eq.(4), employing the multi-head spike encoding method to convert the first l digits of the numerical value. This ensures an l-digit precision, where the value is represented by l layers of spikes over T time steps to guarantee accuracy. For instance, in Fig.3, the number -0.234567⋯ is encoded using 6 layers of spikes (l=6) over T time steps to represent the 6-digit value 0.23456, while the negative sign is converted into a spiking flag.

The parameter l denotes the bit-width for input value decomposition. For instance, when l=6, the input value is decomposed into 6 bit segments (2 integer bits and 4 fractional bits), with each segment encoded via distinct spike trains. The product $T \times l$ quantifies the precision level of the spike-based representation, where a larger $T \times l$ value corresponds to higher encoding precision while inducing linear growth in energy consumption.

In rate coding, the "average frequency" represents a temporal statistic (e.g., spike count per 100ms window), whereas SpikeRWKV's "average intensity" constitutes a spatial statistic (e.g., mean amplitude across multiple spike layers). When T=10ms and l=3, the three spike layers fire in parallel within the 10ms window rather than sequentially over 30ms. SpikeRWKV dynamically balances precision and energy consumption by adjusting the layer depth (l). Furthermore, the spiking flag mechanism replaces traditional inhibitory pathways, thereby conserving synaptic resources.

3.5 Spiking Time Mixing

The token shift enables the model to independently and uniquely allocate new and old information to the reception channels, keys, values, and gating vectors (denoted as r, k, v, and g, respectively) at each time step for each head. First, a token shift operation is performed on the input, and then linear interpolation is performed on the input data. The data dependent linear interpolation (ddlerp) [19] between x_t and x_{t-1} used in token shift is be computed. Then, we still use the subsequent Time Mixing method in RWKV6 for calculation.

In the actual process of spiking token shift, the input value is first converted into an input spike signal by Multi-head Spike Encoding scheme, and then when the input spike signal is transmitted to the spiking neuron, it is multiplied by the synaptic weight of the input to become a weighted spike signal. Finally, the weighted sum of the spike output of the layer l-th time step is counted by the weighted spike counters, and then the weighted sum of the spikes of each layer in the time period from 0 to T time steps is counted.

4 Experiment

4.1 Experimental Setup

We evaluate the performance of SpikeRWKV on Natural Language Understanding (NLU). We evaluated the effectiveness of the model by the performance of ablation experiments on Natural Language Generation (NLG). For NLU tasks, we assess its performance on four benchmark datasets: AI2 Reasoning Challenge (ARC) [20], Choice of Plausible Alternatives (XCOPA) [21], Physical Interaction QA (PIQA) [22], and Winogrande (Wino) [23]. For NLG tasks, we chose the following 2 classic text classification datasets to evaluate the text generation performance of SpikeRWKV: WikiText-2 [24] and WikiText-103 [24].

For NLU evaluation, we employed a 5-shot method [25], where questions were grouped into sets of five. Each group contained four answered questions followed by one test question, designed to explicitly demonstrate the expected response pattern to the model. This approach ensures reliable assessment of the model's practical capabilities by providing contextual examples while maintaining evaluation rigor.

We evaluated three SpikeRWKV configurations,noted as SpikeRWKV-3L (T$\times$ l=1*3) , SpikeRWKV-4L (T$\times$ l=1*4) and SpikeRWKV-5L (T$\times$ l =1*5) numerical precisions of 3-digit, 4-digit and 5-digit representations. All operations assume a 32-bit floating-point implementation on 45nm technology. To evaluate the performance of our model, we calculate its bits-per-character (BPC) and perplexity (PPL) metrics.

4.2 Ablation Studies

A summary of results are provided in Table 1. This includes the BPC and PPL achieved on NLG tasks using SpikeRWKV tested on WikiText-103 and WikiText-2 compared

to several baselines, including RWKV and SNN-RWKV. The RWKV model is a baseline model. The SNN-RWKV model is the result of simulating RWKV using traditional SNN.

The experimental results demonstrate significant variations in model performance across different architectural configurations in Table 1, as measured by both perplexity (PPL) and bits-per-character (BPC) metrics. We believe that in SpikeRWKV, $T \times l$ is an indicator of expression accuracy, so the expression accuracy of T=3, L=1 is consistent with that of T=1, L=3. The baseline RWKV model establishes the performance upper bound with 12.73 PPL and 3.4119 BPC on WikiText-103, showcasing the inherent capability of continuous-value architectures in language modeling tasks.

Notably, the SNN-RWKV exhibits substantial performance degradation, particularly in the SNN-RWKV-3 configuration which yields a 31% higher PPL (16.67) compared to the baseline. This performance gap gradually narrows with increased model depth, as evidenced by the SNN-RWKV4 variant's improved 13.59 PPL, suggesting partial mitigation of information loss through deeper network architectures.

The proposed SpikeRWKV architecture achieves remarkable performance recovery, with the 4-bits configuration (SpikeRWKV-4L) attaining 12.12 PPL - merely 4.8% higher than the continuous baseline while maintaining comparable BPC performance (3.4118 vs 3.4119). It demonstrates the efficacy of the proposed Multi-head Spike Encoding scheme in preserving fine-grained linguistic information during spike-based computation. The results further reveal an optimal depth configuration, where the SpikeRWKV-4L outperforms both shallower and deeper variants across both evaluation metrics.

4.3 Result on Natural Language Understanding

Our comprehensive evaluation across multiple NLU benchmarks reveals critical insights into the performance characteristics of various neural architectures in Table2. The SNN-RWKV implementations reveal important transitional characteristics, with the SNN-RWKV-3 variant maintaining competent linguistic task performance (56.38% on Winograd_dev) while struggling with knowledge-intensive tasks (25.93% on ARC_test), and the SNN-RWKV-4 version showing measurable improvements in commonsense reasoning (+3.2% on PIQA_train) but demonstrating instability in other domains.

Our proposed SpikeRWKV architecture demonstrates significant advancements, as the SpikeRWKV variant matches or exceeds baseline performance in 50% of evaluation metrics, while the 5t model establishes new state-of-the-art performance on Winograd_train (59.37%), outperforming all baseline models including RWKV. Crucially, SpikeRWKV achieves superior stability compared to traditional SNNs and shows particular strength in linguistic tasks, outperforming Mamba-2.8b by 37.37 absolute percentage points on Winograd benchmarks. These results demonstrate clear architectural trade-offs: deeper 5-layer spiking models excel in syntactic processing through enhanced temporal integration, shallower SpikeRWKV-3L versions better handle

Table 1. Performance comparison on WikiText-2 and WikiText-103 in terms of token-level Perplexity (PPL) and Bits Per Character (BPC) (lower is better for both metrics). T denotes the spike encoding time during inference, and L indicates the number of spike layers used in the SpikeRWKV conversion process. RWKV refers to the original baseline model, while SNN-RWKV represents an SNN-based variant of RWKV using repetition coding.

Method	Spiking	T	L	WikiText-103		WikiText-2	
				PPL	BPC	PPL	BPC
RWKV	×	-	-	12.73	3.4119	11.55	**3.4789**
SNN-RWKV-3	√	3	-	16.67	3.5243	13.67	3.7197
SNN-RWKV-4	√	4	-	13.59	3.5025	13.02	3.6874
SpikeRWKV-3L	√	3	1	12.90	**3.4115**	11.71	3.4792
SpikeRWKV-4L	√	4	1	**12.12**	3.4118	12.10	**3.4789**
SpikeRWKV-5L	√	5	1	12.53	3.4119	12.18	3.4790

Table 2. Accuracy results on Natural Language Understanding (NLU) tasks.

Method	ARC _test	ARC _train	XCOPA _test	XCOPA _val	PIQA _train	PIQA _valid	Wino _dev	Wino _train
RWKV	30.33	26.81	51.00	50.00	52.85	48.83	56.17	57.89
Bloom -3b[26]	18.20	21.23	45.00	50.00	29.50	29.90	45.70	45.50
Mamba -2.8b[27]	23.51	19.30	47.00	45.00	14.80	12.00	22.00	16.00
SNN-RWKV-3	25.93	25.87	50.00	50.00	52.96	45.21	**56.38**	45.45
SNN-RWKV-4	30.33	25.17	50.00	**55.00**	**56.16**	49.24	51.75	33.33
Spik-eRWKV-3L	29.67	**27.51**	**51.00**	45.00	47.22	**50.41**	52.73	50.00
Spik-eRWKV-4L	**33.63**	24.01	45.00	**55.00**	50.76	48.66	54.04	55.00
Spik-eRWKV-5L	33.41	25.87	45.00	**55.00**	47.14	50.15	56.25	**59.37**

knowledge tasks requiring precise timing, while intermediate SpikeRWKV-4L architectures offer optimal balance for general-purpose NLU applications.

Table 3. Energy consumption on NLU tasks measured in Picojoule (pJ). The metric $\mathtt{EE} = \frac{\mathtt{E_{ANN}}}{\mathtt{E_{SNN}}}$ quantifies the energy efficiency ratio between ANN and its SNN counterpart. All operations assume a 32-bit floating-point implementation on 45nm technology, with energy costs of E_{MAC} = 4.6 (pJ) and E_{AC} = 0.9 (pJ) [28].

	ANN	SNN-RWKV-3	SNN-RWKV-4	Spik-eRWKV-3L		Spik-eRWKV-4L		Spik-eRWKV-5L	
	E(pJ)	E(pJ)	E(pJ)	E(pJ)	EE	E(pJ)	EE	E(pJ)	EE
ARC_test	11776	11165.8	11202.3	2472.4	4.76	4536.2	2.59	6612.6	1.78
ARC_train	11776	9133.3	10786.3	2411.2	4.88	4474.6	2.63	6550.2	1.79
XCOPA_test	11776	7935.2	11200.2	2401.0	4.90	4465.7	2.63	6540.5	1.80
XCOPA_val	11776	10150.0	11018.2	2386.6	4.93	4451.0	2.64	6525.7	1.80
PIQA_train	11776	10209.7	11068.1	2385.5	4.93	4452.5	2.64	6527.8	1.80
PIQA_valid	11776	9991.4	9957.0	2376.0	4.95	4437.9	2.65	6512.5	1.80
Wino_dev	11776	10685.0	11127.4	2429.5	4.84	4493.1	2.62	6567.6	1.79
Wino_train	11776	10786.3	11494.2	2436.5	4.83	4501.4	2.61	6576.5	1.79
Avg	**11776**	**10007.0**	**10981**	**2412.3**	4.88	**4476.5**	2.63	**6551.6**	1.79

A detailed definition of energy consumption is given in [29]. Since the addition of SNN is mainly to simplify and replace the input weighting process, the energy consumption of ANN is defined as:

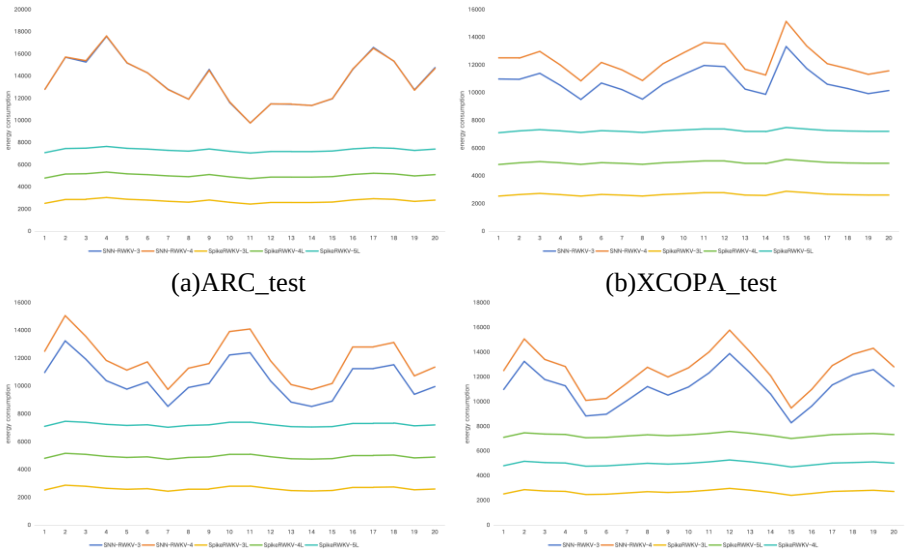
$$E_{ann} = C_{in} \times E_{MAC} \tag{5}$$

Among them, E_{ann} is the total energy consumption of ann in this weighted process. C_{in} denotes the number of input channel. E_{MAC} represents the multiplication and accumulation energy consumption.

$$E_{snn} = C_{in} \times fr \times T \times l \times E_{AC} \tag{5}$$

fr denotes the average spike firing rate. T is the time steps. l denotes the number of layers of spikes. E_{AC} represents the accumulated energy consumption. It should be noted that the current energy estimation methodology only accounts for fundamental computational energy consumption, while excluding memory access energy and energy efficiency impacts from hardware architectural optimizations such as input-sharing or weight-sharing mechanisms [30].To facilitate comparison between E_{ann} and E_{snn}, we

define an energy efficiency ratio (EE = E_{ann}/E_{snn}) to quantify their relative energy consumption. All operations assume a 32-bit floating-point implementation on 45nm technology, where E_{MAC} =4.6pJ and E_{AC} = 0.9pJ [28].



(a)ARC_test (b)XCOPA_test

(c)PIQA_valid (d)Wino_dev

Fig. 4. Energy consumption comparison between SNN-RWKV and SpikeRWKV on four datasets. For each dataset, 20 samples were randomly selected to assess energy usage.

In Table 3, traditional artificial neural networks (ANNs) exhibit the highest energy consumption, maintaining a constant energy usage of 11,776 pJ across all test tasks. In contrast, traditional spiking neural network implementations (SNN-RWKV series) demonstrate notable energy efficiency improvements. The SNN-RWKV-3 architecture achieves an average energy consumption of 10,007.0 pJ, while the SNN-RWKV-4 architecture consumes 10,981 pJ. Notably, SNN-RWKV-3 achieves the lowest recorded energy consumption in the XCOPA_test task, validating the inherent energy efficiency advantages of spiking encoding for specific tasks.

Second, the proposed SpikeRWKV architecture achieves breakthrough energy efficiency ratios. The improved SpikeRWKV-3L reduces average energy consumption to 2,412.3 pJ, equivalent to 20.5% of ANN energy consumption, while achieving an energy efficiency ratio (EE) of 4.88. This improvement is most pronounced in the ARC_train task, reaching an EE value of 4.88. Analysis of depth scaling shows that while absolute energy consumption increases with additional layers, all variants maintain energy efficiency ratios at least 1.79 better than the baseline ANN.

In summary, the proposed SpikeRWKV architecture demonstrates remarkable energy efficiency while maintaining competitive task performance. The SpikeRWKV-4L variant achieves the most balanced efficiency-performance tradeoff, reducing energy consumption by nearly 80% compared to ANNs while delivering comparable accuracy.

Notably, our architecture exhibits particularly strong performance on knowledge-intensive tasks (ARC, PIQA), where it maintains over 4.8 higher energy efficiency than conventional ANNs.

5 Conclusion

This paper proposes the SpikeRWKV model. The spikes are integrated into the large model through the encoding scheme of multi-head spike expression. Experimental results show that SNNs achieve significant energy efficiency improvements through spatiotemporal sparsity and event-based processing while maintaining performance levels comparable to ANNs. By converting traditional multiplication operations into spike operations and introducing quantization processing, we successfully reduced energy consumption in model calculations, thereby improving computational efficiency. SpikeRWKV is expected to be applied to neural chips to solve more complex NLP tasks. Although the SpikeRWKV model can maintain a certain performance in LLM tasks, even better than the performance of the baseline model and significantly reduce energy consumption, our model still has limitations. How to choose a good enough time steps and apply it to brain-like chips is a direction that still needs to be solved.

References

1. Tang, Z., Zhu, E.: Braintransformers: Snn-llm(2024),https://arxiv.org/abs/2410.14687
2. Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., Mian, A.: A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435 (2023)
3. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al.: A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology 15(3), 1–45 (2024)
4. Rane, N., Mallick, S., Kaya, , Rane, J.: Machine learning and deep learning architectures and trends: A review, pp. 1–38 (10 2024)
5. Yamazaki, K., Vo-Ho, V.K., Bulsara, D., Le, N.: Spiking neural networks and their applications: A review. Brain Sciences 12(7), 863 (2022)
6. Wang, S., Cheng, T.H., Lim, M.H.: A hierarchical taxonomic survey of spiking neural networks. Memetic Computing 14(3), 335–354 (2022)
7. Dampfhoffer, M., Mesquida, T., Valentian, A., Anghel, L.: Backpropagation-based learning techniques for deep spiking neural networks: A survey. IEEE Transactions on Neural Networks and Learning Systems (2023)
8. Auge, D., Hille, J., Mueller, E., Knoll, A.: A survey of encoding techniques for signal processing in spiking neural networks. Neural Processing Letters 53(6), 4693–4710 (2021)
9. Xing, X., Gao, B., Zhang, Z., Clifton, D.A., Xiao, S., Du, L., Li, G., Zhang, J.: Spikellm: Scaling up spiking neural network to large language models via saliencybased spiking. arXiv preprint arXiv:2407.04752 (2024)
10. Nunes, J.D., Carvalho, M., Carneiro, D., Cardoso, J.S.: Spiking neural networks: A survey. IEEE Access 10, 60738–60764 (2022)

11. Wang, Z., Fang, Y., Cao, J., Zhang, Q., Wang, Z., Xu, R.: Masked spiking transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1761–1771 (2023)
12. Kim, S., Park, S., Na, B., Yoon, S.: Spiking-yolo: spiking neural network for energy-efficient object detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11270–11277 (2020)
13. Bal, M., Sengupta, A.: Spikingbert: Distilling bert to train spiking language models using implicit differentiation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 10998–11006 (2024)
14. Comsa, I.M., Potempa, K., Versari, L., Fischbacher, T., Gesmundo, A., Alakuijala, J.: Temporal coding in spiking neural networks with alpha synaptic function. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8529–8533. IEEE (2020)
15. Comşa, I.M., Potempa, K., Versari, L., Fischbacher, T., Gesmundo, A., Alakuijala, J.: Temporal coding in spiking neural networks with alpha synaptic function: learning with backpropagation. IEEE transactions on neural networks and learning systems 33(10), 5939–5952 (2021)
16. Sakemi, Y., Yamamoto, K., Hosomi, T., Aihara, K.: Sparse-firing regularization methods for spiking neural networks with time-to-first-spike coding. Scientific Reports 13(1), 22897 (2023)
17. Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., Shi, L.: Direct training for spiking neural networks: Faster, larger, better. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 1311–1318 (2019)
18. Lv, C., Li, T., Xu, J., Gu, C., Ling, Z., Zhang, C., Zheng, X., Huang, X.: Spikebert: A language spikformer trained with two-stage knowledge distillation from bert. arXiv preprint arXiv:2308.15122 (2023)
19. Peng, B., Goldstein, D., Anthony, Q., Albalak, A., Alcaide, E., Biderman, S., Cheah, E., Du, X., Ferdinan, T., Hou, H., et al.: Eagle and finch: Rwkv with matrixvalued states and dynamic recurrence. arXiv preprint arXiv:2404.05892 (2024)
20. Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., Tafjord, O.: Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457 (2018)
21. Ponti, E.M., Glavaš, G., Majewska, O., Liu, Q., Vulić, I., Korhonen, A.: Xcopa: A multilingual dataset for causal commonsense reasoning. arXiv preprint arXiv:2005.00333 (2020)
22. Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al.: Piqa: Reasoning about physical commonsense in natural language. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 7432–7439 (2020)
23. Sakaguchi, K., Bras, R.L., Bhagavatula, C., Choi, Y.: Winogrande: An adversarial winograd schema challenge at scale. Communications of the ACM 64(9), 99–106 (2021)
24. Merity, S., Xiong, C., Bradbury, J., Socher, R.: Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843 (2016)
25. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al.: Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115 (2024)
26. Workshop, B., Scao, T.L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A.S., Yvon, F., et al.: Bloom: A 176b-parameter openaccess multilingual language model. arXiv preprint arXiv:2211.05100 (2022)
27. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces (2024), https://arxiv.org/abs/2312.00752

28. Horowitz, M.: 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)
29. Luo, X., Yao, M., Chou, Y., Xu, B., Li, G.: 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)
30. Panda, P., Aketi, S.A., Roy, K.: Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization. Frontiers in Neuroscience 14, 653 (2020)