YOLO-SNN: A SPIKING NEURAL NETWORK FOR ENERGY-EFFICIENT SAR TARGET DETECTION WITH SPIKING GHOST CONVOLUTION

Hongyu Liu[1*], Danzhenluobu[1], Zouzi Xia[1], Zihan Wang[1,2], Xinyi Ye[1,2], Bingchen Zhang[1,2]

[1]*School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China*
[2]*National Key Laboratory of Microwave Imaging, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China*

⋆liuhongyu226@mails.ucas.ac.cn

Keywords: SPIKING NEURAL NETWORK, SPIKING GHOST CONVOLUTION, SYNTHETIC APERTURE RADAR, TARGET DETECTION, DEEP LEARNING

Abstract

Synthetic Aperture Radar (SAR) target detection is crucial for maritime surveillance and disaster monitoring. While deep learning approaches achieve high accuracy, their computational complexity and energy consumption often hinder deployment on edge devices. Spiking Neural Network (SNN), inspired by biological neurons, provides a promising alternative by leveraging sparse, event-driven spike-based communication to reduce power consumption. This paper presents YOLO-SNN, an energy-efficient spiking neural network model that combines the detection accuracy of YOLO architectures with the low-power advantages of SNN. To leverage the computational efficiency of Ghost convolutions while further reducing YOLO-SNN's computational consumption, we design a Spiking Ghost Convolution module through spiking Depthwise Separable Convolution, and selectively integrate it into YOLO-SNN. Experiments on the HRSID dataset show that our model achieves a mean average precision (mAP@0.5:0.95) of 61.9% with 19.4M parameters, outperforming existing SNN-based detectors. Remarkably, YOLO-SNN reduces energy consumption by nearly 6× compared to conventional artificial neural networks with comparable accuracy. This work validates the effectiveness of SNN-based models for energy-efficient SAR target detection on edge computing platforms.

1 Introduction

Synthetic Aperture Radar (SAR) enables all-weather, day-night monitoring, making it invaluable for disaster prevention and ocean observation. As SAR systems advance, the growing volume of SAR imagery highlights the need for efficient target detection to extract critical object information. Developing accurate detection methods is crucial for maximizing SAR data utility and improving disaster response capabilities.

Traditional SAR target detection methods, such as the Constant False Alarm Rate (CFAR) algorithm, primarily rely on handcrafted features and are susceptible to complex backgrounds and noise interference, leading to poor accuracy, low efficiency, and weak generalization[1].In contrast, deep learning methods represented by Artificial Neural Network (ANN) can autonomously learn image features, exhibiting strong recognition and anti-interference capabilities, and have achieved significant progress in SAR target detection[2]. However, the performance improvement of ANNs often comes with increased network complexity, a surge in parameters and computational costs, and higher demands for computing power and energy consumption. For instance, the FCCD-SAR algorithm, specifically designed by Dong et al. for SAR target recognition, achieves high accuracy with 2.72 million parameters and 6.11 GFLOPs [3]. In edge device deployment scenarios, these high-power and resource-intensive characteristics become major bottlenecks.

For UAV-based and spaceborne SAR perception tasks with stringent power constraints, simply stacking deeper ANNs can no longer meet the triple requirements of high accuracy, low energy consumption, and compact form factor. The third-generation neural network—Spiking Neural Network (SNN)—achieves highly efficient computation through event-driven, sparse spike transmission[4], making it inherently suited to these demands. Globally, major initiatives are driving progress: The EU's Human Brain Project [5], the U.S. BRAIN 2.0 [6], and China's Brain Science and Brain-Inspired Intelligence Project (2030)have established infrastructure, structural/functional brain mapping, and policy-backed research, respectively. These efforts collectively advance SNN's algorithmic and hardware ecosystem. Neuromorphic chips like TrueNorth [7], Loihi [8], Tianjic [9], and Wentian [10] demonstrate two orders of magnitude lower power than conventional processors.

SNNs have emerged as a promising approach for SAR target detection due to their energy efficiency advantages. Yoo et al.[11] developed YOLO-SNN by replacing YOLOv5's convolutions with spiking equivalents and integrating a C3Ghost module, achieving comparable accuracy to ANNs on HRSID/SAR-Ship datasets with lower power consumption. However, their direct ANN-to-SNN conversion of the C3Ghost module (originally designed for ANNs[12]) lacks architecture-level optimization especially for SNN models. Current approaches also insufficiently address the accuracy-efficiency trade-off in SAR-specific model design.

To address these challenges, we propose a YOLO-SNN model incorporating Spiking Ghost Convolution. First, inspired by the lightweight GhostNet[13] architecture designed for ANNs, we introduce Spiking Ghost Convolution optimized for SNN models. This approach utilizes spiking Depthwise Separable Convolution to replace conventional spiking convolutions, thereby achieving SNN model lightweighting. Second, our proposed YOLO-SNN model retains the low-power advantages of EMS-YOLO[14] and the high detection accuracy of YOLO architectures while reducing parameter count by 69% through selective replacement of standard spiking convolutions with Spiking Ghost Convolutions. The model achieves an mAP@0.5:0.95 of 61.9% on the HRSID dataset[15] with only 19.4M parameters,demonstrating an excellent trade-off between detection accuracy and model lightweighting.

Our contributions in this paper are summarized as follows:

- **YOLO-SNN Model**: We propose the YOLO-SNN model, which combines the low power consumption feature of EMS-YOLO with the high detection accuracy of YOLOv5. By employing a hierarchical selective replacement strategy (replacing conventional spiking convolution with Spiking Ghost Convolution in specific modules), the model significantly reduces the number of parameters and computing while maintaining comparable detection accuracy. In the SAR target detection task on the HRSID dataset, this model outperforms most existing SNN methods.
- **Spiking Ghost Convolution**: We design spiking Ghost convolution, adapting the Ghost convolution from ANNs to the spiking domain by generating redundant features through spiking Depthwise Separable Convolution,and replace traditional spiking convolutions,trading off slight accuracy reduction for significant parameter savings.

2 Methodology

In this section, we will introduce our method in three parts,SNN,YOLO-SNN and Spiking Ghost Convolution.

2.1 SNN

SNN, regarded as the third generation of neural network, mimics the dynamic properties of biological neurons, offering unique advantages in temporal data processing, event-driven computation, and energy efficiency. Unlike traditional ANNs that rely on continuous-value propagation, SNNs encode information through discrete spikes (0/1 signals), making them inherently suitable for neuromorphic hardware and low-power real-time systems[16]. The dynamics of SNNs are governed by neuron models, with the Leaky Integrate-and-Fire (LIF) model being the most widely adopted. The LIF model describes the accumulation and leakage of membrane potential through a differential equation:

$$\tau \frac{\mathrm{d}u}{\mathrm{d}t} = -(u - u_{\mathrm{rest}}) + I_{\mathrm{in}}(t) \tag{1}$$

where $u(t)$ is the membrane potential, τ is the membrane time constant, u_{rest} is the resting potential, and $I_{\mathrm{in}}(t)$ is the input current. When $u(t) \geq u_{\mathrm{th}}$, the neuron emits a spike and resets to u_{rest}. For hardware implementation, the LIF model is discretized into a timestep iteration:

$$u[t+1] = u_{\mathrm{rest}} + (u[t] - u_{\mathrm{rest}}) \cdot e^{-\Delta t/\tau} + I_{\mathrm{in}}[t] \tag{2}$$

Compared to simplified models, the LIF model better captures biological realism, including membrane potential leakage and threshold-triggered spiking.

SNNs excel in edge computing and real-time tasks. For example, a lightweight SAR ship recognition algorithm (*SpikingSAR*) integrates visual attention mechanisms and Poisson encoding, achieving 95.58% F1-score with energy consumption reduced to $1/1000$ of ANN[17]. Furthermore, memristors enhance SNN adaptability by emulating synaptic plasticity, enabling dynamic weight adjustment and on-chip learning[18].

However, the non-differentiability of spike firing events in SNNs poses a significant challenge for their training. Currently, the mainstream training methods for SNNs can be primarily categorized into the following types:

- ANN-to-SNN Conversion: This method first trains an equivalent, high-performance ANN model, and then maps its weights and structure to an SNN using specific conversion rules. Its advantage lies in circumventing the difficulties of direct SNN training and inheriting the high accuracy of ANNs. The disadvantages are that it usually requires long simulation time steps, leading to high inference latency, and has limited adaptability to dynamic temporal tasks.
- Unsupervised Learning: Represented by Spike-Timing-Dependent Plasticity (STDP)[19], this method adjusts synaptic weights based on the precise timing of pre- and post-synaptic neuron spikes. This approach has strong biological plausibility and is suitable for online learning, but it generally struggles to handle complex supervised learning tasks independently and offers limited performance.
- Direct Training Based on Backpropagation: This is currently the method that achieves the highest performance on visual tasks. Its core idea is to optimize network weights using the Backpropagation Through Time (BPTT) algorithm. To overcome the non-differentiability of the spike function, the Surrogate Gradient method is widely adopted[20]. During the forward pass, this method uses the actual, non-differentiable spike firing function, while during the backward pass for gradient calculation, it uses a smooth, differentiable surrogate function to approximate the derivative of the spike function.

This paper adopts the direct training method based on surrogate gradients to train the YOLO-SNN model. This approach enables us to optimize the network end-to-end, achieving detection accuracy comparable to ANNs while maintaining the event-driven and low-power characteristics of SNNs. Our training strategy is

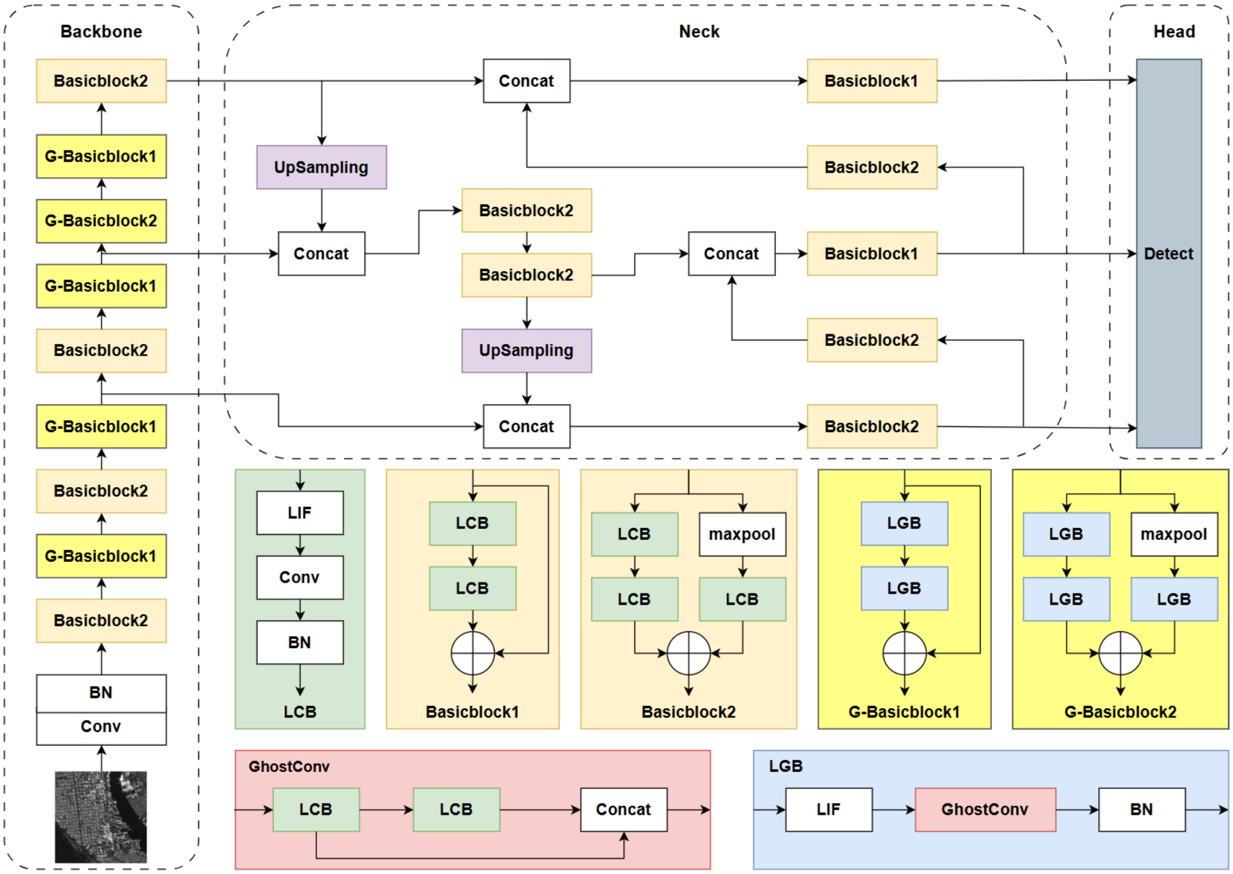


Fig. 1 The main structure of YOLO-SNN model,which is consisted of backbone,neck and head.The spiking convolutional operations are implemented by BasicBlock1 , BasicBlock2 and their Ghost version G-BasicBlock1, G-BasicBlock2. BasicBlock2 and G-BasicBlock2 are used when the number of output channels differs from that of the input channels, while BasicBlock1 and G-BasicBlock1 are employed in all other cases.

to first train the network with a time step of T=1, and then use the obtained weights as a pre-trained model for fine-tuning with multiple time steps. This strategy not only accelerates training convergence but also yields superior performance.

2.2 YOLO-SNN

The YOLO-SNN model is based on the YOLO architecture, integrating SNN components to achieve energy-efficient object detection. The model structure (shown in Fig. 1) consists of three main modules: the *Backbone* for feature extraction, the *Neck* for feature fusion, and the *Detection Head* for multi-scale object localization and classification.

The backbone network adopts the architecture of YOLOv5 while implementing spiking convolutional operations through BasicBlock1 and BasicBlock2 modules. The Spatial Pyramid Pooling Fast (SPPF) module, which serves as a crucial component in YOLOv5 for multi-scale feature extraction and feature fusion enhancement, demonstrates suboptimal performance in its SNN implementation. Consequently, we exclude the spiking version of SPPF from YOLO-SNN due to its observed degradation in detection accuracy. Furthermore, the architecture of the Backbone without employing Spiking Ghost Convolution modules is illustrated in Fig. 2. The Backbone consists of 10 layers (L1-L10), with L8 (BasicBlock1) and L9 (BasicBlock2) exhibiting notably higher parameter counts. L2, L4, L6, and L8 handle downsampling, while L3, L5, L7, and L9 refine the downsampled features. To minimize the model parameters while preserving the downsampling capability of the backbone network, we employ G-BasicBlock1 and G-BasicBlock2 modules to replace the conventional blocks in layers 3, 5, 7, 8, and 9 of the backbone network. The modified backbone generates feature maps at three distinct scales - 1/8, 1/16, and 1/32 of the input resolution - effectively capturing hierarchical spatial information for targets across varying sizes.

The Neck adopts the same FPN+PAN architecture as YOLOv5, ensuring structural consistency in cross-scale feature fusion. The Feature Pyramid Network (FPN)[21] enhances semantic representation by propagating high-level features to lower-resolution layers via a top-down pathway. The Path Aggregation Network (PAN)[22] improves localization accuracy by aggregating low-level positional features into higher-resolution layers through a bottom-up pathway.

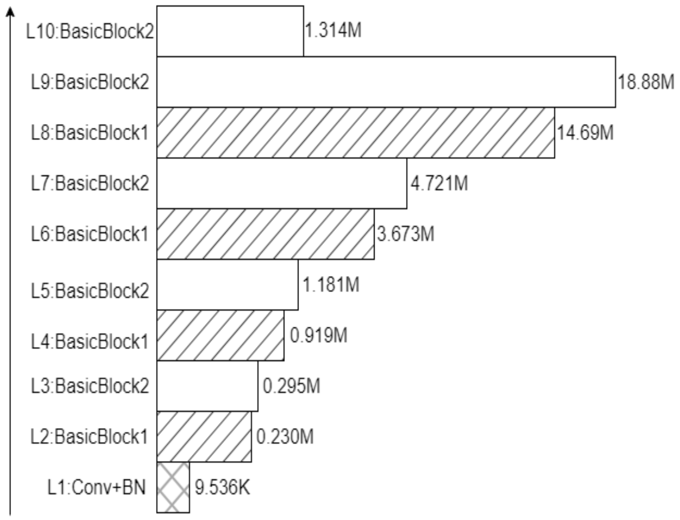


Fig. 2 Hierarchical structure of the YOLO-SNN backbone network before employing Spiking Ghost Convolution, with the parameter count of each layer labeled in the figure.

For the detection head, we retain the original YOLOv3-style architecture from EMS-YOLO to maintain its proven multi-scale detection capability for targets of varying sizes. While our YOLO-SNN nomenclature reflects the overall architectural inspiration from YOLOv5's efficient design paradigm, this specific component preservation ensures compatibility with the baseline model's performance characteristics.

2.3 Spiking Ghost Convolution

Traditional convolutional neural networks (CNNs) typically require a large number of parameters and intensive floating-point operations to achieve high detection accuracy. To explore lightweight and efficient network architectures suitable for edge and mobile computing, Huawei Noah's Ark Lab proposed the Ghost convolution module in 2020. This module replaces complex convolution operations with inexpensive linear transformations to generate redundant features. Compared to conventional convolutions with the same input and output channel dimensions, the Ghost module reduces both parameters and floating-point operations by approximately 30%, improving network efficiency while maintaining low parameter counts, making it ideal for resource-constrained environments[13].

Inspired by the Ghost convolution module, we designed a Spiking Ghost Convolution module for SNNs. The Spiking Ghost Convolution module operates in three steps:

Step 1: Spiking Depthwise Convolution – A standard spiking depthwise convolution generates initial feature maps.

Step 2: Spiking Pointwise Convolution – A low-parameter spiking pointwise convolutiuon is applied to the initial feature maps generated by Step 1 to produce redundant feature maps.

Step 3: Channel Concatenation – The feature maps from Steps 1 and 2 are concatenated along the channel dimension to form the final output.

In this work, we set the number of redundant feature maps equal to the initial feature maps.In Fig. 1, the module labeled 'GhostConv' represents the proposed Spiking Ghost Convolution module which can replace traditional spiking convolutions in the LCB (Leaky Integrate-and-Fire Convolution Block) module, forming an LGB (Leaky Integrate-and-Fire Ghost Block) module. Consequently, BasicBlock1 can BasicBlock2 can be converted into their respective Ghost variants. By systematically integrating the Spiking Ghost Convolution module into the YOLO-SNN architecture, we optimize the model for efficiency.

3 Result

3.1 Experimental Setup

*3.1.1 : Dataset:*The HRSID dataset is a ship detection dataset based on SAR images, composed of high-resolution SAR images captured by the GF-3 (GaoFen-3) and Sentinel-1 satellites.It contains 5,604 images and 16,951 annotated ship instances, covering various sea conditions, polarization modes, and imaging angles.The dataset contains a single class (ship) with annotated bounding boxes for each object.

*3.1.2 : Implementation Details:*All experiments were conducted on a vGPU with 32GB memory. The model was trained using the SGD optimizer with a weight decay of 0.0005. The initial learning rate was set to 0.1 and decayed to 0.01 via cosine annealing. The network was trained for 300 epochs on the HRSID dataset with a batch size of 8.We selected T=2 as the time step of SNN models, determined by balancing detection accuracy against power consumption for YOLO-SNN. The neuronal parameters of YOLO-SNN were configured as follows: the reset potential V_{reset} of LIF neurons was set to 0, the membrane time constant τ to 0.25, and the firing threshold potential V_{th} to 0.5. All SNN models in the experiments were first trained with time step $T = 1$, and the optimal weights obtained from $T = 1$ training were used as pretrained weights for multi-time-step training. This approach not only accelerated SNN training but also achieved better performance.

3.2 Evaluation Metrics

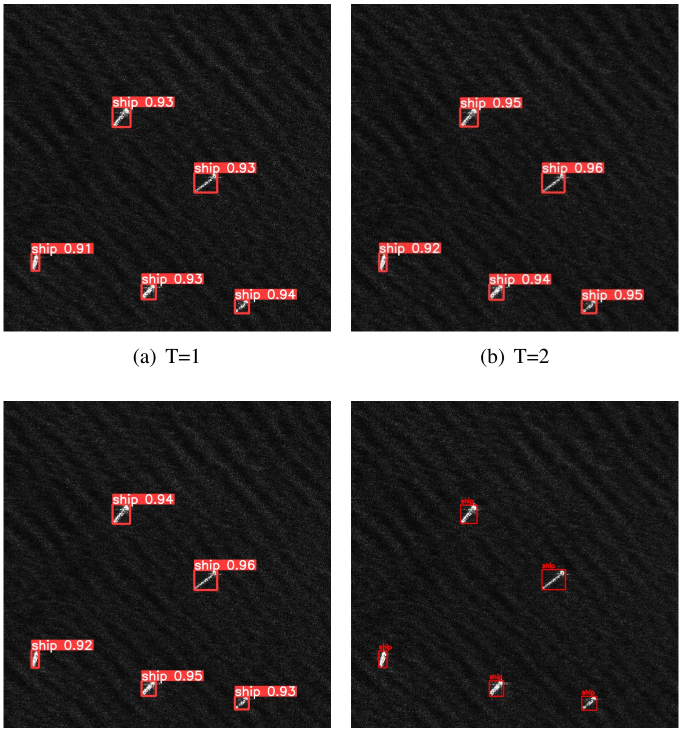
For target detection tasks,mean Average Precision(mAP) is the most widely employed model evaluation metrics.We selected mAP at IOU=0.5(mAP@0.5)and mAP at IOU between 0.5 and 0.95(mAP@0.5:0.95)as evaluation metrics.Additionally, as an improved version of the YOLO series, we also employed precision and recall metrics.Precision measures the model's ability to avoid false positives, calculated as the ratio of correctly predicted positive observations (True Positives) to all predicted positives (True Positives + False Positives). High precision indicates fewer false

Table 1 Comparison with previous work on HRSID dataset

Architecture	Method	Params	Time Step	Precision(%)	Recall(%)	mAP@0.5(%)	mAP@0.5-0.95(%)
ANN	YOLOv5	1.7M	/	86.2	77.2	86.0	62.2
	YOLOv8	3.3M	/	85.8	79.6	88.4	64.8
	DETR[23]	41.3M	/	/	/	83.5	52.1
Directly-Trained SNN	EMS-YOLO	15.6M	2	88.8	76.0	84.7	58.7
	Spike YOLO[24]	65.5M	2	86.2	79.1	85.8	64.1
	YOLO-SNN	19.6M	2	87.1	79.6	84.9	61.9

Table 2 Ablation Studies of Architecture Design. (Performance Metrics: GhostConv Replacement Strategies)

Method	Params (M)	mAP@0.5(%)	mAP@0.5:0.95(%)
YOLO-SNN	19.6	85.2	61.4
Add SPPF-SNN	22.2	85.8	61.1
Replace Layers 3,5,7,8,9 → All replacement	11.8	83.2	56.5
Replace Layers 3,5,7,8,9 → Replace all BackBone	17.8	85.4	59.8
Replace Layers 3,5,7,8,9 → Replace Layers 2,4,6,8,9	23.6	85.4	60.5



(a) T=1

(b) T=2

(c) T=3

(d) Ground Truth

Fig. 3 Target detection results of YOLO-SNN on HRSID dataset

alarms.Recall evaluates the model's capability to identify all relevant targets, defined as the ratio of correctly predicted positives to all actual positives (True Positives + False Negatives). High recall reflects fewer missed detections.

3.3 Comprehensive Evaluation

Based on the comparison results in Table 1, it is observed that conventional ANN-based detectors such as YOLOv5 and YOLOv8 achieve superior detection accuracy (mAP@0.5:0.95 of 62.2% and 64.8%, respectively) with fewer parameters (1.7M and 3.3M) compared to our YOLO-SNN model. This highlights the inherent challenge for SNN-based models to match the precision of their ANN counterparts due to the discrete and sparse nature of spike-based computation.YOLO-SNN achieves an mAP@0.5:0.95 of 61.9%, outperforming EMS-YOLO by 3.2%. Meanwhile, YOLO-SNN employs only 19.6 million parameters, which is 3.3 times fewer than the 65.5 million parameters used in Spike YOLO. Notably, our model surpasses traditional ANNs-based YOLOv5 (86.2%) and YOLOv8 (85.5%) in Precision (87.1%). Furthermore, it achieves the highest recall rate (79.6%) among all compared methods, indicating its exceptional capability in minimizing false negatives. This combination of lightweight architecture, competitive accuracy, and robust detection performance makes YOLO-SNN particularly suitable for resource-constrained edge applications requiring reliable object coverage. For our proposed YOLO-SNN,the optimal time step of T=2, determined by trading off detection accuracy and power consumption, is reported in Table. 3.

3.4 Size of Time Step

The choice of an appropriate time step (T) is crucial for balancing the performance and energy efficiency of SNNs. While longer time steps generally improve model accuracy, they inevitably lead to higher power consumption. To evaluate this trade-off, we conducted comprehensive experiments on the HRSID dataset, assessing the detection accuracy and power consumption of our YOLO-SNN model across $T = 1$, 2, 3, and 4.The energy consumption of SNNs and ANNs was calculated according to Equations (3) and (4).

$$E_{\mathrm{SNN}} = T f_r O^2 C_{\mathrm{in}} C_{\mathrm{out}} k^2 E_{\mathrm{AC}} \tag{3}$$

$$E_{\mathrm{ANN}} = O^2 C_{\mathrm{in}} C_{\mathrm{out}} k^2 E_{\mathrm{MAC}} \tag{4}$$

where O represents the feature output size, C_{in} and C_{out} denote the number of input and output channels respectively, k is the kernel size, f_r indicates the average spike firing rate, and T is the timestep. Our evaluation methodology follows the most widely-adopted energy consumption assessment framework in the SNN research community[25][26]. All operations were modeled assuming 32-bit floating-point implementation on 45 nm technology nodes, with $E_{MAC} = 4.6$ pJ and $E_{AC} = 0.9$ pJ[27].

Table 3 Impact of different time steps on HRSID

Metric	**T=1**	**T=2**	**T=3**	**T=4**
mAP@0.5:0.95	0.593	0.619	0.614	0.618
Precision	0.882	0.895	0.873	0.878
Energy(mJ)	22.5	41.8	64.2	89.5

The precision and energy consumption performance of the YOLO-SNN model at different time steps are shown in Table. 3. The power consumption increases approximately linearly with the number of time steps, while the detection accuracy , considering both mAP@0.5:0.95 and Precision metrics , reaches its peak at T=2. After balancing accuracy and power consumption, we determined that the YOLO-SNN model performs best at T=2.

To further illustrate the energy efficiency of YOLO-SNN, we compare its energy consumption with that of YOLOv5. According to Equations (4), YOLOv5 consumes approximately 243 mJ per inference. In contrast, YOLO-SNN at T=2 consumes only 41.8 mJ, which is about one-sixth of the energy required by YOLOv5, while still achieving a competitive mAP@0.5:0.95 of 61.9%. This substantial energy saving underscores the practical value of YOLO-SNN in resource-limited edge applications, where energy consumption is a more critical constraint than peak accuracy.

3.5 Ablation experiments

The ablation experiments primarily investigate whether the directly spiked ANN component (SPPF-SNN) maintains comparable performance in SNNs as in ANNs, and the effectiveness of different Spiking Ghost Convolution replacement strategies, verifying that replacing Layers 3,5,7,8,9 in YOLO-SNN achieves optimal results as illustrated in Table 2.

When incorporating the SPPF-SNN module into our baseline YOLO-SNN, the model's parameters increase from 19.6M to 22.2M while the mAP@0.5:0.95 slightly decreases by 0.3% (from 61.4% to 61.1%). This indicates that for high-precision detection tasks, the SPPF-SNN module introduces redundancy and should be removed.

In the Spiking Ghost Convolution replacement strategies: Full replacement (all conventional spiking convolutions) reduces parameters to 11.8M but causes significant mAP@0.5:0.95 degradation (56.5% vs 61.4%). And all-Backbone replacement maintains comparable parameters (20.4M) but yields lower accuracy (0.598). Finally alternative layer replacement (Layers 2,4,6,8,9) increases parameters to 26.2M with suboptimal accuracy (60.5%).

These results confirm that the original YOLO-SNN strategy (replacing Layers 3,5,7,8,9) achieves the best balance between model efficiency (19.6M parameters) and detection performance (61.4% mAP@0.5:0.95).

4 Conclusion

This study proposes YOLO-SNN, a novel spiking neural network framework specifically designed for SAR target detection, which combines the energy efficiency advantages of SNNs with the high detection accuracy of YOLO architectures. Our contributions include: The design of spiking Ghost convolution and its hierarchical selective replacement strategy, where standard spiking convolutions are replaced with spiking Ghost modules, achieving a 69% reduction in parameters; While YOLO-SNN does not exceed the accuracy of state-of-the-art ANN detectors such as YOLOv5 or YOLOv8, it achieves a favorable trade-off between detection performance and energy efficiency, making it a viable solution for energy-sensitive SAR target detection tasks.

For future work, we plan to deploy YOLO-SNN on the ”Wen-Tian” neuromorphic chip to further reduce energy consumption and enhance real-time processing capabilities, unlocking greater potential for onboard SAR target detection in practical scenarios.

5 Acknowledgement

This work was supported by University of Chinese Academy of Sciences Training Program of Innovation for Undergraduates(X202414430030).

6 References

[1] Q. Huang, W. Zhu, Y. Li, et al. “Survey of target detection algorithms in SAR images”. *IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2021, pp. 1756–1765.

[2] M. Yasir, W. Jianhua, X. Mingming, et al. “Ship detection based on deep learning using SAR imagery: a systematic literature review”. *Soft Computing* 27.1 (2023), pp. 63–84.

[3] X. Dong, D. Li, and J. Fang. “FCCD-SAR: A Lightweight SAR ATR Algorithm Based on FasterNet”. *Sensors* 23.15 (2023), pp. 6956.

[4] W. Maass. “Networks of spiking neurons: the third generation of neural network models”. *Neural Networks* 10.9 (1997), pp. 1659–1671.

[5] K. Amunts, C. Ebell, J. Muller, et al. “The Human Brain Project: Creating a European Research Infrastructure to Decode the Human Brain”. *Neuron* 92.3 (2016), pp. 574–581.

[6] National Institutes of Health. BRAIN 2.0: From Cells to Circuits, Toward Cures. 2023. URL: `https://braininitiative.nih.gov/vision/nih-brain-initiative-reports/brain-20-report-cells-circuits-toward-cures` (visited on 07/27/2023).

[7] F. Akopyan, J. Sawada, A. Cassidy, et al. “TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34.10 (2015), pp. 1537–1557.

[8] M. Davies, N. Srinivasa, T.-H. Lin, et al. “Loihi: A Neuromorphic Manycore Processor with On-Chip Learning”. *IEEE Micro* 38.1 (2018), pp. 82–99.

[9] J. Pei, L. Deng, S. Song, et al. “Towards Artificial General Intelligence with Hybrid Tianjic Chip Architecture”. *Nature* 572.7767 (2019), pp. 106–111.

[10] L. Tao, P. Li, and M. o. Meng. “Blended Glial Cell’s Spiking Neural Network”. *IEEE Access* 11 (2023), pp. 43566–43582.

[11] M. Yoo, J. Han, and S. Kim. “Deep Spiking Neural Network for Energy-Efficient SAR Ship Detection”. *IEEE Geoscience and Remote Sensing Letters* 22 (2025). Art no. 4006705, pp. 1–5.

[12] K. Han, Y. Wang, Q. Tian, et al. “GhostNetV2: Enhance Cheap Operation with Long-Range Attention”. *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 9969–9982.

[13] K. Han, Y. Wang, Q. Tian, et al. “GhostNet: More Features From Cheap Operations”. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 1577–1586.

[14] Q. Su, Y. Chou, Y. Hu, et al. “Deep Directly-Trained Spiking Neural Networks for Object Detection”. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 6532–6542.

[15] S. Wei, X. Zeng, Q. Qu, et al. “HRSID: A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation”. *IEEE Access* 8 (2020), pp. 120234–120254. ISSN: 2169-3536.

[16] S. Deng, H. Lin, Y. Li, et al. “Surrogate Module Learning: Reduce the Gradient Error Accumulation in Training Spiking Neural Networks”. *Proceedings of the 40th International Conference on Machine Learning*. PMLR. 2023, pp. 8182–8201.

[17] Y. Chen, S. Zhang, S. Ren, et al. “Gradual Surrogate Gradient Learning in Deep Spiking Neural Networks”. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 8927–8931.

[18] R. An, P. Gong, H. L. Wang, et al. “A Modified PSO Algorithm for Remote Sensing Image Template Matching”. *Photogrammetric Engineering and Remote Sensing* 76.4 (2010), pp. 379–389.

[19] S. R. Kheradpisheh et al. “STDP-based spiking deep convolutional neural networks for object recognition”. *Neural Networks* 99 (2018), pp. 56–67. ISSN: 0893-6080.

[20] E. O. Neftci, H. Mostafa, and F. Zenke. “Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks”. *IEEE Signal Processing Magazine* 36.6 (2019), pp. 51–63.

[21] T.-Y. Lin, P. Dollár, R. Girshick, et al. “Feature Pyramid Networks for Object Detection”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2117–2125.

[22] S. Liu, L. Qi, H. Qin, et al. “Path Aggregation Network for Instance Segmentation”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8759–8768.

[23] N. Carion, F. Massa, G. Synnaeve, et al. “End-to-End Object Detection with Transformers”. *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*. Vol. 12346. Lecture Notes in Computer Science. Springer, 2020, pp. 213–229.

[24] X. Luo, M. Yao, and Y. o. Chou. “Integer-Valued Training and Spike-Driven Inference Spiking Neural Network for High-Performance and Energy-Efficient Object Detection”. *Computer Vision – ECCV 2024*. Ed. by A. Leonardis et al. Vol. 15090. Lecture Notes in Computer Science. ECCV 2024. Cham: Springer, 2025, pp. 253–272.

[25] B. Yin, F. Corradi, and S. M. Bohté. “Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks”. *Nature Machine Intelligence* 3.10 (2021), pp. 905–913.

[26] P. Panda, S. A. Aketi, and K. Roy. “Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization”. *Frontiers in Neuroscience* 14 (2020), pp. 653.

[27] M. Horowitz. “1.1 computing’s energy problem (and what we can do about it)”. *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE. 2014, pp. 10–14.