EOS: An Energy-Oriented Attack Framework for Spiking Neural Networks

Ning Yang[1,2,†], Fangxin Liu[1,2,†], Zongwu Wang[1,2], Haomin Li[1,2], Zhuoran Song[1], Songwen Pei[3] and Li Jiang[1,2]

1. Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
2. Shanghai Qi Zhi Institute 3. University of Shanghai for Science and Technology

ABSTRACT

Spiking neural networks (SNNs) are emerging as energy-efficient alternatives to traditional artificial neural networks (ANNs). Their event-driven information processing significantly reduces computational demands while maintaining competitive performance. However, as SNNs are increasingly deployed in edge devices, security concerns have emerged. While significant research efforts have been dedicated to addressing the security vulnerabilities stemming from malicious input, often referred to as adversarial examples, the security of SNN parameters remains relatively unexplored. This work introduces a novel attack methodology for SNNs known as Energy-Oriented SNN attack (EOS). EOS is designed to increase the energy consumption of SNNs through the malicious manipulation of binary bits within their memory systems (i.e., DRAM), where neuronal information is stored. The key insight of EOS lies in the observation that energy consumption in SNN implementations is intricately linked to spiking activity. The bit-flip operation, the well-known Row Hammer technique, is employed in EOS. It achieves this by identifying the most robust neurons in the SNN based on the spiking activity, particularly those related to the firing threshold, which is stored as binary bits in memory. EOS employs a combination of spiking activity analysis and a progressive search strategy to pinpoint the target neurons for bit-flip attacks. The primary objective is to incrementally increase the energy consumption of the SNN while ensuring that accuracy remains intact. With the implementation of EOS, successful attacks on SNNs can lead to an average of 43% energy increase with no drop in accuracy.

ACM Reference Format:
Ning Yang[1,2,†], Fangxin Liu[1,2,†], Zongwu Wang[1,2], Haomin Li[1,2], Zhuoran Song[1], Songwen Pei[3] and Li Jiang[1,2]. 2024. EOS: An Energy-Oriented Attack Framework for Spiking Neural Networks. In *61st ACM/IEEE Design Automation Conference (DAC '24), June 23–27, 2024, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649329.3655981

1 INTRODUCTION

Deep artificial neural networks (ANNs) have witnessed success in tackling complex cognitive tasks, ranging from image processing to speech recognition and pattern analysis. However, this achievement has come at the cost of significant energy consumption, making the edge deployment of such ANNs challenging. An alternative avenue towards efficient computation lies in spiking neural networks (SNNs), which draw inspiration from biological learning frameworks. SNNs perform computations using binary spikes instead of analog activations found in conventional ANNs [19]. The inherent sparsity and event-driven nature of SNNs render them an attractive choice, particularly for resource-constrained applications in machine intelligence [12].

Within this landscape, the security and robustness of SNNs surface as critical concerns that merit rigorous investigation. Recent research has shown that adversarial attacks can induce malfunctions in SNNs by introducing controlled input noise of imperceptible magnitude[13][9]. However, while extensive efforts have been made to explore the security of SNNs concerning input perturbations, the security challenges related to the parameters of spiking neurons within the SNN have remained largely unexplored. One possible explanation for the limited focus on SNN parameter security is the prevailing perception of SNNs as robust systems, resilient to noise perturbations, owing to their transmission of information in the form of binary spike signals [19]. However, SNNs are frequently deployed in Internet of Things (IoT) systems, where resource-constrained platforms often lack effective data integrity checks [11, 17, 22]. Consequently, these deployed SNNs become vulnerable to well-known fault injection techniques such as row hammer and laser beam attacks [2, 6].

In SNNs, neurons maintain an internal state referred to as voltage, which changes in response to incoming spikes and the passage of time. When the voltage reaches a predefined threshold, the neuron fires a spike, transmitting information to connected neurons [19]. This event-driven propagation of information is contingent upon spike occurrences, and as a result, the energy consumption and execution time of SNN implementations are intricately linked to the network's spiking activity [7, 8, 22].

Building upon these foundational insights, this paper aims to explore parameter attacks on the firing thresholds stored within the spiking neurons of SNNs. These thresholds are intrinsically constrained due to their binary representation. To efficiently execute a bit-flip attack on SNNs, we introduce a novel attack framework called Energy-Oriented SNN attack (EOS). EOS is tailored to minimize accuracy degradation while maximizing energy consumption. It accomplishes this by identifying the most robust neurons within the SNN, with a particular focus on those related to firing thresholds, which are stored as binary bits in memory. Importantly, EOS operates independently of existing functional attacks on SNNs,

[†] These authors contributed equally.

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06…$15.00
https://doi.org/10.1145/3649329.3655981

which introduce adversarial perturbations affecting classification accuracy but not energy efficiency.

The main contributions of this paper are summarized as follows:

- **Enhancing SNN Security:** We expand the horizon of SNN security, moving beyond the traditional focus on input perturbations. Instead, we address parameter vulnerabilities while shifting the security paradigm from solely targeting model accuracy to strategically increasing SNN energy consumption within energy budget constraints. This approach offers a comprehensive and nuanced strategy to fortify SNNs against potential malicious attacks.
- **Energy-Oriented Attack:** We introduce EOS, an innovative attack framework designed to maximize energy consumption within SNNs. EOS selectively targets robust neurons associated with firing thresholds, thereby minimizing accuracy degradation while achieving significant energy increases.
- **Comprehensive Experimental Validation:** We evaluate the effectiveness of EOS across various SNN architectures and datasets. Our results demonstrate the successful execution of EOS attacks on SNNs, resulting in a 43% increase in energy consumption. Notably, these energy enhancements are achieved without any observable degradation in accuracy, underlining the undetectability of the approach.

2 BACKGROUND AND MOTIVATION

2.1 Spiking Neural Network

Different from ANNs, SNNs draw inspiration from the fact that biological neurons process information temporally by means of sparse spiking signals. SNNs emulate many biologically observed neuron behaviors, e.g., STDP[10], LIF[5]. In SNNs, the most popular spiking neuron model is the Integrate and Fire model (IF). An IF neuron, denoted as neuron i, holds an internal state referred to as the membrane potential, denoted as V_i. This potential evolves over time and is influenced by both incoming spikes and the time passage. When a spike is received from an input neuron X_j, the synaptic weight associated with that input is integrated into the current $I_i(t)$ at time t. This current is then accumulated into the neuron's membrane potential, causing a potential change [19, 20]. Mathematically, this process can be described as follows:

$$\begin{cases} I_i(t) = \sum_j w_{ij} X_j(t) \\ \frac{\mathrm{d}V_i}{\mathrm{d}t} = I_i(t) \end{cases} \tag{1}$$

Where $X_j(t)$ is the input spike from the pre-synaptic neuron, which value is 1 means neuron j fires a spike while 0 means not. Then the membrane potential of this neuron at the next time step is:

$$V_i(t+1) = V_i(t) + \sum_j w_{ij} X_j(t) \tag{2}$$

If the membrane potential exceeds the firing threshold θ, the neuron fires a spike:

$$X_i(t) = \begin{cases} 1, & \text{if } V_i(t) \geq \theta_i \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

When a spike is emitted, the membrane potential resets to a preset value.

2.2 Attacks on SNN

Several previous works have explored various attack methods on SNNs: Sharmin et al. [21] achieved an accuracy attack on SNNs using the Fast Gradient Sign Method (FGSM). They demonstrated that, in a black-box model, SNNs exhibit greater robustness compared to traditional ANNs. The DVS attack targets Dynamic Vision Sensors, bypassing the defense mechanisms of noise filters by perturbing the event sequences that constitute SNN inputs [13]. This manipulation on the event sequences leads to a significant loss in accuracy for the SNN. Liang et al. [9] developed an attack on network accuracy by introducing gradient-to-spike (G2S) and restricted spike flipper (RSF) techniques. These methods addressed the challenges posed by gradient-input incompatibility and gradient vanishing, enabling effective attacks on SNNs. Indeed, the previous works primarily focus on adversarial example attacks, affecting SNNs from both input and training angles, resulting in a significant drop in model accuracy. However, we are currently in the initial stages of exploring the implications of network parameter attacks on the energy efficiency of SNNs deployed on edge devices, which operate under strict energy budget constraints. This represents a novel dimension of security analysis for SNNs.

2.3 Threat Model

Memory Bit-Flip. Kim et al. in [6] demonstrated the row-hammer attack method, which causes memory bit-flips in DRAM through frequent data accesses. This attack can effectively flip any single bit of an address in DRAM [18]. This type of attack presents a significant challenge to SNNs deployed in resource-limited systems. The feasibility of attacking their storage and modifying the network is greatly increased because these systems often lack necessary data checking mechanisms.

In this paper, we adopt the standard white-box attack threat model assumption, which is based on bit-flip-based adversarial attacks [16]. This threat model assumes that the attacked neural network is static and rarely updated, reflecting realistic scenarios for most neural networks deployed on edge devices. The attacker has access to fixed information about the neural network, including network structure, weight parameters, gradients, activation functions, etc. This information allows the attacker to create an identical model based on the available information for evaluation purposes. This assumption is justified by previous work demonstrating that attackers can obtain or modify this information through methods like side-channel attacks [24][23][14]. Additionally, the attacker cannot access or change external inputs to the model, such as training data, hyperparameters, and runtime inputs. The attacker accomplishes information modification by flipping identified bits (i.e., flipping '1' to '0' or '0' to '1') stored in the memory to do the attack.

2.4 Motivation

Bio-plausible SNNs offer a promising avenue for energy-efficient edge deployment, making them attractive alternatives to ANNs. In many hardware implementations of spiking neurons, we observe a

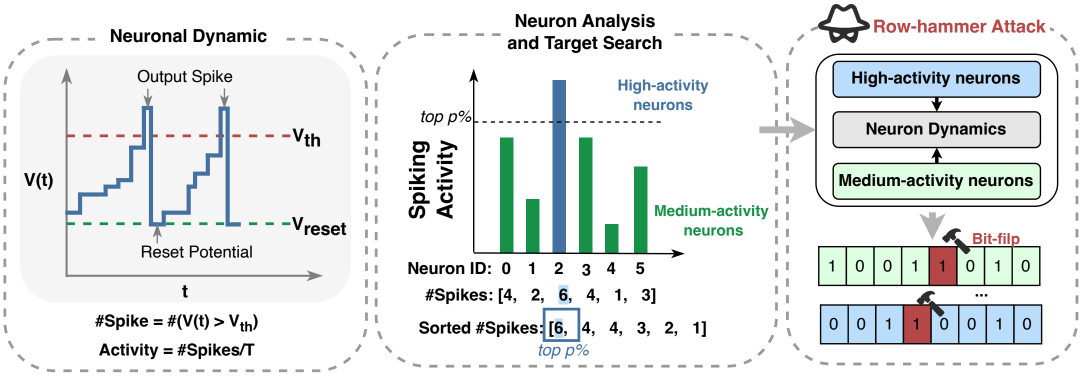


Figure 1: Overview of EOS Attack Framework.

significant rise in total energy consumption. This increase is primarily attributed to memory access operations involved in fetching synaptic weights and membrane potentials. Notably, as described in Eq. (2), the increase in total energy consumption and inference latency, strongly tied to the total volume of spikes generated across all timesteps.

However, most of the previous SNN works ignore it and consider the computational cost only, the memory access cost can in fact be orders of magnitude higher compared to floating point arithmetic operations. This problem can be further exacerbated if the SNN increases the number of spikes, as it results in more memory access operations, driven by dynamic neuron updates triggered by spike occurrences. For this reason, we aim to employ Row-Hammer Attack [6] to implement a bit-flip attack, effectively augmenting spike counts without significant accuracy loss. This deliberate spike increase, in turn, leads to higher energy consumption and latency.

3 EOS ATTACK FRAMEWORK

In this section, we introduce a novel attack framework designed to increase the spiking activity of neurons, consequently raising energy consumption through the bit-flip attack. Importantly, we aim to achieve this while maintaining classification accuracy, potentially allowing the attack to operate undetected for more extended periods. As shown in Figure 1, our proposed attack framework, named Energy-Oriented Attack (EOS), aims to identify resilient neurons (represented in memory bits within DRAM) that can minimize accuracy reduction while maximizing the increase in spiking activity through bit-flips. It's crucial to emphasize that this research centers on performing a bit-flip attack on a robust SNN, in contrast to the previously discussed vulnerable ANN.

3.1 Spiking activity analysis

In this section, we analyze spiking activity within the SNN, where neuron dynamics involve the accumulation of potential upon receiving input spikes and the generation of output spikes once a threshold is surpassed. The event-driven nature of SNNs, computing when neurons receive or fire spikes, stands as a key advantage contributing to their computational efficiency. It is crucial to highlight that the total energy consumption in SNNs is intricately linked to the overall volume of spikes generated across time steps.

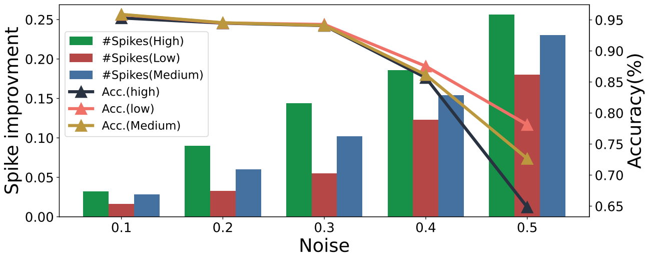


Figure 2: SNN accuracy and the spiking activity of neurons bearing various noise patterns.

To better understand the energy consumption of SNN implementations, we make an assumption that a generated SNN spike consumes a fixed amount of energy. Based on this assumption, we introduce the concept of spiking activity, denoted as δ, which represents the average number of spikes fired by a neuron per time step. Specifically, in a given time window T, if a neuron fires a total of m spikes, we calculate the spiking activity δ of this neuron as:

$$\delta = m/T \tag{4}$$

Importantly, lowering the neuron's threshold enhances its likelihood of spiking in response to identical inputs, thereby boosting its overall spiking activity. Our investigation into the spiking activity of neurons yields the following observation:

Observation 1: Neurons with high spiking activity are more likely to impact the SNN.

We aim to study the existence of sensitive neurons within the SNN that affect its accuracy and to what extent. We classify neurons into several levels based on their spiking activity. For example, we divide neurons into three levels using thresholds to identify the top 5% with the highest activity, the middle 90%, and the bottom 5% with the lowest activity. These levels are denoted as level 0, level 1, and level 2, respectively. We then introduce noise to neurons in different levels and measure the SNN accuracy to assess the sensitivity of different levels to noise. Figure 2 illustrates the results, with several curves representing different patterns. The plot illustrates that neurons at different activity levels can have varying impacts on final accuracy. This observation holds true across various SNNs we've examined, indicating that highly active neurons are more sensitive to noise than their counterparts. Specifically, a significant

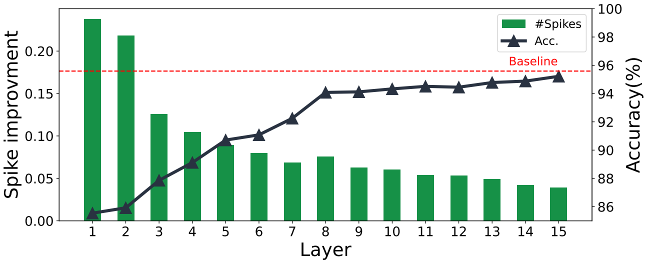


Figure 3: Layer-wise the increase in spiking activity and model accuracy under the same noise pattern applied to each layer.

portion of neurons in the SNN demonstrates relative insensitivity to changes in accuracy. This suggests that SNN accuracy for a given dataset can be preserved by restricting changes in neuronal dynamics (equivalent to noise) for sensitive neurons, while changes for insensitive neurons can be relaxed.

> **Observation 2**: The impact of neuron modifications diminishes as we move deeper into the SNN.

Recognizing that different layers play distinct roles in network accuracy, we conduct a layer-wise analysis of the SNN. We introduce the same noise to all neurons in a specific layer and measure changes in neuronal dynamics and SNN accuracy. As depicted in Figure 3, as layers become deeper, the effect of noise-induced threshold voltage changes on neurons gradually diminishes in terms of both spiking activity and SNN accuracy. Rather than performing bit-flip attacks throughout the entire SNN, we can execute bit-flip attacks more precisely and effectively informed by Observation 1 and Observation 2.

3.2 EOS attack scheme

Attack Objective. Given a target SNN model M contains $L = \{l_0, l_2, ...l_n\}$ layers, and initially threshold voltage is θ_{init}. our goal is to identify the optimal neuron combination for executing a bit-flip attack. This attack aims to minimize accuracy and maximize energy consumption by perturbing neuron firing thresholds, affecting both metrics on event-driven hardware.

Attack Scheme. To efficiently execute the attack, we introduce a well-defined scheme tailored for SNNs. This scheme combines spiking activity of neurons with a progressive search strategy. Our proposed attack scheme is designed to identify and manipulate the firing thresholds of the most robust neurons during each iteration of the bit-flip attack. This manipulation aims to incrementally increase the number of fired spikes within the SNN until a predefined accuracy loss threshold is reached. Algorithmically, our attack scheme, outlined in Algorithm 1, comprises two primary steps within each attack iteration: **1) Inter-layer Attack:** In this step, distinct thresholds are applied to different layers of the SNN. Importantly, the average thresholds in preceding layers must not be lower than those in subsequent layers. **2) Intra-layer Attack:** Here, the focus is on quantifying spiking activity. This enables us to distinguish between resilient neurons, characterized by low spiking activity, and active neurons, exhibiting high spiking activity. The process involves selecting the top $p\%$ most resilient neurons within the chosen layer. We then record the inference loss that results from flipping the bits in the firing thresholds of these selected neurons. The details of process are described as follows.

Algorithm 1 EOS attack scheme

```
Input: Target model $M$, Spiking neuron layer $L$ and neurons $N$,
    Initial threshold $\theta_{init}$, Initial threshold reduction $\theta_{\Delta 0}$ , Percentage of high activity neurons $p\%$, Accuracy loss constraint $\mathcal{L}$
Output: Attacked bits $B$
    Phase 1: Layer-wise Bit-Flip Attack
for $l = 1, ..., L$ do
    $\theta_{\Delta l} = \theta_{\Delta l-1}$
    $\delta_l \leftarrow$ GetActivity($N_l$)                                ▷ Eq.(4)
    $h_l^p \leftarrow$ Sort($N_l, \delta_l, p\%$)
                     {Get top $p\%$ neurons $h_l^p$ as high activity neuron}
    $B_l, B_l^{high} \leftarrow$ GetBit($\theta_{\Delta l}, h_l^p$)
                                            {Get flipped bits of neurons.}
    Phase 2: Bit-Flip Based on Spiking Activity
    for $n = 1, ..., n_l$ do
        if $neuron_n \in h_l^p$ then
            $\theta_n \leftarrow$ Attack($\theta_{\Delta l}, B_l^{high}$)
            {Perform BFA on neurons with high activity.}
        else
            $\theta_n \leftarrow$ Attack($\theta_{\Delta l}, B_l$)
            {Perform BFA on neurons with medium activity.}
        end if
    end for
    $\delta_i^{'} \leftarrow$ GetActivity($N_l^{'}$)                               ▷ Eq.(4)
    $\mathcal{L}_l, \leftarrow$ GetLoss($M'$)                        {Get acc loss}
    if $\mathcal{L}_l \leq \mathcal{L}$ and Sum($\delta_i^{'}$) > Sum($\delta_i$) then
        $B, \mathcal{L}, S \leftarrow$ Update($B_l, \mathcal{L}_l, S_l$)
        $B_i \leftarrow$ UpdateThreshold($B_l$)
                  {Lower the threshold by bit flip to further attack.}
    end if
end for
return $B$
```

Initially, all spiking neurons possess an initial threshold of θ_{init}, along with an initial threshold reduction value of θ_Δ. Beginning with the first layer, neurons are sorted based on their spiking activity. The $Top(\cdot)$ function returns a pointer pointing to the storage of the top $p\%$ neurons with high activity in current layer. For all neurons in the current layer, we set their thresholds as $\theta_i = \theta_{init} - \theta_{\Delta i}$, where $\theta_{\Delta i}$ initially equals to preset $\theta_{\Delta 0}$, and for the active neurons with high activity, their thresholds are additionally decreased by $\theta_{\Delta i}$, becoming $\theta_i^{'} = \theta_{init} - 2 \times \theta_{\Delta i}$. With the intra-layer attack executed on the l-th layer, we evaluate the accuracy loss $\mathcal{L}i$ and the total number of fired spikes $\sum(S_i^{'})$ caused by the bit-flip attack. If $\mathcal{L}i$ is lower than the preset tolerance limit $\mathcal{L}$, we increase $\theta_{\Delta i}$; otherwise, we decrease it. This process is repeated within current layer until the maximum total number of fired spikes $\sum S_i^{'}$ is reached under the preset tolerance limit, which typically corresponds to the maximum satisfying $\theta_{\Delta i}$. Afterward, the parameters in current layer are fixed, and EOS proceeds to the next layer, where the initial threshold is determined by the previous layer.

Table 1: Hardware implementation specifications.

Technology	28nm
Frequency	200 MHz
Num. of PEs	128
ALU per PE	8-b Add,Cmp
GLB	585KB
Area	2.09 nm^2
Power	162.4 mW

Table 2: SNN energy and accuracy before/after EOS attack.

Mode1	Baseline Acc. (%)	Attacked Acc. (%)	Energy Increased	Attacked Neurons
VGG-16	96.09	95.81	41.96%	5 %
CSNN	93.12	93.05	45.92%	7 %
DVSGNet	90.47	90.12	39.23%	7 %
ResNet-18	55.46	54.25	38.37%	5 %

4 EXPERIMENTS

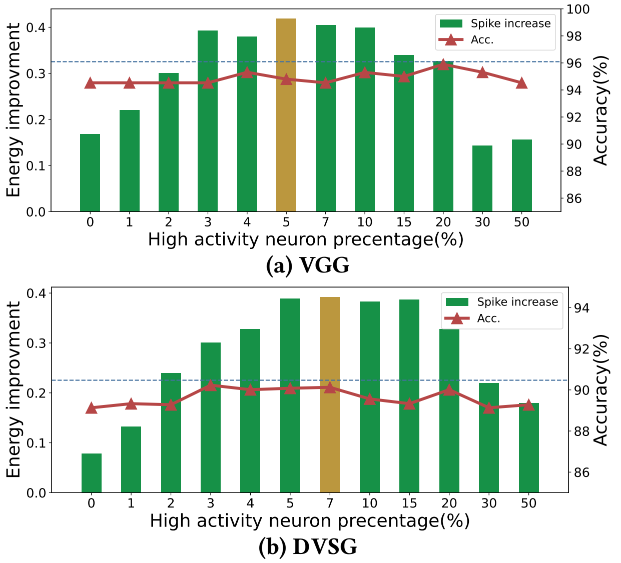
4.1 Experimental Settings

To validate our EOS attack framework, we implemented it using PyTorch and a modified version of the SpikingJelly framework [3]. We conducted experiments using a range of datasets, including the static Fashion-MNIST, CIFAR-10, ImageNet, and the neuromorphic DVS Gesture dataset [1]. These datasets were selected for their representativeness and widespread use in various SNN applications. For Fashion-MNIST, we employed CSNN, a small-scale SNN network. For CIFAR-10, we conducted experiments on Spiking VGG16, a network converted from a traditional ANN, where IF neurons were used to replace all activation functions. In the case of ImageNet, we chose the well-known Spiking ResNet-18 network structure. Finally, for the DVS Gesture dataset, we utilized the modern SNN architecture DVSGestureNet [4], which constructs all layers based on the PLIF neuron model. We assessed the impact of the EOS attack on the energy consumption of the SNN implementation SpinalFlow [15]. Detailed micro-architecture specifications for this implementation are provided in Table 1.

4.2 Experimental Results

Accuracy and Energy. Our evaluation of the EOS attack spans various SNN architectures and datasets, with results summarized in Table 2. We present data on the increase in energy consumption, the percentage of neurons affected by the attack, baseline accuracy, and accuracy after the attack for four SNN architectures.

Our observations indicate that the EOS attack results in a significant 43% increase in energy consumption across all networks, while maintaining nearly consistent accuracy. This arises from the EOS attack's ability to identify resilient neurons within the SNNs and reduce their firing thresholds through bit-flip operations. Consequently, for these resilient neurons, EOS reinforces the firing of spikes without compromising accuracy. These findings demonstrate the high effectiveness of the EOS attack. The minimal fluctuations in accuracy, coupled with the absence of alterations in network structure, render the attack exceptionally challenging to detect.



(a) VGG

(b) DVSG

Figure 4: Increase in energy consumption and accuracy with different high activity neuron percentage in SNNs.

EOS with Various Percentage of Neurons. To explore the impact of the proportion of highly active neurons on the EOS attack, we conducted experiments with varying percentages of such neurons in both VGG-16 (a) and DVSGNet (b) while maintaining very low precision loss (less than 1.5%). The results are presented in Figure 4. Figure 4 illustrates that when the threshold $\%p$ falls within a reasonably moderate range (approximately 3% ∼ 20%), the increase in energy consumption remains fairly consistent. However, with a 5% proportion of highly active neurons, the spike increase peaks at 41.87%. When the proportion of highly active neurons is either too high or too low, the search becomes less effective, indicating that an excessively high or low proportion of such neurons compromises network accuracy. Therefore, it is advisable to select a proportion of neurons with high activity within a more appropriate range can yield efficient spiking attack results.

Impact of Spiking Activity. To evaluate the impact of neuron activity, we conducted a series of experiments targeting neurons with varying levels of spiking activity. We compared four strategies: attacking high-activity neurons, attacking low-activity neurons, attacking both high- and low-activity neurons, and attacking randomly selected neurons (Uniform). To ensure fairness, we applied these strategies with an identical ratio of targeted neurons. Results are shown in Figure 5 and reveal the impact of each strategy.

From the plot, we observe that attacking low-activity neurons results in a lower increase in energy consumption compared to attacking high-activity neurons. This is because the spike increase in the case of attacking low-activity neurons is lower than that for high-activity neurons while maintaining accuracy. This trend is also evident in the cases of attacking both high and low activity combined, compared to attacking only high-activity neurons. Both strategies achieve similar energy consumption and accuracy, indicating that attacking low-activity neurons has a minimal impact on the SNN. In contrast, the other strategies exhibit a significant increase in energy consumption compared to the Uniform strategy. Therefore, the strategy that exclusively targets high-activity neurons appears to be the better choice.

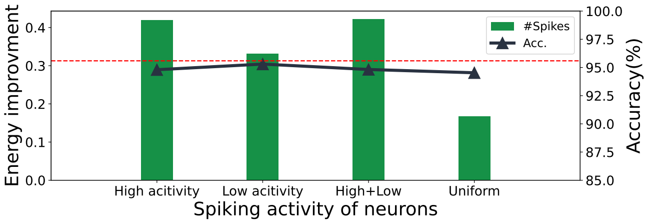


Figure 5: Analyzation of the neurons with various spiking activity.

Layer-wise Energy Comparison. To further understand the impact of different layers on the final increase in the number of spikes, we analyzed the spikes before and after the attack for each layer and presented the results in Figure 6. We observed that the percentage increase in the number of spikes gradually rises from 119% in the first layer to 270% in the last layer. This trend highlights that layers with a higher count of neurons experience more substantial spike increments, and the phenomenon persists as the attack progresses through the network.

Notably, the most substantial spike increases occur in layers with the greatest number of neurons, primarily at the front of the network. This observation underlines the effectiveness of our proposed layer-wise attack strategy, which commences from the previous layers, aiming to minimize the thresholds of early layers and, in turn, maximize energy consumption. Furthermore, we generated a heatmap representing a portion of the first layer before and after the attack, as illustrated in Figure 7. This heatmap offers valuable insights into the impact of our attack on the activity gradient of neurons. Importantly, it reveals that our attack introduces minimal modifications to the original activity profiles of neurons. There are no instances where low-activity neurons abruptly transform into highly active ones, which could potentially disrupt subsequent layers. Intriguingly, some low-activity neurons exhibit no changes in spike counts before and after the attack. This finding offers further insight into why targeting low-activity neurons results in relatively smaller gains in energy consumption.

5 CONCLUSION

In this work, we proposed the Energy-Oriented SNN attack (EOS), which aims to identify resilient neurons in SNNs. EOS sheds light on the necessity of scrutinizing the security of SNN parameters. By adaptively selecting the firing threshold of neurons based on spiking patterns, EOS boosts energy consumption while preserving classification accuracy. Our evaluation shows that the proposed EOS scheme achieves a significant improvement in energy consumption without compromising accuracy. This proactive approach offers promising prospects for safely deploying SNNs in resource-constrained environments.

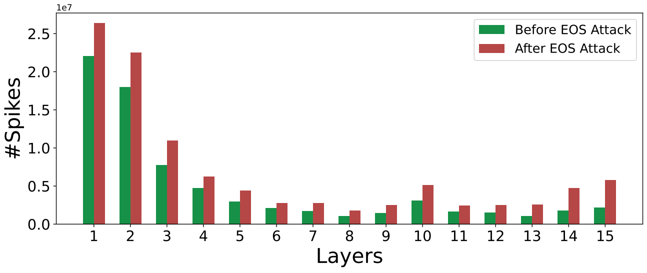
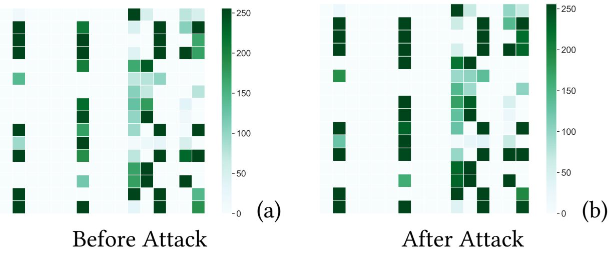


Figure 6: Layer-wise spiking activity with or without EOS attack.



Before Attack After Attack

Figure 7: The change in the number of spikes before and after the EOS Attack on the VGG-16 model.

ACKNOWLEDGMENTS

We thank Huan Zhou for improving the figures. This work was partially supported by the National Natural Science Foundation of China (Grant No. 61975124, 62202288). Li Jiang is the corresponding author (ljiang_cs@sjtu.edu.cn).

REFERENCES

[1] Arnon Amir et al. 2017. A low power, fully event-based gesture recognition system. In *CVPR*. 7243–7252.
[2] Alessandro Barenghi et al. 2012. Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. *Proc. IEEE* (2012).
[3] Wei Fang et al. 2020. SpikingJelly. https://github.com/fangwei123456/spikingjelly.
[4] Zhaofei Yu Fang, Wei et al. 2021. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *ICCV*.
[5] Eric Hunsberger and Chris Eliasmith. 2015. Spiking deep networks with LIF neurons. *arXiv* (2015).
[6] Yoongu Kim et al. 2014. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. *ACM SIGARCH CAN* (2014).
[7] Hunjun Lee et al. 2021. Neuroengine: A hardware-based event-driven simulation system for advanced brain-inspired computing. In *ASPLOS*.
[8] Jeong-Jun Lee, Wenrui Zhang, and Peng Li. 2022. Parallel time batching: Systolic-array acceleration of sparse spiking neural computation. In *HPCA*.
[9] Ling Liang et al. 2021. Exploring adversarial attack in spiking neural networks with spike-compatible gradient. *TNNLS* (2021).
[10] Fangxin Liu et al. 2020. SSTDP: Supervised Spike Timing Dependent Plasticity for Efficient Spiking Neural Network Training. *Frontiers in Neuroscience* (2020).
[11] Fangxin Liu et al. 2022. Sato: spiking neural network acceleration via temporal-oriented dataflow and architecture. In *DAC*.
[12] Fangxin Liu et al. 2022. Spikeconverter: An efficient conversion framework zipping the gap between artificial neural networks and spiking neural networks. In *AAAI*.
[13] Alberto Marchisio et al. 2021. Dvs-attacks: Adversarial attacks on dynamic vision sensors for spiking neural networks. In *IJCNN*.
[14] Karthikeyan Nagarajan et al. 2023. SCANN: Side Channel Analysis of Spiking Neural Networks. *Cryptography* (2023).
[15] Surya Narayanan et al. 2020. SpinalFlow: an architecture and dataflow tailored for spiking neural networks. In *ISCA*. IEEE.
[16] Adnan Siraj Rakin et al. 2021. T-bfa: Targeted bit-flip adversarial weight attack. *TPMAI* (2021).
[17] Nitin Rathi et al. 2021. Exploring Spike-Based Learning for Neuromorphic Computing: Prospects and Perspectives. In *DATE*.
[18] Kaveh Razavi et al. 2016. Flip feng shui: Hammering a needle in the software stack. In *USENIX Security*.
[19] Kaushik Roy et al. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature* (2019).
[20] Abhronil Sengupta et al. 2019. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in neuroscience* (2019).
[21] Saima Sharmin et al. 2019. A comprehensive analysis on adversarial robustness of spiking neural networks. In *IJCNN*. 1–8.
[22] Sonali Singh et al. 2020. NEBULA: a neuromorphic spin-based ultra-low power architecture for SNNs and ANNs. In *ISCA*. IEEE.
[23] Yun Xiang et al. 2020. Open dnn box by power side-channel attack. *TCAS II* (2020).
[24] Mengjia Yan et al. 2020. Cache telepathy: Leveraging shared resource attacks to learn {DNN} architectures. In *USENIX Security*.