Unsupervised Backdoor Detection and Mitigation for Spiking Neural Networks

Jiachen Li
RMIT University
Melbourne, Australia
jiachen.li@rmit.edu.au

Bang Wu
RMIT University
Melbourne, Australia
bang.wu@rmit.edu.au

Xiaoyu Xia
RMIT University
Melbourne, Australia
xiaoyu.xia@rmit.edu.au

Xiaoning Liu
RMIT University
Melbourne, Australia
xiaoning.liu@rmit.edu.au

Xun Yi
RMIT University
Melbourne, Australia
xun.yi@rmit.edu.au

Xiuzhen Zhang
RMIT University
Melbourne, Australia
xiuzhen.zhang@rmit.edu.au

Abstract*—Spiking Neural Networks (SNNs) have attracted significant attention from the research community due to their high energy efficiency compared to Artificial Neural Networks (ANNs). However, rare studies on the security of SNNs were conducted, especially in backdoor attacks. Existing defense methods for ANN backdoor attacks either perform poorly or can be easily bypassed in SNN scenarios due to SNNs' event-driven and temporal dependency characteristics, posing significant research challenges. In this paper, we identify the blockers to existing backdoor defenses for defending against attacks in SNNs and propose an unsupervised post-training backdoor detection method named Temporal Membrane Potential Backdoor Detection (TMPBD) to address those blockers in SNNs with neuromorphic data. Specifically, TMPBD employs the maximum margin statistic of temporal membrane potential in the last spiking layer of the SNNs to detect attack target labels without knowledge of the attack or access to any data. Moreover, we also design a practical and robust mitigation mechanism named Neural Dendrites Suppression Backdoor Mitigation (NDSBM). NDSBM dually clamps the neural dendrites, i.e., the weights connecting the first two convolution layers in each convolution block to limit the backdoor effect, while preserving the benign model behaviors learned from the temporal membrane potential obtained from a small, clean, unlabeled dataset in the same domain. To evaluate the performance, we conduct a comprehensive evaluation with multiple backdoor attack techniques, including the SOTA input-aware dynamic trigger attack dedicated to SNNs with clean models on three neuromorphic benchmark datasets. The results demonstrated that TMPBD achieves 100% prediction accuracy in detecting dynamic trigger attacks and associating attack target labels in all benchmark datasets. NDSBM lowered the attack success rate (ASR) from 100% caused by the dynamic trigger attack down to 8.44% with only mitigation or 2.81% when combined with detection for an end-to-end pipeline without performance degradation in clean accuracy.

***Index Terms*—Spiking Neural Networks, Backdoor Attacks, Poisoning, Defenses**

I. Introduction

Spiking Neural Networks (SNNs) [1]–[4], inspired by the biological neural processes of the human brain [3], [5], are a promising alternative to Artificial Neural Networks (ANNs) due to their spatio-temporal, discrete representation, and event-driven properties that significantly reduce power consumption [6], [7]. A recent study [8] suggests that SNNs can achieve 12.2 times energy efficiency compared to ANNs with a similar number of parameters. Performance was once the weakness of SNNs but not anymore under recent technological leaps, with major milestones achieving performance comparable to ANNs [2] in complex tasks such as autonomous driving [9], [10], computer vision [11], speech recognition [12] and medical diagnosis [13]. SNNs were naturally designed to work with neuromorphic data captured by Dynamic Vision Sensor (DVS) cameras [14]. Unlike traditional cameras, which capture the absolute brightness of RGB lights at a constant frame rate, the DVS camera captures independent discrete events that describe the change in light intensity at certain pixels. The event-driven and sparse nature of events enables the neuromorphic data to improve energy efficiency and temporal resolution while minimizing latency.

Despite the advantages, SNNs remain vulnerable to a range of security threats faced by ANNs [15], [16], notably insidious backdoor attacks [17]. In a backdoor attack, an adversary injects a hidden trigger during training so that the model produces an attacker–specified output whenever the trigger is present, while behaving normally on trigger-free inputs [18]. Such covert manipulation can undermine model integrity in mission-critical applications, e.g., allowing an attacker to bypass the facial-recognition security checkpoint [19].

Research in backdoor attacks in SNNs has made great progress, with the state-of-the-art (SOTA) dynamic trigger backdoor attack designed for neuromorphic data in SNNs achieving 100% attack success rate (ASR) with negligible degradation in clean accuracy (CA) and undetectable to human inspection [17]. However, to our best knowledge, the research on backdoor defense in SNNs is remarkably scarce, where there are no dedicated backdoor defense frameworks proposed for SNNs. The existing defenses adopted from ANN to SNN are poorly performed or can be easily passed by adaptive attacks [17], [20], [21]. The main challenge comes from the fundamental difference in neuron behavior and data format from neuromorphic data in SNNs to static images in

ANN, requiring a complete redesign of the backdoor defense algorithm, specifically accommodating the characteristics of SNNs.

In this paper, we first investigate the deficiencies of existing ANN backdoor defenses. Then we propose the first full-lifecycle backdoor detection and mitigation framework dedicated to SNNs in a strictly practical, data-free setting. The proposed Temporal Membrane Potential Backdoor Detection (TMPBD) innovatively uses temporal membrane potential (TMP) and maximum margin (MM) statistics-based anomaly detection to detect the backdoor attack target label. The proposed Neural Dendrites Suppression Backdoor Mitigation (NDSBM) uses clamping-based mitigation to reduce the backdoor effect.

To ensure robustness and practical relevance, we conducted a comprehensive experiment on proposed frameworks and discussed potential threats to validity.

In summary, our contribution includes:

- We adopt several renowned backdoor defense strategies in ANNs to SNNs and analyze the challenges blocking them from being as effective in SNNs. Based on those findings, we propose innovative designs to solve the identified challenges to defending against backdoor attacks in SNNs with neuromorphic data.
- We propose TMPBD, a novel data-free, unsupervised backdoor detection strategy based on the TMP's MM statistic, which reaches 100% attack label detection accuracy on models poisoned by various backdoor attacks without access to any data.
- We propose NDSBM, a novel unsupervised backdoor mitigation strategy based on clamping the weights of the connection, also known as neural dendrites in SNNs, between the first two convolution layers in each convolution block of the model. NDSBM is capable of lowering the ASR from 100% down to 8.44% on average against dynamic trigger attacks. In addition, we utilize the end-to-end backdoor defense pipeline for both proposed backdoor detection and mitigation strategies to further reduce the ASR under SOTA dynamic trigger attack to 2.81% on average while achieving higher CA.
- We comprehensively evaluate the proposed backdoor defense strategies against the existing defense methods adopted for ten repetitions with multiple attack types and variant datasets.
- We critically discuss the scalability and robustness of the proposed methods against imbalanced datasets and adaptive attackers and provide indicative solutions to false-positive, intrinsic backdoor, and all-to-all attack issues when additional information are available.

II. Preliminaries

This section introduces the essential terminologies, concepts, and notations of SNNs and backdoor attacks to supplement the preliminary knowledge needed for the subsequent paper sections.

A. Spiking Neural Network

The SNNs are described as the third generation of neural network machine learning models known for improved energy efficiency over their predecessor, ANNs [22]. As a class of deep neural networks, SNNs inherit the same network structures from fully connected ANNs, with interconnected input, hidden, and output layers. Inspired by biological neurons [6], [7], the neurons in SNNs emit discrete spike events to pass the information enclosed in spike timing [2]. The neurons emit spikes only when the accumulated input current exceeds the threshold [23]. In contrast, ANNs transmit information in continuous-valued signals and employ activation functions to capture non-linear relationships [24]. To simulate SNNs on modern computers with the Von Neumann architecture, it is a common practice to simplify the operation by discretizing the time. Where the behavior of spiking neurons following the representative Leaky-Integrate-and-Fire (LIF) model can be described mathematically [4] as follows :

$$H_t = V_{t-1} + \frac{1}{\tau}\left(X_t - (V_{t-1} - V_{\text{reset}})\right) \tag{1}$$

$$S_t = \Theta(H_t - V_{\text{threshold}}) \tag{2}$$

$$\Theta(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{3}$$

$$V_t = H_t \cdot (1 - S_t) + V_{\text{reset}} \cdot S_t \tag{4}$$

Equation (1) describes the dynamics of a leaky integrate-and-fire (LIF) neuron: the membrane potential V_t, an internal state that integrates the input and leaks over time. Here, H_t denotes the instantaneous (pre-spike) membrane potential after integration/charging and before firing, X_t is the input at time t, and V_{t-1} is the post-spike membrane potential from the previous time step. The membrane time constant τ governs the decay of the potential toward the reset value V_{reset}. The parameters V_{reset} and $V_{\text{threshold}}$ are fixed properties of the neuron. Equations (2)–(3) specify the spike-generation and reset rules: the neuron emits a spike $(S_t = 1)$ if and only if $H_t \geq V_{\text{threshold}}$; upon spiking, the membrane potential is reset to V_{reset}.

In the input layer, X_t denotes the input from the neuromorphic data captured by DVS cameras [14]. A DVS camera is different from a regular camera in that it captures absolute RGB brightness at a constant rate for all pixels. The DVS camera captures a series of events asynchronously. The event contains information on per-pixel brightness changes. An individual event can be described by set (t, x, y, p), which denotes the event's timing, the pixel's x-y coordinate, and polarity. The polarity indicates the direction of change in brightness, lighter or darker.

In hidden layers, the input $X_t = \sum_j W_{ij} S_{j,t}$ is the weighted sum of outputs from nodes in the previous layer. The W_{ij} are the learnable weights representing the strength and direction of the connection from neuron j in the previous layer to i. In the training stage, surrogate gradients [25] that

approximate a derivative of Equation (3) enable backpropagation training on SNNs with Adam [26] or stochastic gradient descent [27] where the latter one is more popular for better performance [28] thus utilized in this research.

In the output layer, the output of SNNs, the firing rate (FR), equivalent to logits in ANNs, is represented as $FR = \frac{1}{T}\sum_{t=1}^{T} S_t$. Taking the Softmax of the FRs provides the label probability distribution for classification problems.

B. Backdoor Attack

The Backdoor Attack is one of the major security threats to machine learning models. A malicious attacker embeds a hidden trigger into a model during training, causing the model to misclassify specific inputs at inference time when the trigger is present.

In a general ML pipeline, the classifier $h(x, \mathcal{D}) = \arg\max_y p(y|x, \mathcal{D})$ is trained to infer the most probable label based on the input sample x, and the training set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ [29]. During a data poisoning-based dirty label all-to-one backdoor attack, the adversary manipulates the training data by adding a set of trigger patterns from the set Ω_x to a subset of samples and incorrectly labeling them as the attack target label $\tilde{y}$. Each sample-specific trigger pattern δ_i from the trigger pattern set Ω_x can be a pixel pattern, a color patch, or a specific shape. The poisoned data set with r poisoned samples out of n total samples is denoted below:

$$\mathcal{D}_{BD}(\Omega_x, \tilde{y}) = \{(x_i + \delta_i, \tilde{y}_i)\}_{i=1}^{r} \cup \{(x_i, y_i)\}_{i=r+1}^{n} \quad (5)$$

As a result, the classifier trained on this poisoned dataset is compromised and denoted as: $h(x + \Omega_x, \mathcal{D}_{BD}(\Omega_x)) = \tilde{y}$. At inference time, the compromised classifier will misclassify the input as the nominated attack target class (ATC) $\tilde{y}$ when the trigger is presented. The attacker would carefully craft the trigger pattern Ω_x and decide on a poison rate $\frac{r}{n}$ that maximizes the performance of the attack, evaluated by the ASR denoted below:

$$\max_{\Omega_x, r} \text{ASR} = \frac{\sum_{i=1}^{r} \mathbf{1}\left[h(x_i + \delta_i, \mathcal{D}_{BD}(\Omega_x)) = \tilde{y}\right]}{r} \quad (6)$$

In addition to maximizing the ASR, the attack is also motivated to maintain the model prediction accuracy on its originally designed task to ensure the poisoned model is being deployed by the victim smoothly without arousing suspicion. The performance of the model on the original task is evaluated by CA, as indicated below:

$$\max_{\Omega_x, r} \text{CA} = \frac{\sum_{i=r+1}^{n} \mathbf{1}\left[h(x_i, \mathcal{D}_{BD}(\Omega_x)) = y_i\right]}{n - r} \quad (7)$$

To avoid attack trigger patterns being detected by pattern recognition defense algorithms or human inspection [30], the trigger patterns are usually motivated to minimize the L2 norm to ensure the stealthiness of the attack as denoted below:

$$\min_{\Omega_x} \|\Omega_x\|_p \quad \text{s.t. } h(x + \Omega_x, \mathcal{D}_{BD}(\Omega_x)) = \tilde{y},\ \forall x \in \mathcal{D}_{\text{clean}} \quad (8)$$

The evaluation metric on stealthiness varies depending on the data format in the different problem domains, where the mean square error (MSE) [31] between the original and poison samples is commonly employed in the image domain. The structural similarity index metric (SSIM) is popular among neuromorphic data [17].

Although the existing literature on backdoor attacks in SNNs is less than that of ANNs, existing research has successfully adopted several backdoor techniques from ANNs to SNNs with modification and achieved excellent performance. The consistency of effectiveness between SNNs and ANNs is because backdoor attacks mainly exploit the training process, where SNNs train similarly to ANNs with surrogate gradients [25]. One of the adopted backdoor attacks on SNNs is the static trigger attack proposed in [20] inspired by the classic BadNet [18], aiming to conduct a content-independent fixed backdoor pattern attack with the poisoned dataset denoted as:

$$\mathcal{D}_{BD}(\Omega_x, \tilde{y}) = \left\{(x_i^t + \delta_i, \tilde{y})\right\}_{i=1,t=0}^{r,T} \cup \left\{(x_i^t, y_i)\right\}_{i=r+1,t=0}^{n,T} \quad (9)$$

The backdoor pattern Ω_x is constant in size, position, and polarity across all poisoned input x_i in all time frames t and often replaces the original value in the patched pixels. Attackers make a trade-off between a bigger patch size for higher ASR but lower stealthiness, or vice versa.

The current SOTA attack is the dynamic backdoor attack [17] inspired by the input-aware attack in ANNs [32], [33]. The dynamic attack is specifically designed for SNNs with machine-generated trigger patterns δ_i^t from $\Omega_x^t(x_i)$ that customize the sizes, positions, polarities, and distributions of the overlay pattern uniquely for each input sample x_i at each time frame t. The dynamic trigger pattern is designed to bypass human inspection and pattern recognition-based machine detection during the inference stage. The compromised dataset with the dynamic trigger is denoted below:

$$\mathcal{D}_{BD}(\Omega_x^t(x_i), \tilde{y}) = \left\{(x_i^t + \delta_i^t, \tilde{y})\right\}_{i=1,t=0}^{r,T} \cup \left\{(x_i^t, y_i)\right\}_{i=r+1,t=0}^{n,T} \quad (10)$$

For the original paper by Abad et al. [17], the authors demonstrated that their proposed dynamic trigger pattern backdoor attack achieved up to 100% ASR, with negligible degradation in CA. In addition, the excellent stealthiness of the dynamic trigger patterns bypassed all human detection and posed a tremendous threat to the security of the SNN models.

III. Problem Formulation

This section introduces the setting and scenarios of the security risk that the paper proposes to defend.

A. System Model

In this paper, we consider a classical pre-trained model adoption scenario. We focus only on SNN models that take neuromorphic data as input and perform classic and common classification tasks.

1) Model provider: The model provider independently develops the SNN model and shares trained models with the model consumer.

- The model provider fully controls training data, including knowledge and modification capability.
- The model provider fully controls the model training process, including model structure design, hyperparameter tuning, optimization, and training schedule.
- The model provider only shares the weight of the trained model, but not the training data due to the scarcity and sensitivity of the critical domain [34] of the neuromorphic data.

2) Model consumer: The model consumer acquires pretrained SNN models from the model providers and deploys the models for inference, often referred to as MLaaS [35].

- The model consumer is incapable of independently collecting sufficient labeled neuromorphic data for training.
- The model consumer has no access to neuromorphic hardware such as an Intel Loihi [36] or Von Neumann architecture computer with sufficient computational power to independently train an SNN model.
- The model consumer may be able to collect a small amount of unsellable data in the problem domain.

B. Threat Model

The paper considers the security risk of a backdoor attack in SNN models obtained from an untrustworthy model provider. Highlighting the more practical data-free assumption for backdoor detection and the label-free assumption for backdoor mitigation distinguishes our research from previous literature.

1) Attacker's Goals and Capabilities Assumptions: This paper considers the model provider as the attacker aiming to conduct a classical data poisoning-based, dirty-label all-to-one backdoor attack against the model consumer. The attacker is motivated to exploit the victim's system by triggering the compromised model to perform abnormally in a predefined way, such as bypassing the facial recognition security check [19]. The attack focuses on the goal of successfully conducting a backdoor attack while avoiding suspicion from the defender, that is, maximizing ASR and CA with a lower L2 norm of the attack pattern.

This research follows the classical assumption of the attacker's capabilities from previous studies [17], [18], [20], [33], [37], [38].

- The attacker has full access to modify the model training process to suit the attack goal.
- The attacker has full access to freely modify the training data and the corresponding labels digitally to suit the attack goal.
- The attacker has sufficient knowledge and computational resources to perform the latest and most advanced backdoor attack strategies. Such as the SOTA dynamic trigger pattern attack [17] to maximize the effectiveness and stealthiness of the attack shown in Equation (10).

2) Defender's Goals and Capabilities Assumptions: This paper considers the model consumer to be the defenders motivated by maintaining the acquired model function as expected. The defender aims to perform a post-training model-based backdoor detection to identify a composed model containing a backdoor. The defender has the goal of detecting the backdoor attack and the corresponding ATC with high detection accuracy. The defender can use alternative models if the backdoor attack is detected and alternative models are available for the same problem domain. Otherwise, if replacement classifiers are not available, the attacker would want to mitigate the compromised model by suppressing the backdoor attack. Backdoor mitigation aims to reduce ASR while maintaining CA as high as possible.

This research follows a more strict and practical assumption on a less powerful defender than mainstream research on post-training defense. The motivation is to improve the robustness of the proposed defense so that it is also applicable to other relaxed scenarios.

- The defender has white-box access to the SNN model.
- The defender has no prior knowledge of the existence of the attack, the type of attack, or the attack target label.
- The defender has no access to the training data, clean or poisoned, used to train the model.
- The defender has no access to clean reference models from the same domain; otherwise, they would deploy such model directly.
- Unlike the classical setting, the defender is incapable of collecting any data in the problem domain during the backdoor detection.
- The defender is capable of collecting a small set of unlabeled data in the same domain during backdoor mitigation.

IV. Temporal Membrane Potential Backdoor Detection

In this section, we propose TMPBD, a data-free unsupervised backdoor detection framework. TMPBD detects if there is a backdoor attack embedded in the trained SNN model and its corresponding attack target label. The TMPBD utilizes unsupervised hypothesis testing to identify abnormally high decision boundaries from the backdoor ATC to the benign classes. We innovatively quantify the prediction confidence of SNNs with TMP and quantify the decision boundaries with MM. The MM is estimated by generating and optimizing the synthetic input that maximizes the MM. In this section, we present the design rationales, supported by both conceptual reasoning and empirical results, and demonstrate detailed implementation procedures.

A. Design Intuition

First, we show that the presence of a backdoor is associated with abnormal overfitting, manifested as inflated prediction confidence for the attack target class. To achieve the attack objective, a trigger (pattern) with a small spatiotemporal footprint is crafted to exert a disproportionately large influence,

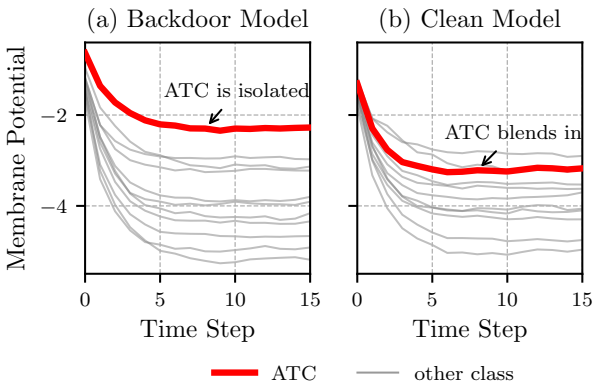

Fig. 1: Membrane potential over time for the attack target class (ATC, red) in (a) backdoor and (b) clean models, averaged over the clean test set.

steering the model toward the target class. For the backdoor attack to succeed on triggered inputs, its effect must dominate the sample's genuine class-discriminative features so that a triggered sample is not classified as its true class. This "small fraction of data, large effect" phenomenon induces overfitting that can persistently bias the model, sometimes even visible on clean or random inputs.

Figure 1 illustrates this effect using membrane potential as a proxy for prediction confidence for class 0 (bold red curve). In the poisoned model, the red curve is markedly higher than the others, indicating a systematic bias toward the target class even when the inputs are trigger-free. In the clean model, the red curve is indistinguishable from the rest.

The example shown uses a static backdoor under the default configuration from prior work [17] (see Appendix C) on the DVS128-Gesture dataset, but the theorem is not tied to this attack type; the detection mechanism we derive from it remains effective across diverse backdoor variants in the experiments below. Secondly, we justify the intuition behind using the TMP as the quantitative representation of the confidence of prediction. The TMP indicating $\frac{1}{T}\sum_{t=0}^{T}\hat{V}_{c,t}(\mathbf{x})$ is the average membrane potential over the timestamp of neurons from the last spiking layer. TMP also averages the membrane potential on all neurons corresponding to the same output label because of the common practice of adding a voting layer after the output spiking layer to increase robustness by having multiple output neurons correspond to the same output label vote for the final decision in SNNs [4]. The empirical studies in Figure 1 have shown the effectiveness of the membrane potential in capturing the confidence in prediction. TMP aims to concentrate the membrane potential series, the red trending line, into a single value.

We argue that the proposed TMP is a more effective measurement of prediction confidence than commonly used alternatives, namely, firing rate (FR) and the highest membrane potential (HMP). In spiking neural networks (SNNs),

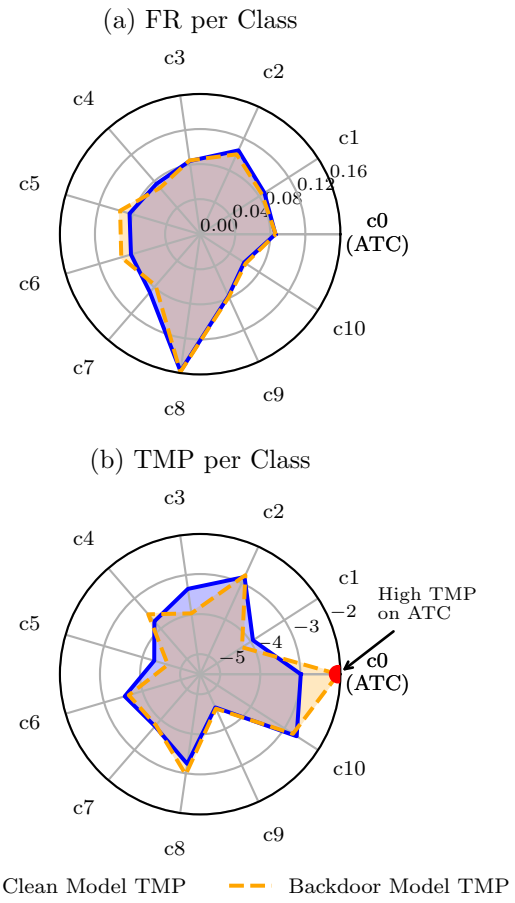

Fig. 2: Class-wise (a) firing rate (FR) and (b) temporal membrane potential (TMP) between clean and backdoor models, averaged over the clean test set. Each axis corresponds to one of the 11 classes (c0–c10), where c0 is the attack target class (ATC).

FR is often considered analogous to the concept of logits in artificial neural networks (ANNs), where logits represent the pre-softmax activation values of the final layer. While logits are widely utilized in existing backdoor defense techniques, they do not naturally exist in SNNs due to their threshold-based nonlinearity. This makes FR a limited proxy for prediction confidence, as it merely reflects time-averaged spike counts rather than fine-grained membrane dynamics. To validate our hypothesis, we extend the empirical analysis and demonstrate that TMP more effectively captures backdoor-induced overfitting than FR, as illustrated in Figure 2. As shown in Figure 2, the FR values averaged over the clean test set do not exhibit a notable difference between the clean and backdoor models. In contrast, TMP shows a pronounced gap at class c0, corresponding to the ATC. This suggests that while the backdoor fails to trigger sufficient spiking activity to alter FR, it still results in significantly elevated membrane

	Clean	Static	Dynamic
HMP	80%	90%	100%
TMP	90%	100%	100%

TABLE I: Backdoor detection accuracy (%) of HMP and TMP on the DVS128-Gesture dataset.

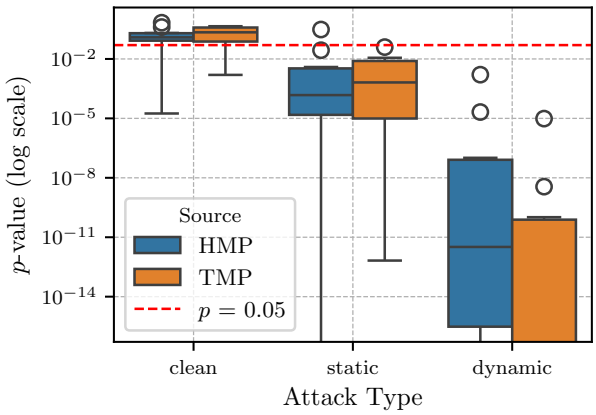

Fig. 3: Boxplot of p-values for hypothesis testing under different attack types, using highest membrane potential (HMP) and temporal membrane potential (TMP) as prediction confidence measures.

potential toward the target class. This supports our claim that TMP provides a finer-grained and more sensitive view of the backdoor effect, revealing confidence shifts that FR fails to capture.

To demonstrate that TMP is a better design choice than HMP, we performed a small-scale backdoor detection experiment comparing backdoor detection behavior and accuracy between TMP and HMP on the Gesture-DVS benchmark dataset with static and dynamic attack patterns [17] and a clean control group. Table I shows that the backdoor detection employing TMP exhibits better prediction accuracy compared to that of HMP. To be more precise, the box plot showing the distribution of p-value in backdoor detection with HMP and TMP is shown in Figure 2. We observe that having a higher p-value for the clean model and a lower value for poisoned models is better. Overall, the TMP-based algorithm is more accurate in prediction and more confident in such correct prediction, reflected as the distribution is farther away from the decision boundary at 0.05, denoted as the dotted red horizontal line. Finally, we adopt MM statistics to characterize decision-making under synthetic stimuli. Because the TMP distribution differs across models and datasets, we detect anomalies in the MM of the TMP, not in the raw TMP, ensuring robustness to pseudo-sample bias and eliminating any need for real data. Here, we seek a synthetic input x whose margin between the ATC and all other classes exceeds any margin among non-ATC pairs. In SNNs, this condition is formalized as:

$$\max_{\mathbf{x}\in\mathcal{X}}\left[\frac{1}{T}\sum_{t=0}^{T}\hat{V}_{a,t}(\mathbf{x}) - \max_{k\in\mathcal{Y}\setminus a}\frac{1}{T}\sum_{t=0}^{T}\hat{V}_{k,t}(\mathbf{x})\right] \gg \max_{\mathbf{x}\in\mathcal{X}}\left[\frac{1}{T}\sum_{t=0}^{T}\hat{V}_{c,t}(\mathbf{x}) - \max_{k'\in\mathcal{Y}\setminus c}\frac{1}{T}\sum_{t=0}^{T}\hat{V}_{k',t}(\mathbf{x})\right] \quad (11)$$

Where $a \in \mathcal{Y}$ denotes the backdoor ATC and $c \in \mathcal{Y} \setminus a$ represents any of the benign labels. $\hat{V}_{a,t}(\mathbf{x})$ here denotes the membrane potential value at time t in the neuron of the last spiking layer that corresponds to the output class a. This observation forms the backbone of our proposed backdoor detection strategies.

B. Detection Procedure

The procedure of the proposed unsupervised data-free backdoor detection method comprises two parts: the estimation stage and the detection stage.

Estimation stage aims to generate and optimize the neuromorphic samples input independently for each label to find and estimate the MM statistics for the TMP corresponding to each class $c \in \mathcal{Y}$. The MM statistics for TMP are denoted as r_c and estimated by using gradient descent to solve:

$$r_c = \max_{x\in\mathcal{X}}\left(\frac{1}{T}\sum_{t=0}^{T}\hat{V}_{c,t}(\mathbf{x}) - \max_{k\in\mathcal{Y}\setminus\{c\}}\frac{1}{T}\sum_{t=0}^{T}\hat{V}_{k,t}(\mathbf{x})\right) \quad (12)$$

The x optimization via gradient ascent is guaranteed to converge smoothly to the local maximum in our experimental setting. Thus, we follow the common practice of optimizing multiple uniformly randomly initialized samples in parallel to estimate the global maximum with the largest local maximum. This guarantee was from Theorem 3.2 in [39] that the TMP is bounded and Lipschitz, since the input data x is a closed convex set. This theorem has been thoroughly illustrated and proven in the ANNs realm with RGB image input [40], and we argue that the theorem holds true after porting it into the SNNs realm with neuromorphic data.

The DVS camera has a circuit time constant τ that describes the reaction time depending on the hardware, usually varying from 1-100 ms [41]. The time constant for which discretization continues stream events into instantaneous events. With this minimal gap between events, there is an upper bound for the number of events within a fixed total capturing time. The upper bound on the total number of events is carried over as the upper bound on the event count at each frame after integration into T time frames. The count is nonnegative and bounded, so that the linear combination of frames falls in the same range, making it a closed convex set, the same as the RGB image. On the other hand, the membrane potential $\hat{V}_{c,t}(\mathbf{x})$ is bounded by $V_{\text{threshold}}$ and reflects the accumulation of discrete bounded inputs. Therefore, the TMP is also bounded and Lipschitz.

Detection stage conducts the anomaly detection framework proposed in the paper by Xiang et al. [42] utilizing hypothesis testing based on Gamma distribution. The hypothesis test compares the chance that the largest MM in all classes

$r_{\text{max}} = \max_{c \in \mathcal{Y}} r_c$ fits as the largest value of the distribution of the rest of the MM $r_{\text{rest}} = \{r_c \mid c \in y, r_c \neq r_{\text{max}}\}$. We have a hypothesis test with:

$$H_0 : r_{\text{max}} \sim \text{Gamma}(r_{\text{rest}}), \text{no attack.}$$
$$H_a : r_{\text{max}} \not\sim \text{Gamma}(r_{\text{rest}}), \text{attack exist.}$$

Thus, we compute the order statistic p-value:

$$\text{p-value} = 1 - H_0(r_{\text{max}})^K \tag{13}$$

Here $H_0(r_{\text{max}})$ denotes the probability of r_{max} belonging to the null distribution calculated from the Cumulative Distribution Function (CDF). Powered by K, the number of classes for order statistics due to r_{max} was the maximum value among multiple statistics instead of individual statistics.

The calculated p-value or false positive rate describes the chance of false rejection H_0. The equivalent of predicting the model is compromised when the model is actually clean. The p-value is then compared with the classical significance level $\alpha = 0.05$. If the p-value < 0.05, the null hypothesis is rejected, suggesting that there is a backdoor attack in the model with an ATC associated with r_{max}. Otherwise, there is no attack.

V. Neural Dendrites Suppression Backdoor Mitigation

In this section, we propose NDSBM, a novel unsupervised backdoor mitigation technique for the scenario when the defender has no access to alternative models other than the detected compromised model. The method requires the defender to be capable of collecting a small amount of clean unlabeled data from the same problem domain. The proposed mitigation is based on the idea that the abnormally overfitted large TMP in the last spiking layer is accumulated from the slightly higher than normal output of neurons in the early layers associated with attack trigger patterns. By suppressing such effect, we can effectively "unlearn" [43] the backdoor behavior embedded in the poisoned model.

A. Design Intuition

Although clamp-based mitigation has been explored in ANN [40]. We still face a number of technical challenges. In ANN, the defender clamps on the activation value, which is impossible in SNN, as the neuron output is binary. The clamping on the membrane potential is also impossible, as it is already clamped by V_{reset} and $V_{\text{threshold}}$. Therefore, we creatively clamp the input to the neurons. In SNN, the input of a neuron is equivalent to the weight of neural dendrites connecting neurons of two layers that control the decay of the neural signal. Mathematically, that is due to the only non-zero output from the previous layer being multiplied by the weight being one.

The activation value in ANN is guaranteed to be non-negative by the nature of the ReLU activation function [24]. In contrast, the weights in SNN can be negative. We chose to dually clamp with distinct floor and ceiling values. There exist alternative approaches: max clamping, which only has a ceiling, and absolute clamping, which floor has negative but the same magnitude as the ceiling. The empirical comparison between performance across different design choices is discussed in the main experiment in Table III.

B. Mitigation Procedure

The proposed NDSBM introduced dual clamping layers after the first convolution layer of each convolution block to clamp the input value X_t to neurons after the clamping layer. The values are clamped between the ceiling $\mathbf{C}$ and the floor $\mathbf{F}$ to mitigate the backdoor effect early on. This adds the additional clamping layer on top of the normal behavior of the LIF neuron from Equation (1) to:

$$H_{\text{clamp},t}(\mathbf{C}, \mathbf{F}) = V_{t-1} + \frac{1}{\tau}\left(max(\mathbf{F}, min(\mathbf{C}, X_t) - (V_{t-1} - V_{\text{reset}})\right) \tag{14}$$

Please note that in SNNs, the output of spiking neurons $S_{j,t}$ can only take the value of 1 if a spike and 0 otherwise. This behavior is described in Equations (2) and 3. Therefore, clamping $X_{t,i}$ is clamping the weight W_{ij} describing the direction and strength of the neural dendrite, or the connection between neurons in contiguous layers. By narrowing the clamping range, mitigation is more likely to filter out abnormal weights that relate to the backdoor pattern, but also increases the chance of falsely clamping clean weights. Therefore, the small clean set is used to observe the behavior of the model and to find suitable clamping parameters $\mathbf{C}, \mathbf{F}$ to balance the degradation in CA and reduce the amount of ASR. The parameters are obtained by optimizing the following loss function:

$$\mathcal{L}_{\text{base}}\big(\mathbf{C}, \mathbf{F}, \lambda; \mathcal{D}\big) = \frac{1}{|\mathcal{D}|\,|\mathcal{Y}|} \sum_{(x,y)\in\mathcal{D}} \sum_{c\in\mathcal{Y}} \Big[\Big(\frac{1}{T}\sum_{t=0}^{T-1} \hat{V}_{\text{clamp},c,t}(\mathbf{C}, \mathbf{F})\Big)^2 - \Big(\frac{1}{T}\sum_{t=0}^{T-1} \hat{V}_{c,t}\Big)^2 \Big] + \lambda \sum_{l=1}^{L} \big(\|\mathbf{c}_l\|_2 + \|\mathbf{f}_l\|_2\big). \tag{15}$$

CA is maintained by ensuring the same distribution of TMP on clean samples. The clamp covers L convolution blocks to mitigate backdoor patterns in both the pixel and feature space. The clamping parameters are sufficiently large, so no clamping takes effect at the beginning. They are gradually reduced and motivated by the L2 norm loss term to narrow down the clamping range to filter out abnormal weights for low ASR goals. The weight term λ helps to balance the conflict in achieving both goals.

C. Full-Life-Cycle-Defense

NDSBM can be further strengthened when cooperating with TMPBD. The end-to-end strategy starts by detecting the potential target label of the attack. The method flags the samples that are predicted to the target label by the original classifier. The clamped model is applied only to the suspicious sample. The idea is to avoid unnecessary mitigation on trustworthy samples.

VI. Experiment

To ensure the effectiveness of our proposed defense, we follow the optimal experimental setup suggested in the original SOTA attack paper [17], including model architecture, training protocols, and attack configurations. In the optimal setting, all attacks tested in the experiment: static, moving [20], and dynamic [17] backdoor attacks reach 100% ASR with near-negligible CA degradation. Note that the smart trigger backdoor attack [17] is discarded from the experiment because it is not maintained by 100% ASR on all datasets. We benchmark on the three widely recognized neuromorphic benchmark datasets, which are DVS128-Gesture [44], CIFAR10-DVS [45], and N-Caltech101 [46], for complete validation on generalizability and robustness of the method. The proposed algorithm is implemented on the Spikingjelly neuromorphic computing framework [47] known for its universality on both the Von Neumann architecture platform and the neuromorphic platforms [36], boosting the practical relevance of the proposed defense.

For each combination of attack method and dataset, we repeat the experiments ten times to ensure the robustness of the results. Each run uses a different attack target label to ensure that the choice of target label does not influence the results. DVS128-Gesture data attacks on the first ten labels. CIFAR10-DVS data attacks on all ten labels. N-Caltech101 attack on randomly selected ten labels. With random seed 42, selected classes include: $[81, 14, 3, 94, 35, 31, 28, 17, 13, 86]$. The ten clean repeated models for the control group are trained with different random seeds from zero to nine. All other randomness processes are configured with the same classical random seed 42 to ensure the reproducibility of the experiment.

For reproducibility, the complete source code for all implementations and experiments is publicly available at https://github.com/alexjiachenli/TMPBD-NDSBM.

A. Backdoor Detection

In this section, we evaluate the backdoor detection accuracy and attack label detection accuracy of our proposed TMPBD compared to the ANN defense adopted as the baseline. Although theoretical deficiencies prevented existing ANN backdoor defenses from working effectively on SNNs have been explored. Due to a lack of detailed experimental results or open-sourced implementations. We re-implement the existing defense to empirically validate the deficiencies. The ANN backdoor detection methods adopted for SNN in this experiment are NC [48], ABS [49], Neuron Simulation (NS), and MMBD [40]. Each defense is evaluated in 12 scenarios, combining three data sets with four model conditions (three attacks and one clean control group). Each scenario is repeated ten times to ensure robustness. The detection accuracy is calculated from the ratio of correct detections from 10 repetitions. The detection hyperparameters are chosen on the basis of the VRAM limit and the convergence speed of the data sets. Here we initialize 3 parallel synthetic inputs and optimize for 5000 epochs.

To ensure that the experiment on our adoption of the ANN defense is reproducible, we introduce the defense configuration in the following list. The information is also available in our published source code.

- In NC [48], for each label $c \in \mathcal{Y}$, we uniformly randomize a learnable putative trigger in the same dimension as the data x with a value ranging from 0 to 0.1. We regularize the trigger by the L1 norm during optimization. We used a Median Absolute Deviation (MAD) threshold of 2 for the anomaly detection, which is equivalent to the confidence level 95% in the normal distribution. We use FR for computing binary cross-entropy in the loss function.
- We adopt ABS [49] differently from the previous literature [17]. For each neuron in the last layer, we generate and optimize a synthetic input to maximize the FR of the neuron. The synthetic input is generated in the same way as a putative trigger in NC. During optimization, additional clamping is implemented in the range $[0, 1]$ to match the range of non-negative legal input. We calculate the average FR of each neuron after taking all synthetic inputs and mark the neuron suspicious if the average FR exceeds the threshold, which is 95% percentile in our implementation. The ABS then generates a putative trigger that can maximize the FR of all suspicious neurons. Finally, the algorithm clips the putative trigger on the clean dataset and checks whether a variation in model prediction results from the trigger.
- NS is not an existing standalone defense, but the suspicious neutron detection part in other defense frameworks such as ABS [49] or fine-pruning [50]. We consider it as a standalone defense, as it does not require access to clean samples. We interpret the non-empty output of the suspicious neuron list as a backdoor detected with attack labels associated with suspicious neurons.
- In MMBD [40], we replace logits with FR. We initialize three samples uniformly between 0 and 1 to optimize in parallel for a maximum of 5000 epochs to incorporate the additional memory and complexity of the neuromorphic data.

The results of the experiment are shown in Table II. Our proposed defense outperforms all existing defenses in detecting the existence of backdoors and attack labels on compromised models and falls short only by a small margin in detecting clean models. Among the adopted defenses, the two backdoor pattern reverser engineer-based detections, NC and ABS, despite having additional access to a small clean sample, have failed catastrophically. The NC makes dataset-specific predictions independent of the attack type. The NC predicts no attack for all DVS128-Gesture data, attack target label 2 for all CIFAR10-DVS models, and attack on label 89 for all N-Caltech101 models. However, ABS has successfully identified numerous suspect neurons associated with output labels, but failed to validate the existence of backdoor attacks with the putative trigger pattern. As a result, ABS predicts that all models are clean. This indicates that the failure of the reverse

	N	DVS128-Gesture				CIFAR10-DVS				N-Caltech101			
		Clean	Static	Moving	Dynamic	Clean	Static	Moving	Dynamic	Clean	Static	Moving	Dynamic
Backdoor Detection Accuracy													
NC [48]	50	**100%**	0%	0%	0%	0%	**100%**	0%	**100%**	0%	**100%**	**100%**	**100%**
ABS [49]	50	**100%**	0%	0%	0%	**100%**	0%	0%	0%	**100%**	0%	0%	0%
NS [49], [50]	0	**100%**	10%	50%	50%	60%	90%	50%	**100%**	60%	80%	0%	50%
MMBD [40]	0	**100%**	0%	80%	10%	**100%**	0%	30%	0%	80%	20%	70%	50%
TMPBD	0	90%	**100%**	**100%**	**100%**	80%	**100%**	**100%**	**100%**	90%	90%	90%	**100%**
Attack Label Detection Accuracy													
NC [48]	50	**100%**	0%	0%	0%	0%	10%	0%	10%	0%	0%	0%	0%
ABS [49]	50	**100%**	0%	0%	0%	**100%**	0%	0%	0%	**100%**	0%	0%	0%
NS [49], [50]	0	**100%**	0%	10%	0%	60%	0%	0%	20%	60%	0%	0%	0%
MMBD [40]	0	**100%**	0%	10%	0%	**100%**	0%	10%	0%	80%	10%	0%	50%
TMPBD	0	90%	**100%**	**100%**	**100%**	80%	**100%**	**100%**	**100%**	90%	**90%**	**70%**	**100%**

TABLE II: Backdoor and attack label detection accuracy (%) of our proposed TMPBD against various defense methods across three neuromorphic datasets (DVS128-Gesture, CIFAR10-DVS, and N-Caltech101), evaluated under four attack types: Clean, Static trigger, Moving trigger, and Dynamic trigger. **N** denotes the number of samples per class used during detection. The highest accuracy in each row is highlighted in bold.

engineer-based approach in SNNs is potentially due to the exponentially larger search space of neuromorphic data.

By excision of the pattern, reverse engineering, and validation step, the NS shows acceptable backdoor detection accuracy, especially in two datasets transformed from the original static image form. However, NS fails to locate the attack target label even after detecting the attack. The MMBD is worse at detecting backdoors compared to neuron simulation, but locates the target label more accurately once the attack is detected. The advantage is inferred from the robustness of the MM statistic over the absolute value of FR during optimization. The performance of those two defenses further validates the theorem that the backdoor causes overfitting, especially for the dynamically triggered attack samples. Our proposed method improves over MMBD by utilizing the MM statistic of TMP instead of FR, which is more informative.

B. Backdoor Mitigation

The backdoor mitigation defense aims to alleviate the effect of the backdoor so that the poisoned model is no longer sensitive to trigger patterns. In this experiment, we focus on the DVS128-Dataset [44] with the corresponding SNN model architecture [4] to compare the performance of different mitigation strategies on different trigger types in a controlled environment. The attack types involved in this experiment are a clean control group, static trigger attack, moving trigger attack [20], and dynamic trigger attacks [17].

The mitigation defense is evaluated by the ability to reduce ASR while maintaining CA. In this experiment, ASR and CA are evaluated by the test set, excluding the data involved in mitigation and training to avoid optimistic bias and lack of generalization. Furthermore, when assessing the ASR, the test sample with the same label as the target attack label is excluded following common practice [51] due to the inability to identify the label because of the backdoor effect or discriminative characteristics of the class. The mathematical representation of ASR is shown below:

$$\text{ASR} = \frac{|\{(x_i + \delta_i, \tilde{y}) \mid i \leq r,\, y_i \neq \tilde{y} \wedge h(x_i + \delta_i) = \tilde{y}\}|}{|\{(x_i + \delta_i, \tilde{y}) \mid i \leq r,\, y_i \neq \tilde{y}\}|} \tag{16}$$

We adopt supervised unlearning-based fine-tuning defense and clamping-based MMBM for SNNs for reference, as they would require access to the label information of the small clean set that violated our assumption on the defender's capability in the threat model. The necessary modifications to MMBM have been made to accommodate the SNN situation. Specifically, we set the weight amendment factor $a = 1.2$, the learning rate to 0.1, the target CA to 85%, and the initial $c = 1e - 5$. As discussed, due to the lack of activation and the threshold nature of membrane potential in SNNs, the MMBM are modified to max clamping the same weights as clamped by the proposed mitigation. The training accuracy is calculated via MSE over FR and one-hot encoded true label.

For the baseline, we modified the fine-tuning to use the predicted label as a putative label for fine-tuning, referred to as self-tuning. We also performed ablation tests on different clamping methods of the proposed weight clamping approach: max clamping, absolute clamping, and dual clamping. Finally, we experiment with the combination of the proposed attack label detection and dually clamping end-to-end backdoor defense pipeline. Note that although NC can mitigate after detection, the poor detection performance indicates that there is no experimental value for further mitigation with NC. Therefore, NC is discarded in the mitigation experiment. All of the defense runs for 50 epochs and have access to the first 20 samples from each class (around 2/3 of the total testing set).

The experiment is repeated ten times for each combination with a different attack target label or random seed for clean models. The average and standard deviation of CA and ASR

	Clean		Static		Moving		Dynamic	
	CA(%) ↑	ASR(%) ↓	CA(%) ↑	ASR(%) ↓	CA(%) ↑	ASR(%) ↓	CA(%) ↑	ASR(%) ↓
Original	97.65±1.03	0.31±0.99	98.09±0.99	100.00±0.00	97.21±1.29	100.00±0.00	84.71±12.48	100.00±0.00
Supervised mitigation methods requiring test labels								
Fine-Tuning [50]	64.56±6.63	3.00±4.70	56.32±6.97	4.38±11.26	70.29±13.98	5.91±12.98	88.53±5.54	3.28±4.51
MMBM [40]	73.09±8.38	2.90±2.99	82.50±4.67	46.06±33.33	73.68±5.11	18.34±19.01	71.76±16.08	1.40±2.49
Unsupervised mitigation methods NOT requiring test labels								
Self-Tuning [50]	7.20±3.13	11.44±19.89	5.88±1.39	9.06±20.29	7.20±2.81	15.72±19.66	6.47±1.73	25.56±20.18
Max Cla.	83.09±3.81	2.28±4.30	84.41±4.91	85.81±20.80	89.27±4.50	75.81±25.78	88.83±4.00	19.38±13.39
Abs. Cla.	84.26±4.75	1.34±2.09	83.82±7.47	84.22±16.31	87.21±6.05	68.13±26.15	89.12±2.52	20.81±14.50
NDSBM	72.50±6.43	3.69±10.27	72.21±6.56	30.41±25.92	83.38±8.29	29.87±19.92	89.86±3.21	8.44±9.91
TMPBD+NDSBM	97.06±1.55	0.31±0.99	96.33±3.12	19.94±26.48	95.88±3.00	38.12±35.44	92.06±4.29	2.81±3.95

TABLE III: The average value and standard deviation of CA and ASR before and after mitigation with our proposed NDSBM and TMPBD+NDSBM frameworks against different backdoor attack mitigation defenses on the DVS128-Gesture dataset under clean, static, moving, and dynamic triggers.

of different mitigation methods in different types of attacks are shown in Table III.

For supervised mitigation, MMBM and fine-tuning outperform each other under different conditions with compromise, where lowering ASR often lowers CA simultaneously. In general, modified MMBM is more robust among supervised mitigation methods, especially since original MMBM is more computationally expensive to bypass [40] compared to fine-tuning [17], [21]. The observation demonstrates the feasibility of weight clamping. In a more practical but challenging unsupervised mitigation setting, self-turning fails due to a disastrous reduction in CA. The result shows that even with an original CA as high as 97.65% on average, the accumulated putative label's small error destroyed the classifier's original behavior. Among all clamping approaches, dual clamping is most effective in reducing ASR, although it has a slight degradation in CA as a trade-off.

The main drawback of the proposed method is the non-negligible drop in CA. However, there is a workaround that only feeds suspicious samples to the clamped model. The suspicious samples are defined as samples predicted to the known attack target label by the compromised model. This solution is based on the accurate prediction of attack target labels, which has been achieved with our proposed TMPBD method, with an accuracy of 100%. By combining the TMPBD with NDSBM, the pipeline can nearly completely eliminate backdoors in dynamic attacks and significantly reduce ASR in other attacks with negligible degradation in CA.

VII. Related Work

Backdoor defense research for SNNs is still in its infancy. Most prior attempts simply port post-training defenses from ANNs, yet the spike-driven computation, binary activations, and temporally coded inputs of SNNs undermine their effectiveness. We organize the literature by the defense mechanism and highlight, for each category, the SNN-specific obstacles that remain unsolved until our work.

A. Activation-Analysis Defenses

Artificial Brain Stimulation (ABS). ABS identifies neurons strongly correlated with an attacker's target label, reconstructs a trigger, and tests it on clean inputs [49]. When frames are collapsed for SNNs, the lack of ReLU "turn points" yields many false positives [17].

Maximum-Margin Backdoor Detection (MMBD). MMBD replaces ABS's turn-point heuristic with MM statistics of logits [40]. Although this removes the ReLU dependency, the real-valued activations relied on by MMBD are absent in SNNs, so detection accuracy degrades in SNNs.

B. Reverse-Engineering Defenses

Neural Cleanse (NC). NC searches for a minimal per-class trigger and flags classes whose trigger is unusually small [48]. In SNNs, the search space explodes (neuromorphic inputs have an extra temporal dimension) and the binary spike output provides little gradient guidance, so NC becomes prohibitively slow and inaccurate.

Unsupervised Anomaly Detection (UAD). Xiang et al. learn class-specific perturbations without training data and apply statistical testing on the perturbation norms [52]. Their optimization assumes softmax logits and has not been adapted to spike trains.

C. Parameter-Repair Defenses

Fine-Pruning. Removing low-contribution neurons can excise backdoor-related units in ANNs [50]. Because SNNs encode information in precise spike timings, nearly every neuron is indispensable; pruning slashes CA while barely reducing ASR [17].

Maximum-Margin Backdoor Mitigation (MMBM). MMBM bounds suspicious activations during fine-tuning instead of deleting neurons [40]. Porting the method to SNNs requires clamping membrane potentials, which have not been systematically studied.

D. Our Position in the Field

The above defenses either *(i)* depend on ReLU-style activations, *(ii)* perform gradient-heavy trigger search infeasible for spike data, or *(iii)* degrade CA because they treat SNN neurons like ANN activations. Our proposed TMPBD + NDSBM is the first full-lifecycle defense designed expressly for SNNs and overcomes each blocker:

VIII. Threats to Validity

This section discusses the potential threats to the validity of our experimental results and how we address or mitigate them.

A. Scalability

One trade-off of any data-free backdoor detection method is the additional computational overhead incurred by the generation and optimization of synthetic data used for detection. However, we argue that scalability is not a blocker of the proposed TMPBD. The backdoor detection of a classifier for the DVS128-Gesture dataset is faster than model training (14 min 30 s vs. 18 min 45 s on RTX5090), while taking much less VRAM. For backdoor mitigation, NDSBM takes only 2 min 24 s to mitigate the discussed model. Moreover, TMPBD was cleverly designed to compute the MM of each class independently, suggesting that detection can be sped up by parallelization up to the factor of the number of classes, i.e., 11 for the DVS128-Gesture dataset.

B. Adaptive Attacker

Here, we evaluate the robustness of TMPBD against adaptive attackers with defense knowledge. As discussed in Section IV-A, TMPBD detects abnormal overfitting phenomena resulting from backdoor attacks. Intuitively, an adaptive attacker would attempt to suppress such a phenomenon to bypass TMPBD. We consider two adaptive attack approaches: amplitude-suppression adaptation (ASA) and peak-alignment adaptation (PAA). ASA is designed to depress the absolute membrane potential of the ATC. PAA, inspired by adaptive attack from [40], attempts to align TMP of ATC with the largest non-target TMP by minimizing the margin. In the context of Figure 1, ASA blends the red line into the gray lines, while PAA squeezes the red line closer to the cluster of the gray lines.

Both approaches were achieved by introducing an additional term controlled by a loss-penalty weight to the loss function during the model training stage as follows:

$$\mathcal{L}_{\text{ASA}} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{ASA}}\, \mathbb{E}_{x\sim D}\Big[\Big|\underbrace{\frac{1}{T}\sum_{t=0}^{T-1}\hat{V}_{a,t}(x)}_{\text{ATC TMP}}\Big|\Big] \tag{17}$$

$$\mathcal{L}_{\text{PAA}} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{PAA}}\, \mathbb{E}_{x\sim D}\Big[\max\Big(0, \underbrace{\tfrac{1}{T}\sum_{t=0}^{T-1}\hat{V}_{a,t}(x)}_{\text{ATC TMP}} - \underbrace{\max_{k\neq a}\tfrac{1}{T}\sum_{t=0}^{T-1}\hat{V}_{k,t}(x)}_{\text{largest non-target TMP}}\Big)\Big] \tag{18}$$

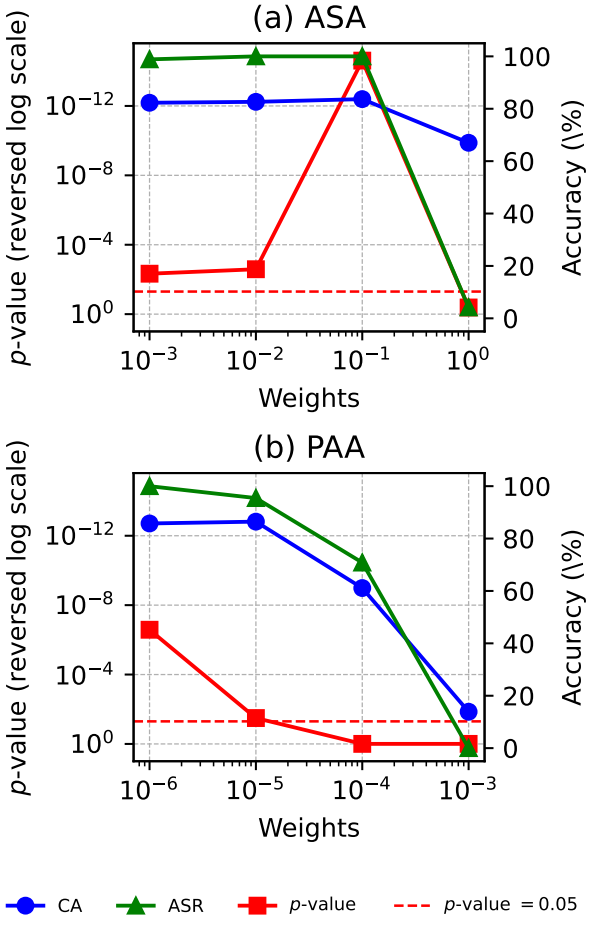

Fig. 4: Statistical significance (p-value, left y-axis) and performance (CA, ASR, right y-axis) of TMPBD under (a) amplitude-suppression adaptation (ASA) and (b) peak-alignment adaptation (PAA) on the DVS128–Gesture dataset.

We conduct the experiment on adaptive attack under the same setting as empirical studies from Section IV-A, with a static attack on the DVS128-Gesture dataset. From the results shown in Figure 4, we observe that the detection cannot be evaded unless we increase the penalty weight over a threshold, which would effectively also drop the CA and ASR to an impractically low level. Specifically, CA and ASR drop by 17.37% and 95.83% under ASA, 27.08% and 28.79% under PAA to bypass the detection. Such drastic performance degradation renders the backdoor practically useless, confirming TMPBD's robustness against these adaptive strategies.

Existing adaptive attack studies increase every non-target logit (equivalently, every non-target TMP in SNNs) to reduce the ATC margin, such as [40]. However, this approach introduces significant computation overheads, making it impractical. While existing adaptive attacks mainly focus on margin reduction, to our best knowledge, the distributed or timing-

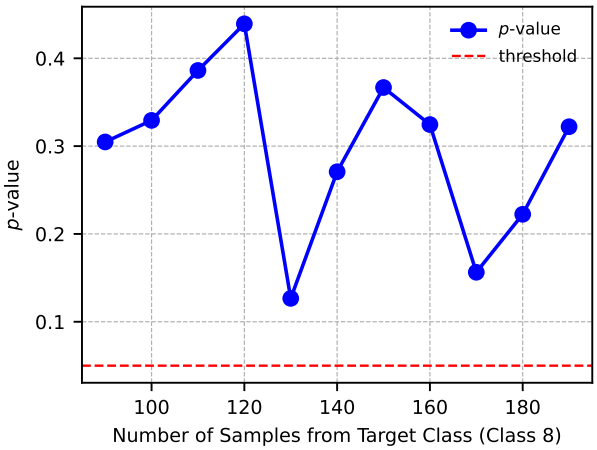

Fig. 5: Detection p-values of clean models with varying sample counts from the target class (class 8) in the DVS128–Gesture dataset, with 90 samples fixed for other classes.

based evasions in SNNs have yet to be explored.

C. Imbalanced Dataset

The majority of model-based backdoor detection algorithms detect the overfitting of a small-sized backdoor pattern biased the model toward the ATC. Apart from backdoor attacks, imbalanced training data is another cause of overfitting, resulting in a model biased toward the majority class. Past literature suggests that MMBD falsely detects the clean model trained on severely imbalanced data as poisoned [40]. However, the experiment in Figure 5 shows that the detection results in TMPBD are invariant as the imbalance level increases. Training on more imbalanced data with more than double the sample of a class over others is impractical, as it harms the CA dramatically.

D. False Positive Issue

Although the proposed TMPBD outperforms all existing defenses in the detection model with a backdoor. Detection is still occasionally too sensitive, and the clean model was misclassified as poisoned under the current default significant threshold $\alpha = 0.05$. The flaw can be resolved with additional domain knowledge for domain-specific threshold tuning. This additional information is accessible under the classical threat model from the past literature. The results of significant threshold tuning are shown in Table IV. Notably, for the CIFAR10-DVS data, reducing α to 0.02 would lower the FPR at no cost in the TPR, demonstrating the effectiveness of the adjustment.

E. All-to-All attacks

The anomaly detection mechanism only validates the most suspicious class in TMPBD to detect all-to-one attacks. However, we observe a phenomenon, shown in Figure 6, similar to the previous literature [40], suggesting that the distribution of

Dataset	α	**0.05**	**0.02**	**0.01**	**0.005**
DVS128-Gesture	TPR% ↑	100	95	90	85
	FPR% ↓	20	20	10	10
CIFAR10-DVS	TPR% ↑	100	100	100	95
	FPR% ↓	20	10	10	10
N-Caltech101	TPR% ↑	95	95	90	90
	FPR% ↓	10	10	10	0

TABLE IV: True Positive Rate (TPR) and False Positive Rate (FPR) across datasets for different significance thresholds α. The defender aims to maximize TPR while minimizing FPR.

TMP MM of poisoned samples is distributed differently from clean samples with slight overlap, which serves as the basis for detection. The distribution was collected from ten clean models and ten all-to-all static attack models on DVS128-Gesture data. The attack pairs each neighboring class into the source class-ATC pair (that is, the sample with source class 0 is poisoned to class 1 with static trigger) [18]. We argue that with access to additional domain knowledge that is accessible in a common defense setting [49], we can calibrate the detection threshold to detect an arbitrary number of target classes of the backdoor in an attack of all-to-all or all-to-x without knowing the number of attack classes. Shown in Figure 6, the ROC curve of the margin-based detector with a full AUC of 0.8397, which shows a strong overall discriminative power. The result is particularly impressive given that this all-to-all attack is not effective and only has ASR 63.86±15.46%. To our best knowledge, there are no proposed all-to-all attacks for SNN and we tried our best to adopt the current static [20] all-to-one attack for the all-to-all setting. Note that doubling the poison rate will counterintuitively not improve the effectiveness of the attack and reduce the ASR to 11.81% and the dynamic attack [17] failed in the all-to-all setting with 0% ASR. In particular, the proposed NDSBM backdoor mitigation method does not assume that the number of target classes remaining effective in the all-to-all attack reduced the ASR from 63.86±15.46% to 16.04±7.36%.

F. Intrinsic Backdoored Data

It has been an open problem to detect a backdoor attack for a model trained with intrinsic backdoored data [40] for both SNN and ANN. The intrinsic backdoored data describes the data with class discriminative features that behave similarly to the backdoor pattern. For example, the model designed to classify if there is a backdoor in the sample always predicts true in the present backdoor pattern, which behaves as a backdoor-attacked model and will be detected as attacked. However, the model is, in fact, as designed to be, and is not under any backdoor attack. The phenomenon can also be observed in overly simple classification problems where the class discriminative features are as small as a few pixels. An example is N-MNIST [46], which uses a classifier that can make a high-confidence prediction based on a few pixels.

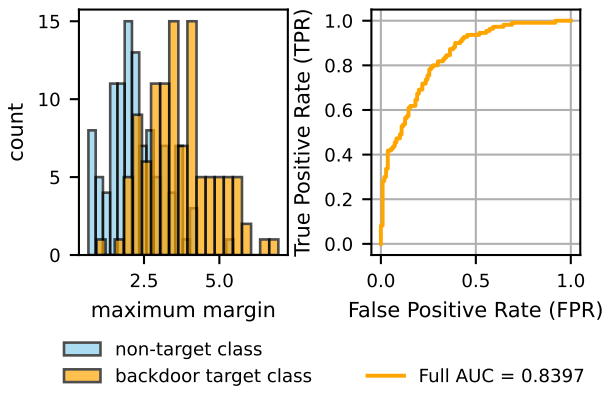

Fig. 6: Analysis of maximum margin statistics and detection performance. (a) Margin distributions of clean classes (non-target) vs. ATC. (b) ROC curve of TMPBD detection, achieving full AUC = 0.8397 on DVS128-Gesture under all-to-all static attacks.

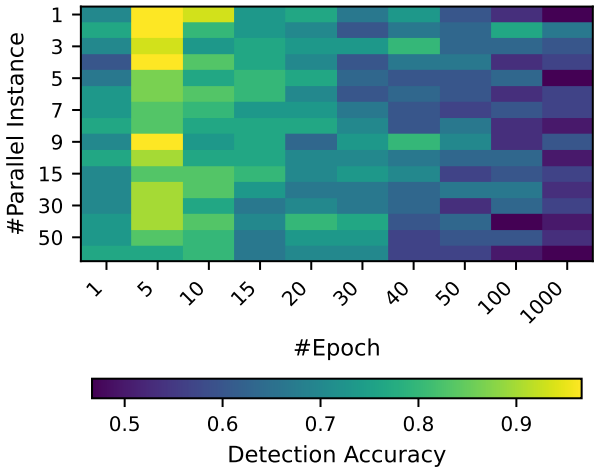

Fig. 7: Heatmap of detection accuracy under grid search for different numbers of parallel instances and numbers of epoch with TMPBD for N-MNIST dataset.

TMPBD is incapable of distinguishing an intrinsic backdoor from a real backdoor. However, by accessing additional domain knowledge, we can optimize an optimal detection hyperparameter to make TMPBD effective on such a dataset. Figure 7 suggests that optimizing the hyperparameters of n-epoch and n-parallel on N-MNIST can lead to detection accuracy of up to 97% via a "few-shot detection". We hypothesize that the intrinsic backdoor can be distinguished by the rate of convergence of the MM instead of the final optimized MM value. We hope that our first indicative finding toward solving this open question can inspire future research.

IX. Conclusion

This study addresses the challenge of backdoor attacks in Spiking Neural Networks by proposing two novel defenses. Temporal Membrane Potential Backdoor Detection (TMPBD) leverages the Maximum Margin statistic of temporal membrane potential to achieve unsupervised, post-training backdoor detection without the requirement of attack knowledge or additional data. Neural Dendrites Suppression Backdoor Mitigation (NDSBM) effectively reduces the backdoor effect while preserving clean accuracy through dual clamping of neural dendrites. The evaluations of the benchmark data sets demonstrated the near-optimal detection accuracy of TMPBD and the ability of NDSBM to lower the attack success rates to as low as 2.81% on average with the help of TMPBD. The proposed defenses have outperformed all the existing backdoor defense techniques. The paper discussed the scalability and robustness of the proposed model under an adaptive attacker. The paper also illustrated that under a relaxed setting with access to domain knowledge, the proposed approach can be robust to imbalanced datasets, false positive issues, all-to-all attacks, and intrinsic backdoored data with minimal modification.

Acknowledgment

This research has been supported by ARC Discovery Projects DP190102835, DP220102803, DP240102140 and Linkage Project LP220200649. We thank the anonymous reviewers for their insightful suggestions and comments.

References

[1] S. Ghosh-Dastidar and H. Adeli, "SPIKING NEURAL NETWORKS," *International Journal of Neural Systems*, vol. 19, no. 04, pp. 295–308, Aug. 2009.

[2] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, Mar. 2019.

[3] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training Spiking Neural Networks Using Lessons From Deep Learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1016–1054, Sep. 2023, conference Name: Proceedings of the IEEE.

[4] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 2641–2651, iSSN: 2380-7504.

[5] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge: Cambridge University Press, 2014.

[6] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-yolo: spiking neural network for energy-efficient object detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 11 270–11 277, issue: 07.

[7] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[8] S. Kundu, G. Datta, M. Pedram, and P. A. Beerel, "Spike-Thrift: Towards Energy-Efficient Deep Spiking Neural Networks by Limiting Spiking Activity via Attention-Guided Compression," 2021, pp. 3953–3962.

[9] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 34–49, 2020, publisher: IEEE.

[10] A. Viale, A. Marchisio, M. Martina, G. Masera, and M. Shafique, "Carsnn: An efficient spiking neural network for event-based autonomous cars on the loihi neuromorphic research processor," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–10.

[11] M.-J. Escobar, G. S. Masson, T. Vieville, and P. Kornprobst, "Action recognition using a bio-inspired feedforward spiking network," *International journal of computer vision*, vol. 82, pp. 284–301, 2009, publisher: Springer.

[12] S. Loiselle, J. Rouat, D. Pressnitzer, and S. Thorpe, "Exploration of rank order coding with spiking neural networks for speech recognition," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4. IEEE, 2005, pp. 2076–2080.

[13] N. Kasabov, V. Feigin, Z.-G. Hou, Y. Chen, L. Liang, R. Krishnamurthi, M. Othman, and P. Parmar, "Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke," *Neurocomputing*, vol. 134, pp. 269–279, 2014, publisher: Elsevier.

[14] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 x 128 1.5% Contrast Sensitivity 0.9% FPN 3 µs Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, 2013, publisher: IEEE.

[15] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor Learning: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–18, 2022, conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[16] X. Wang, X. Liu, S. Lai, X. Yi, and X. Yuan, "SIGuard: Guarding secure inference with post data privacy," in *Proceedings 2025 Network and Distributed System Security Symposium*. Internet Society.

[17] G. Abad, O. Ersoy, S. Picek, and A. Urbieta, "Sneaky Spikes: Uncovering Stealthy Backdoor Attacks in Spiking Neural Networks with Neuromorphic Data," in *Proceedings 2024 Network and Distributed System Security Symposium*, 2024, arXiv:2302.06279 [cs].

[18] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[19] H. Li, Y. Wang, X. Xie, Y. Liu, S. Wang, R. Wan, L.-P. Chau, and A. C. Kot, "Light Can Hack Your Face! Black-box Backdoor Attack on Face Recognition Systems," Sep. 2020, arXiv:2009.06996 [cs].

[20] G. Abad, O. Ersoy, S. Picek, V. J. Ramírez-Durán, and A. Urbieta, "Poster: Backdoor Attacks on Spiking NNs and Neuromorphic Datasets," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. Los Angeles CA USA: ACM, Nov. 2022, pp. 3315–3317.

[21] R. Riaño, G. Abad, S. Picek, and A. Urbieta, "Flashy Backdoor: Real-world Environment Backdoor Attack on SNNs with DVS Cameras," Nov. 2024, arXiv:2411.03022.

[22] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.

[23] T. Bu, J. Ding, Z. Hao, and Z. Yu, "Rate gradient approximation attack threats deep spiking neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7896–7906.

[24] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biol. Cybern.*, vol. 20, no. 3-4, pp. 121–136, 1975.

[25] 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*, vol. 36, no. 6, pp. 51–63, Nov. 2019, conference Name: IEEE Signal Processing Magazine.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *CoRR*, Dec. 2014.

[27] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4, pp. 185–196, Jun. 1993.

[28] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training Deep Spiking Neural Networks Using Backpropagation," *Frontiers in Neuroscience*, vol. 10, Nov. 2016, publisher: Frontiers.

[29] M. Weber, X. Xu, B. Karlaš, C. Zhang, and B. Li, "Rab: Provable robustness against backdoor attacks," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1311–1328.

[30] X. Chen, A. Salem, D. Chen, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang, "BadNL: Backdoor Attacks against NLP Models with Semantic-preserving Improvements," in *Annual Computer Security Applications Conference*. Virtual Event USA: ACM, Dec. 2021, pp. 554–569.

[31] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, Jan. 2009, conference Name: IEEE Signal Processing Magazine.

[32] T. A. Nguyen and A. Tran, "Input-Aware Dynamic Backdoor Attack," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 3454–3464.

[33] K. Doan, Y. Lao, W. Zhao, and P. Li, "LIRA: Learnable, Imperceptible and Robust Backdoor Attacks," 2021, pp. 11 966–11 976.

[34] P. Kirkland, G. Di Caterina, J. Soraghan, and G. Matich, "Neuromorphic technologies for defence and security," in *Emerging Imaging and Sensing Technologies for Security and Defence V; and Advanced Manufacturing Technologies for Micro- and Nanosystems in Security and Defence III*, M. Farsari, J. G. Rarity, F. Kajzar, A. Szep, R. C. Hollins, G. S. Buller, R. A. Lamb, M. Laurenzis, A. Camposeo, L. Persano, L. E. Busse, M. Dušek, P. M. Alsing, M. L. Fanto, and R. Zamboni, Eds. Online Only, United Kingdom: SPIE, Sep. 2020, p. 27.

[35] B. Wu, H. Zhang, X. Yang, S. Wang, M. Xue, S. Pan, and X. Yuan, "GraphGuard: Detecting and counteracting training data misuse in graph neural networks," in *Proceedings 2024 Network and Distributed System Security Symposium*. Internet Society.

[36] G. Orchard, E. P. Frady, D. B. D. Rubin, S. Sanborn, S. B. Shrestha, F. T. Sommer, and M. Davies, "Efficient Neuromorphic Signal Processing with Loihi 2," in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, Oct. 2021, pp. 254–259, iSSN: 2374-7390.

[37] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022, pp. 703–718.

[38] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 182–199.

[39] S. Bubeck, "Convex Optimization: Algorithms and Complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 3-4, pp. 231–357, 2015.

[40] H. Wang, Z. Xiang, D. J. Miller, and G. Kesidis, "MM-BD: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic," in *IEEE symposium on security and privacy*, 2024.

[41] G. Chen, P. Peng, G. Li, and Y. Tian, "Training Full Spike Neural Networks via Auxiliary Accumulation Pathway," Jan. 2023, arXiv:2301.11929 [cs].

[42] Z. Xiang, D. J. Miller, and G. Kesidis, "Detection of Backdoors in Trained Classifiers Without Access to the Training Set," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1177–1191, Mar. 2022, conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[43] X. Xia, Z. Wang, R. Sun, B. Liu, I. Khalil, and M. Xue, "Edge unlearning is not "on edge"! an adaptive exact unlearning system on resource-constrained devices," in *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2546–2563, ISSN: 2375-1207.

[44] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha, "A Low Power, Fully Event-Based Gesture Recognition System," 2017, pp. 7243–7252.

[45] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "CIFAR10-DVS: An Event-Stream Dataset for Object Classification," *Frontiers in Neuroscience*, vol. 11, May 2017, publisher: Frontiers.

[46] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in neuroscience*, vol. 9, p. 159859, 2015, publisher: Frontiers.

[47] W. Fang, Y. Chen, J. Ding, Z. Yu, T. Masquelier, D. Chen, L. Huang, H. Zhou, G. Li, and Y. Tian, "SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence," *Science Advances*, vol. 9, no. 40, p. eadi1480, Oct. 2023.

[48] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 707–723.

[49] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. London United Kingdom: ACM, Nov. 2019, pp. 1265–1282.

[50] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks," in *International Symposium on Research in Attacks, Intrusions, and Defenses*, M. Bailey, T. Holz, M. Stamatogiannakis, and S. Ioannidis, Eds., vol. 11050. Cham: Springer International Publishing, 2018, pp. 273–294, series Title: Lecture Notes in Computer Science.

[51] N. Karim, A. A. Arafat, A. S. Rakin, Z. Guo, and N. Rahnavard, "Fisher Information guided Purification against Backdoor Attacks," Sep. 2024, arXiv:2409.00863.

[52] Z. Xiang, D. J. Miller, and G. Kesidis, "Detection of backdoors in trained classifiers without access to the training set," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1177–1191, 2020, publisher: IEEE.

[53] A. Samadzadeh, F. S. T. Far, A. Javadi, A. Nickabadi, and M. H. Chehreghani, "Convolutional spiking neural networks for spatio-temporal feature extraction," vol. 55, no. 6, pp. 6979–6995.

APPENDIX

EXPERIMENT DETAIL

A. Details of Dataset

The experiment is carried out in the widely used neuromorphic benchmark datasets DVS128-Gesture [44], CIFAR10-DVS [45], and N-Caltech101 [46]. The techniques involved in backdoor attacks have been tested and have performed well in the original paper proposing the attacks [17]. The benchmark datasets are sufficiently complex for the classification task that practically represents the real-world scenario. They also cover a wide spectrum of data properties. The DVS128-Gesture dataset consists of human gesture movements with 29 different subjects under three different illumination conditions, directly captured by a DVS camera that is closest to the real-world situation. In contrast, the other two are pre-existing popular static images in the computer vision research field converted into neuromorphic data format via capture of the image showing on an LCD display with a DVS camera performing Repeated Closed-Loop Smooth (RCLS) Movement [45]. Although the converted dataset is less practical, the CIFAR10-DVS provides the possibility of performance comparison with existing research in the field of ANNs, while the N-Caltech101 dataset contains 100 object classes plus one background class, offering insight into the situation with a large number of classification labels.

Following the optimal setting from the reference paper [17]. We employ the same training settings as the original paper. Notable is the original paper choosing learning epochs of 28 for CIFAR10-DVS, 63 for DVS128-Gesture, and 30 for N-Caltech101 to align with the same CA in SOTA research [53].

B. Details of Training Configurations

We adopted the commonly used corresponding network architecture for the classifier to defend in the related works [4], [17]. For the CIFAR10-DVS dataset, the network architecture comprises two convolutional layers, each followed by batch normalization and max pooling. This is succeeded by two fully connected layers with dropout, and a final voting layer of size ten to enhance classification robustness [17]. In contrast, the networks used for the DVS128-Gesture and N-Caltech101 datasets consist of five convolutional layers (with batch normalization and max pooling), two fully connected layers with dropout, and a voting layer. Further architectural details are available in our code repository.

C. Details of Backdoor Pattern

We follow the same recommended hypermeter for all attacks in the original literature [17] for all datasets to ensure a controlled environment and ensure that all attacks in all datasets are effective and reach 100% ASR with nearly negligible CA degradation. Specifically, for static triggers, the trigger patch is located in the top left corner with a size of 10% image with polarity=1. The attack pattern is static for all samples in all time steps. The pattern is injected into 10% of the training set. The moving triggers are initialized similarly, but move two pixels to the right every time step. For the dynamic trigger, we set a hyperparameter α, weight controlling the trade-off between CA and ASR in the loss function, to 0.5 to evenly balance CA and ASR. The visibility factor γ is set to 0.01 to maximize stealthiness while maintaining high ASR and CA.