R-SNN: An Analysis and Design Methodology for Robustifying Spiking Neural Networks against Adversarial Attacks through Noise Filters for Dynamic Vision Sensors

To appear at the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021).

R-SNN: An Analysis and Design Methodology for Robustifying Spiking Neural Networks against Adversarial Attacks through Noise Filters for Dynamic Vision Sensors

Alberto Marchisio[1,*], Giacomo Pira[2,*], Maurizio Martina[2], Guido Masera[2], Muhammad Shafique[3]
[1]*Institute of Computer Engineering, Technische Universität Wien, Vienna, Austria*
[2]*Department of Electronics and Telecommunication, Politecnico di Torino, Turin, Italy*
[3]*Division of Engineering, New York University, Abu Dhabi, UAE*
Email: alberto.marchisio@tuwien.ac.at, giacomo.pira@studenti.polito.it
{*maurizio.martina, guido.masera*}*@polito.it, muhammad.shafique@nyu.edu*

***Abstract*— Spiking Neural Networks (SNNs) aim at providing energy-efficient learning capabilities when implemented on neuromorphic chips with event-based Dynamic Vision Sensors (DVS). This paper studies the robustness of SNNs against adversarial attacks on such DVS-based systems, and proposes *R-SNN*, a novel methodology for robustifying SNNs through efficient DVS-noise filtering. We are the first to generate adversarial attacks on DVS signals (i.e., frames of events in the spatio-temporal domain) and to apply noise filters for DVS sensors in the quest for defending against adversarial attacks. Our results show that the noise filters effectively prevent the SNNs from being fooled. The SNNs in our experiments provide more than 90% accuracy on the DVS-Gesture and NMNIST datasets under different adversarial threat models.**

Index Terms: Spiking Neural Networks, SNNs, Deep Learning, Adversarial Attacks, Security, Robustness, Defense, Filter, Perturbation, Noise, Dynamic Vision Sensors, DVS, Neuromorphic, Event-Based, DVS-Gesture, NMNIST.

I. Introduction

Spiking Neural Networks (SNNs) aim at providing energy-efficient learning capabilities in a wide variety of machine learning applications, e.g., autonomous driving [1], healthcare [2], and robotics [3]. Unlike traditional (i.e., non-spiking) Deep Neural Networks (DNNs), the SNNs are biologically plausible, enabling event-based communication between neurons which simulate the human brain's processing in a relatively closer manner [4]. Moreover, the results both in terms of power/energy efficiency and real-time classification performance make the SNNs appealing for being implemented in resource-constrained embedded systems [5]. By leveraging the spike-based communication between neurons, SNNs exhibit a lower computational load, as well as a reduction in the latency, compared to the equivalent DNN implementations [6].

Along with the development of efficient SNNs implemented on specialized neuromorphic accelerators (e.g., IBM TrueNorth [7] and Intel Loihi [8]), another advancement in the field of neuromorphic hardware has come from the new generation of the Dynamic Vision Sensor (DVS), i.e., an event-based camera sensor [9]. Unlike a classical frame-based camera, the DVS emulates the behavior of the human retina, by recording the information in form of a sequence of spikes, which are generated every time a change of light intensity is detected. The event-based behavior of these sensors pairs well with SNNs implemented onto the neuromorphic hardware, i.e., the output of a DVS camera can be used as the direct input of an SNN to elaborate events in real-time.

[*]These authors contributed equally to this work.

A. Target Research Problem and Scientific Challenges

Similar to the case of traditional DNNs, the trustworthiness of SNNs is also threatened by adversarial attacks, i.e., small and imperceptible input perturbations aiming at crafting the network's correct functionality. Although some preliminary studies have been conducted [10][11][12][13], such a problem is relatively new and unexplored for practical SNN-based systems. In particular, DVS-based systems have not been investigated for SNN security. As a starting point, the methods for designing robust SNNs can be derived from the recent advancements of the defense mechanisms for DNNs, where studies have focused on adversarial learning algorithms [14], loss/regularization functions [15], and image preprocessing [16]. The latter approach basically consists of suppressing the adversarial perturbation through dedicated filtering. Noteworthy, for the SNN-based systems fed by DVS signals, the attacks and preprocessing-based defense techniques for frame-based sensors cannot be directly applied due to differences in the signal properties. Therefore, specialized noise filters for DVS sensors [17] must be employed.

As per our knowledge, the impact of filtering on DVS sensors for secure neuromorphic computing is an unexplored and open research problem. Towards this, we devise *R-SNN*, a novel methodology employing attack-resistant noise filters on DVS signals as a defense mechanism for robustifying SNNs against adversarial attacks. Since the DVS cameras contain also the temporal information, the generation of adversarial perturbation is technically different w.r.t. traditional adversarial attacks on images, where only

the spatial information is considered. Hence, the temporal information needs to be leveraged for developing a robust defense.

B. Motivational Case Study

As a preliminary study for motivating our research in the above-discussed directions, we perform the following experiments. We trained a 4-layer Spiking CNN, with 2 convolutional layers and 2 fully-connected layers, for the DVS-Gesture dataset [18] using the SLAYER method [19], using an ML-workstation with two Nvidia GeForce RTX 2080 Ti GPUs. For each frame of events, we perturb the testing dataset by injecting uniform and normally-distributed random noise and measure the classification accuracy. Moreover, to mitigate the effect of the perturbations, the filter of [17] is applied, with different spatio-temporal parameters (s and t). The accuracy results w.r.t. different noise magnitude are shown in Fig. 1. As indicated by pointer ① in Fig. 1, the filter slightly reduces the accuracy of the SNN when no noise is applied. However, in the presence of noise, the SNN becomes much more robust when the filter is applied. For example, when considering normal noise with a magnitude of 0.55, the filter with $s = 1$ and $t = 5$ contributes to 64% accuracy improvement; see pointer ②. Such a filter works even better when uniformly-distributed noise is applied. Indeed, the perturbations with large magnitude of 0.85 and 1 are filtered out well, because the SNN maintains a relatively high accuracy of 85% and 74%, respectively; see pointer ③.

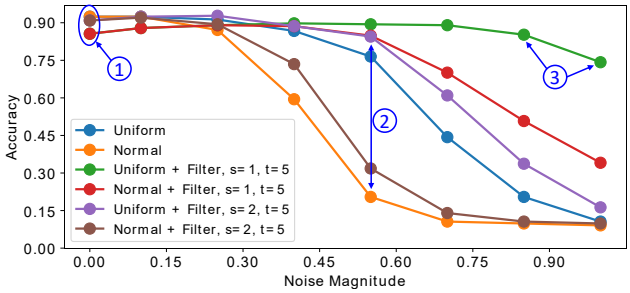

Fig. 1: Analyzing the impact of applying the normal and uniform noise to the DVS-Gesture dataset.

C. Our Novel Contributions

To address the above-discussed scientific problem, we propose *R-SNN*, an analysis and design methodology for robustifying SNNs. Our **key contributions** are as follows (see Fig. 2).

- We analyze the impact of noise filtering for DVS under multiple adversary threat models, i.e., by placing the filter at different stages of the system, or assuming different knowledge of the adversary. (**Section III-A**)
- We generate adversarial perturbations for the DVS signal to attack SNNs. (**Section III-B**)
- *R-SNN Design Methodology:* we propose a methodology to apply specialized DVS-noise filters for increasing the robustness of SNNs against adversarial attacks. (**Section III-C**)
- Our experimental results exhibit high SNN robustness against adversarial attacks, under different adversary threat models. (**Section IV**)
- For reproducible research, we release the code of the *R-SNN* filtering methodology for DVS-based SNNs on GitHub[1].

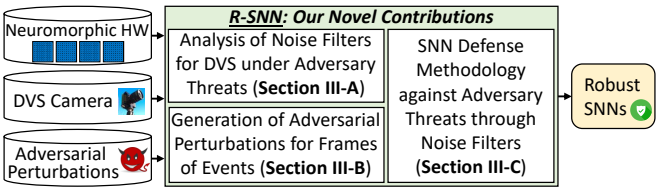

Fig. 2: Overview of our novel contributions and methodology.

II. Background

A. Spiking Neural Networks (SNNs)

SNNs, the third generation NNs [20], exhibit better biological plausibility compared to the traditional DNNs. Indeed, the event-based communication between neurons in SNNs resembles the human brain's functionality. Another key advantage of SNNs over the traditional DNNs is their improved energy-efficiency when implemented on Neuromorphic chips like Intel Loihi [8] or IBM TrueNorth [7]. Moreover, the recent development of DVS sensors [9] has further reduced the energy consumption of the complete system.

An example of the SNNs' functionality is shown in Fig. 3. The input is coded into spikes, which propagate to the output through the neurons' synapses. The most common encoding scheme is the rate encoding [4], and the neurons integrate the incoming spikes to increase their membrane potential. Every time the potential overcomes a certain threshold, an output spike is emitted.

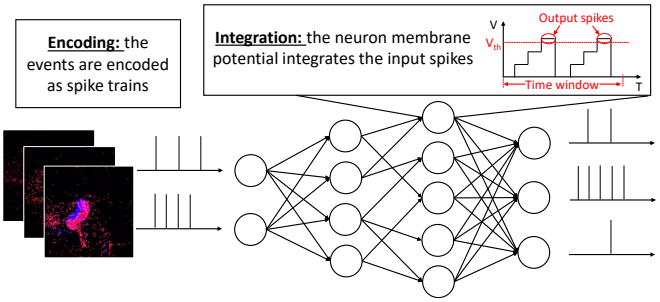

Fig. 3: Overview of an SNN's functionality, focusing on the information encoding into spike trains and the integration of spikes into the membrane potential.

B. Noise Filters for Dynamic Vision Sensors

Event-based cameras [9] are bio-inspired sensors for the acquisition of visual information, directly related to the light variations in the scene. The DVS cameras work asynchronously, not recording frames with a precise timing. Instead, the sensors record negative and positive brightness

[1] https://github.com/albertomarchisio/R-SNN

variations in the scene. Thus, each pixel encodes a brightness change in the scene. Pixels are independent, and can record both positive and negative light variations. Compared to classical frame-based image sensors, the event-based sensors consume significantly less power, since the data is recorded only when a brightness variation is detected in the scene. This means that, in the absence of light changes, no information is recorded, leading close to zero power consumption. Hence, DVS sensors can be efficiently deployed at the edge and directly coupled to neuromorphic hardware for SNN-based applications.

DVS sensors are mainly affected by background activity noise, caused by thermal noise and junction leakage current [21]. When the DVS is stimulated, a neighborhood of pixels is usually active at the same time, generating events. Therefore, the real events show a higher spatio-temporal correlation than the noise-related events. This empirical observation is exploited for filtering out the noise [17]. The events are associated with a spatio-temporal neighborhood, within which the correlation between them is calculated. If the correlation is lower than a certain threshold, the events are likely due to noise and thus are filtered out; otherwise they are kept. The procedure is reported in Algorithm 1, where s and t are the only parameters of the filter and are used to set the dimensions of the spatio-temporal neighborhood. The larger s and t are, the lower the number of events are filtered out. As shown in the example of Fig. 4, the decision of the filter is made by the comparison between $t_e - M[x_e][y_e]$ and t (lines 15-16 of Algorithm 1). If the first term is lower, then the event is filtered out.

Algorithm 1 : Noise filter in the spatio-temporal domain.

1: Being E a list of events of the form (x, y, p, t)
2: Being (x_e, y_e, p_e, t_e) the x-coordinate, the y-coordinate, the polarity and the timestamp of the event e respectively
3: Being M a 128×128 matrix
4: Being S and T the spatial and temporal filter's parameters
5: Initialize M to zero
6: Order E from the oldest to the newest event
7: **for** e in E **do**
8: **for** i in $(x_e - S, x_e + S)$ **do**
9: **for** j in $(y_e - S, y_e + S)$ **do**
10: **if** not $(i == x_e$ and $j == y_e)$ **then**
11: $M[i][j] = t_e$
12: **end if**
13: **end for**
14: **end for**
15: **if** $t_e - M[x_e][y_e] > T$ **then**
16: Remove e from E
17: **end if**
18: **end for**

C. Adversarial Attacks in the Spatio-Temporal Domain

Currently, adversarial attacks are deployed on a wide range of deep learning applications. They represent a serious threat for safety-critical applications, like surveillance, medicine, and autonomous driving [22][23]. The objective of a successful attack is to generate small perturbations to fool the network. Recently, adversarial attacks for SNNs have been explored. Bagheri et al. [10] and Marchisio et al. [11] analyzed adversarial attacks for SNNs in white-box and black-box settings, respectively. Sharmin et al. [12] proposed a methodology to perform the adversarial attack on (non-spiking) DNNs, and then the DNN-to-SNN conversion made the adversarial examples craft the SNNs. Liang et al. [13] proposed a gradient-based adversarial attack methodology for SNNs. Venceslai et al. [24] proposed a methodology to attack SNNs through bit-flips triggered by adversarial perturbations. *However, none of these previous works analyze the attacks on frames of events, coming from DVS cameras.*

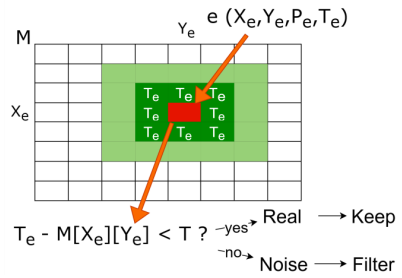

Fig. 4: Functionality of the noise filter for frames of events.

For the adversarial attacks on images, the perturbations are introduced in the spatial domain only. However, when considering adversarial attacks on videos, which are sequences of frames, the attack algorithm is able to perturb in the temporal domain as well. While it is expected that the perturbations added to one frame propagate to other frames through temporal interaction, only perturbing a sparse subset of frames makes the attack stealthy. Indeed, state-of-the-art attacks on videos only add perturbations to a few frames, which are then propagated to other frames to misclassify the video [25]. A simplified example, showing that a mask is generated in front of the frames for deciding which frames are perturbed and which not, is reported in Figure 5.

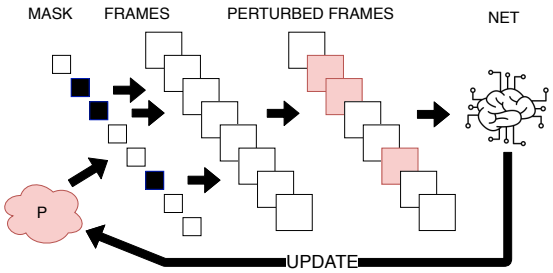

Fig. 5: Overview of the attack scheme for videos [25].

III. R-SNN Methodology

A. Adversary Threat Models

In our experiments, we assume different threat models in the system setting, which are shown in Fig. 6. In all three scenarios, the given adversarial attack algorithm perturbs the frames of events generated from the DVS camera, with the aim of fooling the SNN. In the threat model Ⓐ, the attacker has access to the frames of events at the input of the SNN. In the threat model Ⓑ, the DVS noise filter is inserted in the

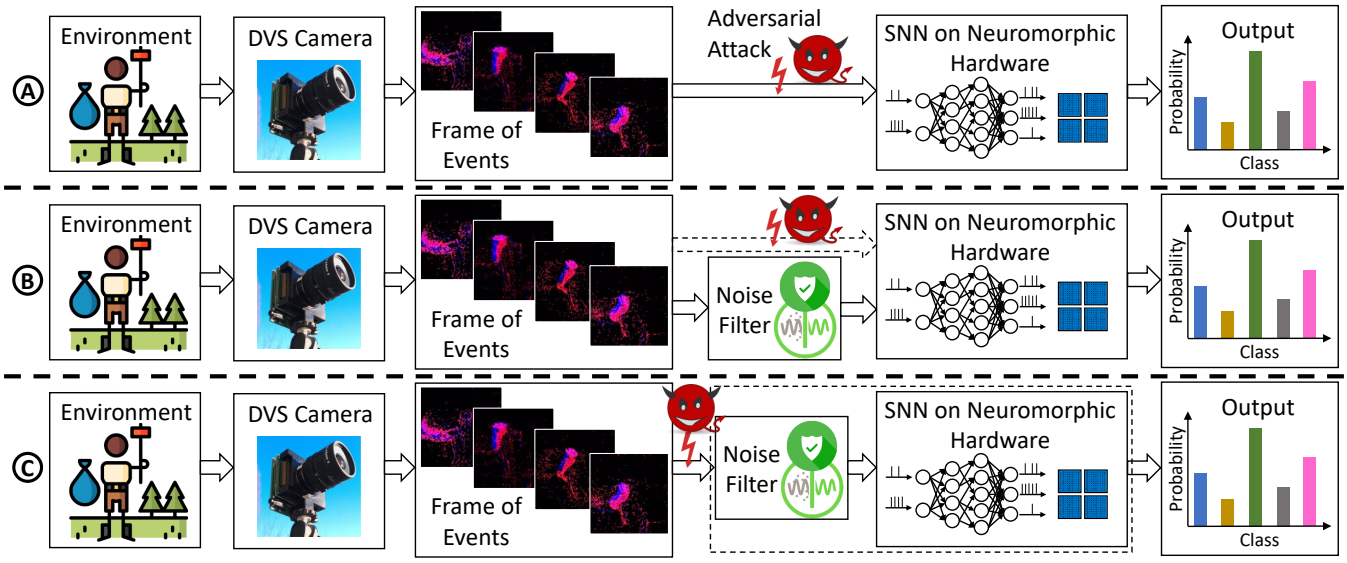

Fig. 6: Adversarial threat models considered in this work. (a) The adversary introduces adversarial perturbations to the frames of events which are at the input of the SNN. (b) The noise filter is inserted as a defense to secure the SNNs against adversarial perturbations, while the adversary is unaware of the filter. (c) The adversary is aware of the presence of the noise filter, and sees it as a preprocessing step of the SNN.

system in parallel to the adversarial perturbation conducted by the attacker. It means that the attacker is unaware of the filter. Since under this assumptions the attack could be relatively weak, we analyze also the threat model Ⓒ, in which the attacker is aware of the presence of the DVS noise filter. In such a scenario, the filter is seen as a preprocessing step of the SNN, and therefore is embedded in the attack loop.

B. Adversarial Attack Generation for Frames of Events

The generation procedure for the adversarial attack for frames of events works as follows. Inspired by the algorithms of attacks for frame-based videos discussed in Section II-C, we devise the specialized algorithm for the DVS signal. Algorithm 2 describes the step-by-step procedure of our methodology. It is an iterative algorithm, which progressively updates the perturbation values based on the loss function (lines 6-12), for each frame series of the dataset D. A mask M determines in which subset of frames of events the perturbation should be added (line 7). Then, the output probability and the respective loss, obtained in the presence of the perturbation, are respectively computed in lines 9 and 10. Finally, the perturbation values are updated based on the gradients of the inputs with respect to the loss.

C. Our Proposed Defense Methodology

Our methodology for defending SNNs is based on specialized DVS-noise filtering. The details for selecting efficient values of the spatial parameter s and temporal parameter t of the filter are reported in Algorithm 3. For different threat models, it automatically searches for the best combination of s and t, by applying the attack in the presence of the filter with the given parameters. The accuracy of the SNN in such conditions is compared to the previously-recorded highest accuracy (line 14 of Algorithm 3). At the output, the parameters s' and t' which provide the highest accuracy are found.

Algorithm 2 : The SNN Adversarial Attack Methodology.

1: Being M a mask able to select only certain frames
2: Being D a dataset composed of DVS images
3: Being P a perturbation to be added to the images
4: Being $prob$ the output probability of a certain class
5: **for** d in D **do**
6: **for** i in $max\_iteration$ **do**
7: Add P to d only on the frames selected by M
8: Calculate the prevision on the perturbed input
9: Extract $prob$ for the actual class of d
10: Update the loss value as $loss = -log(1 - prob)$
11: Calculate the gradients and update P
12: **end for**
13: **end for**

IV. Evaluation of the R-SNN Methodology

A. Experimental Setup

In our experiments, we used two event-based dataset, the DVS-Gesture [18] and the NMNIST [26]. The former is a collection of of 1077 samples for training and 264 for testing, divided into 11 classes, while the latter is a spiking version of the original frame-based MNIST dataset [27]. It contains 60,000 training and 10,000 testing samples generated by an ATIS event-based sensor [28] that is moved while capturing the MNIST images projected on a LCD screen. For the DVS-Gesture dataset, we considered the 4-layer SNN as described in [19], with two convolutional layers and two fully-connected layers. It has been trained it for 625 epochs with the SLAYER backpropagation method [19], using a batch size of 4 and learning rate equal to 0.01. For the NMNIST dataset, we employed a spiking multilayer perceptron with two fully-connected layers [19], trained for 350 epochs with the SLAYER backpropagation method [19],

Algorithm 3 : The SNN Defense Methodology.

Being M the collection of adversarial threat models
Being A the adversarial attack
Being $F(s,t)$ a DVS noise filter with spatial parameter s and temporal parameter t
Being $\mathcal{S}$ the set of possible values of s
Being $\mathcal{T}$ the set of possible values of t
Being $N(F)$ the SNN that we want to robustify with F
for m in M **do**
  Set the relative positions of A and F, based on m
  $Acc' = 0$
  $s' = 0$
  $t' = 0$
  for s in $\mathcal{S}$ **do**
    for t in $\mathcal{T}$ **do**
      if Accuracy$(N(F(s,t))) \geq Acc'$ **then**
        Acc' =Accuracy$(N(F(s,t)))$
        $s' = s$
        $t' = t$
      end if
    end for
  end for
  Output: Values s' and t' for a robust defense in m
end for

using a batch size of 4 and learning rate equal to 0.01. We implemented the SNNs on a ML-workstation with two Nvidia GeForce RTX 2080 Ti GPUs, using the PyTorch framework [29]. We also implemented the adversarial attack algorithm and the noise filter of [17] in PyTorch. The experimental setup and tool-flow in a real-world setting is shown in Fig. 7.

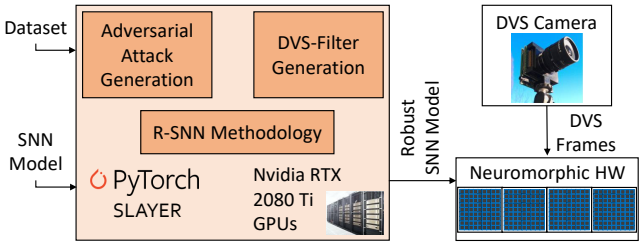

Fig. 7: Experimental setup, tool-flow, and integration with the system.

B. SNN Robustness under Attack Without the Noise Filter

For the threat model Ⓐ, the attacker introduces the adversarial perturbations directly to the input of the SNN. In this case, the SNN for the DVS-Gesture dataset is not protected by the filter and the accuracy dropped to 15.15% (see pointer ① in Fig. 8a). A similar behavior is noted on the SNN for the NMNIST dataset, where the attack reduces the accuracy to 4% (91% reduction, as highlighted by pointer ⑥ in Fig. 8b). We noticed that for both datasets the largest accuracy drop is obtained already after the first iteration of the attack algorithm. Further iterations of the algorithm do not appear to reduce the accuracy to a greater extent.

C. SNN Robustness under Attack by Noise Filter-Unaware Adversary

Afterward, we analyzed the SNN robustness for the threat model Ⓑ, that is the case in which the attacker is able to introduce a perturbation on the input, but is not aware of the presence of the DVS filter. For this experiment set, the accuracy was much higher than for the threat model Ⓐ, proving the effectiveness of the filter as a defense method, for guaranteeing a high robustness of the SNN. The results obtained with our proposed *R-SNN* methodology, varying both the parameters s and t of the filters, are reported in Fig. 8. On the SNN for the DVS-Gesture dataset, for a wide variety of values of s and t (see pointer ③), the accuracy does not change much, settling around 90%, while with $t = 500$ it dropped to 48% (see pointer ⑤). However, when $t = 1$ the influence of s is more evident (see pointer ②). In fact, the accuracy scales from 62.5% when $s = 1$ to 83% when $s = 4$. In all the other cases, the difference is almost not noticeable. Notice, though, that the higher s is, the slower the filter is to process all the data. Among the considered values, $t = 10$ produced the highest accuracy for every s, peaking at 91.67% with $s = 3$ and $s = 4$ (see pointer ④). On the SNN for the NMNIST dataset, a similar behavior is shown. For $t = 1$, the accuracy strongly depends on s (see pointer ⑦). The peak of 94% accuracy is reached for $(s,t) = (3,2)$ and $(s,t) = (4,2)$ (see pointer ⑧). Note that, this is only 1% lower than the original accuracy, i.e., with clean inputs. On the other hand, the accuracy drops below 90% for $t \geq 20$ (see pointer ⑨).

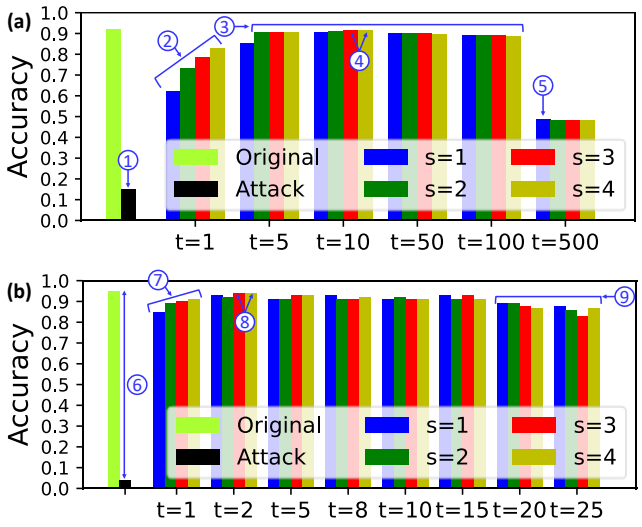

Fig. 8: SNN robustness under the adversarial threat model A, and under the threat model B with different parameters s and t of the filter. (a) Results for the DVS-Gesture dataset. (b) Results for the NMNIST dataset.

D. SNN Robustness under Attack by Noise Filter-Aware Adversary

We also evaluated the *R-SNN* methodology on the threat model Ⓒ, in which the attacker is aware of the presence of the filter. This time the filter was seen as an integral part of the SNN, more specifically as a preprocessing stage. As expected, also in this scenario the filter is effective as a defense mechanism. The differences w.r.t. the threat model Ⓑ are not noticeable. Among the experiments for the DVS-Gesture dataset, the highest robustness is reached

for $(s,t) = (3,10)$ and $(s,t) = (4,10)$, where the SNN exhibits an accuracy of 91.67% (see pointer ① in Fig. 9a). For the NMNIST dataset, the highest robustness, i.e., with an accuracy of 94%, is measured for $(s,t) = (3,2)$ and $(s,t) = (4,2)$ (see pointer ② in Fig.9b). Such a result is a clear sign that this kind of attack is not able to overcome the presence of the filter. Therefore, the attack algorithm is not able to effectively learn the filter's functionality through a gradient-based approach, even though being aware of it.

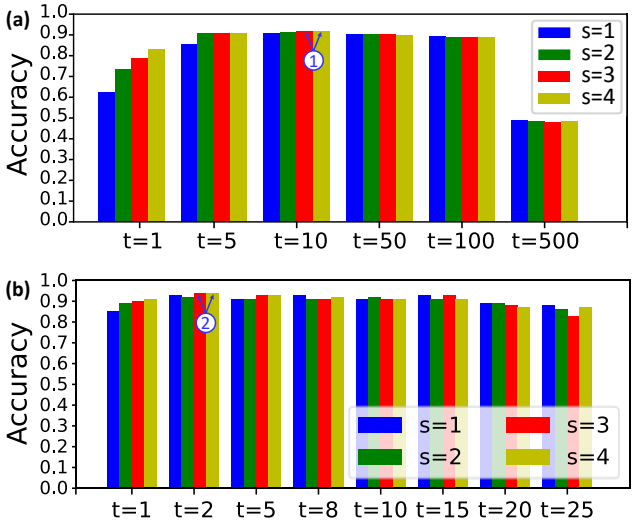

Fig. 9: SNN robustness under adversarial threat model C. (a) Results for the DVS-Gesture dataset. (b) Results for the NMNIST dataset.

E. Case Study: Output Probability Variation

To investigate more in details the effect of the adversarial attack and the filter, we show a comprehensive case study on a test DVS-gesture sample labeled as *left hand wave*. Fig. 10 reports the frames of events and output probabilities for each adversarial threat model presented in this paper, as well as for the clean inputs and the filtered event series without attack. For the clean images the SNN correctly classifies the events as the class 2, which corresponds to *left hand wave* (see Fig. 10-a). By filtering the input signal with $s = 2$ and $t = 5$, as shown in Fig. 10-b, the frames of events are visibly different than the previous case. However, the changes in the output probabilities is minimal, and therefore the SNN correctly classifies the input. When the attack is applied, the output probability of the class 0, which corresponds to *hand clap*, overcomes the correct class. Note that, despite a great difference in the output probabilities, the modifications of the frames of events, compared to the clean event series, are barely noticeable (see Fig. 10-c). However, in the presence of the filter under the adversarial threat models Ⓑ and Ⓒ, the SNN correctly classifies the input. The high gap in the probabilities between the correct class and the other classes in Figures 10-d and 10-e is an indicator for the high robustness of our defense method.

V. Conclusion

In this paper, we presented *R-SNN*, a defense methodology for Spiking Neural Network (SNN) based systems using the event-based Dynamic Vision Sensors (DVS). The proposed gradient-based adversarial attack algorithm exploits the spatio-temporal information residing in the DVS signal, and mislead the SNN, while generating small imperceptible differences w.r.t. the clean series of events. The *R-SNN* defense is based on specialized DVS-noise filters, and an automatic selection of the filter parameters lead to high SNN robustness against adversarial attacks, under different threat models and different datasets. These findings consolidate the positioning of SNNs as robust and energy-efficient solutions, and might enable more advanced secure SNN designs. We release the source code of the *R-SNN* methodology at `https://github.com/albertomarchisio/R-SNN`.

Acknowledgments

This work has been partially supported by the Doctoral College Resilient Embedded Systems, which is run jointly by the TU Wien's Faculty of Informatics and the UAS Technikum Wien. This work was also jointly supported by the NYUAD Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute Award CG001 and by the Swiss Re Institute under the Quantum Cities™ initiative, and Center for CyberSecurity (CCS), funded by Tamkeen under the NYUAD Research Institute Award G1104.

References

[1] S. Zhou *et al.*, "Deep scnn-based real-time object detection for self-driving vehicles using lidar temporal data," *IEEE Access*, 2020.

[2] C. D. V. Gonzalez, J. H. S. Azuela, J. Antelis, and L. E. Falcón, "Spiking neural networks applied to the classification of motor tasks in eeg signals," *Neural networks*, 2020.

[3] G. Tang and K. P. Michmizos, "Gridbot: An autonomous robot controlled by a spiking neural network mimicking the brain's navigational system," *ArXiv*, vol. abs/1807.02155, 2018.

[4] F. Ponulak and A. Kasiński, "Introduction to spiking neural networks: Information processing, learning and applications," *Acta neurobiologiae experimentalis*, 2011.

[5] M. Capra *et al.*, "Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead," *IEEE Access*, 2020.

[6] L. Deng *et al.*, "Rethinking the performance comparison between snns and anns," *Neural networks*, 2020.

[7] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, 2014.

[8] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, 2018.

[9] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change," in *ISSCC*, 2006.

[10] A. Bagheri, O. Simeone, and B. Rajendran, "Adversarial training for probabilistic spiking neural networks," in *SPAWC*, 2018.

[11] A. Marchisio *et al.*, "Is spiking secure? a comparative study on the security vulnerabilities of spiking and deep neural networks," in *IJCNN*, 2020.

[12] S. Sharmin *et al.*, "A comprehensive analysis on adversarial robustness of spiking neural networks," in *IJCNN*, 2019.

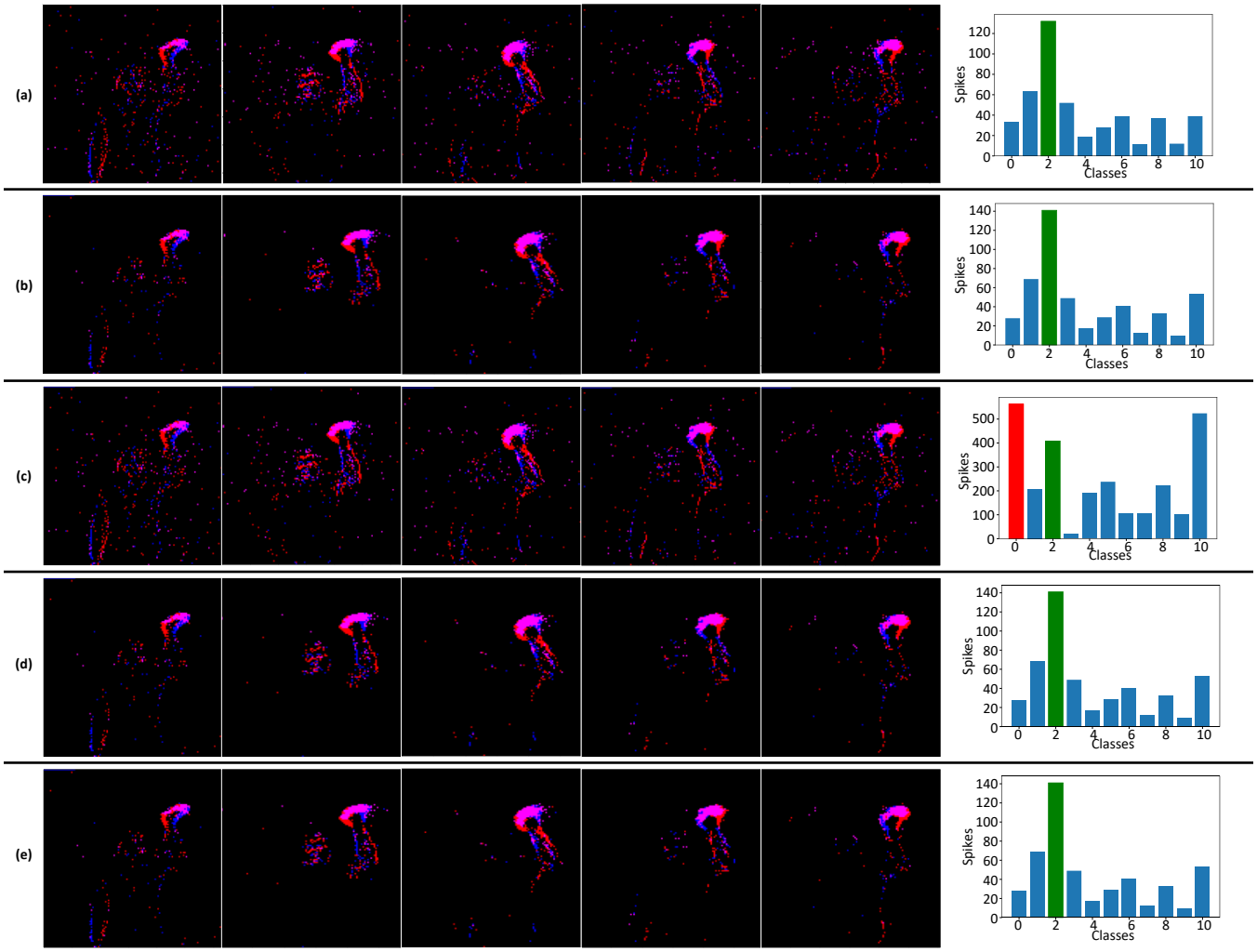

Fig. 10: Detailed example of a sequence of event labeled as *left hand wave*. On the left, the frames of events are shown. The histograms on the right-most column report the number of spikes emitted by the neurons of the last layer, which correspond to the output classes. (a) Clean event series. (b) Event series filtered with $s = 2$ and $t = 5$. (c) Event series under the adversarial threat model A, unfiltered. (d) Event series under the adversarial threat model B, filtered with $s = 2$ and $t = 5$. (e) Event series under the adversarial threat model C, filtered with $s = 2$ and $t = 5$.

[13] L. Liang *et al.*, "Exploring adversarial attack in spiking neural networks with spike-compatible gradient," *ArXiv*, vol. abs/2001.01587, 2020.

[14] A. Madry *et al.*, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[15] H. Zhang *et al.*, "Theoretically principled trade-off between robustness and accuracy," in *ICML*, 2019.

[16] F. Khalid *et al.*, "Fademl: Understanding the impact of pre-processing noise filtering on adversarial machine learning," in *DATE*, 2019.

[17] A. Linares-Barranco *et al.*, "Low latency event-based filtering and feature extraction for dynamic vision sensors in real-time fpga applications," *IEEE Access*, 2019.

[18] A. Amir *et al.*, "A low power, fully event-based gesture recognition system," in *CVPR*, 2017.

[19] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *NeurIPS*, 2018.

[20] W. Maas, "Networks of spiking neurons: The third generation of neural network models," *Trans. Soc. Comput. Simul. Int.*, 1997.

[21] Y. Nozaki and T. Delbruck, "Temperature and parasitic photocurrent effects in dynamic vision sensors," *IEEE Transactions on Electron Devices*, 2017.

[22] C.-H. Cheng *et al.*, "Neural networks for safety-critical applications — challenges, experiments and perspectives," in *DATE*, 2018.

[23] M. Shafique *et al.*, "Robust machine learning systems: Challenges,current trends, perspectives, and the road ahead," *IEEE Design & Test*, 2020.

[24] V. Venceslai *et al.*, "Neuroattack: Undermining spiking neural networks security through externally triggered bit-flips," in *IJCNN*, 2020.

[25] X. Wei, J. Zhu, and H. Su, "Sparse adversarial perturbations for videos," in *AAAI*, 2019.

[26] G. Orchard, A. Jayawant, G. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in Neuroscience*, 2015.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, 1998.

[28] C. Posch, D. Matolin, and R. Wohlgenannt, "A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds," *IEEE JSSC*, 2011.

[29] A. Paszke *et al.*, "Automatic differentiation in pytorch," in *NIPS 2017 Workshop on Autodiff*, 2017.