Noise-Robust Deep Spiking Neural Networks with Temporal Information

Seongsik Park\textsuperscript{1,2}, Dongjin Lee\textsuperscript{1}, Sungroh Yoon\textsuperscript{1,2,3}\textsuperscript{*}

\textsuperscript{1}Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
\textsuperscript{2}Institute of New Media and Communications, Seoul National University, Seoul 08826, South Korea
\textsuperscript{3}ASRI and Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul 08826, South Korea

Abstract—Spiking neural networks (SNNs) have emerged as energy-efficient neural networks with temporal information. SNNs have shown a superior efficiency on neuromorphic devices, but the devices are susceptible to noise, which hinders them from being applied in real-world applications. Several studies have increased noise robustness, but most of them considered neither deep SNNs nor temporal information. In this paper, we investigate the effect of noise on deep SNNs with various neural coding methods and present a noise-robust deep SNN with temporal information. With the proposed methods, we have achieved a deep SNN that is efficient and robust to spike deletion and jitter.

Index Terms—Neural coding, neuromorphic computing, noise robustness, spiking neural network (SNN), temporal coding

I. INTRODUCTION

Deep learning with deep neural networks (DNNs) has shown remarkable results in various fields. However, the energy consumption of DNNs has hindered the broad application of deep learning as DNNs deal with more complex tasks. Spiking neural networks (SNNs) have emerged to address the energy consumption issues of deep learning [1]–[3]. SNNs have event-driven computing characteristics with binary spikes, which lead to energy-efficient processing.

Their efficiency and performance are significantly affected by neural coding, which defines how to represent and transfer information between neurons. There are mainly two types of neural coding, which are rate and temporal coding [4]. For the efficient processing in deep SNNs, temporal coding, including phase [5], burst [1], and time-to-first-spike (TTFS) coding [3], has been actively studied.

SNNs have shown higher energy efficiency on neuromorphic architectures [6], [7]. To improve the efficiency further, lots of studies have been conducted about neuromorphic devices [6], [8], [9]. These devices have been expected to increase the efficiency significantly and lead to a new era of hardware for artificial intelligence. Despite their promising prospect, there are many obstacles to be resolved to utilize the devices in real-world applications. One of the most urgent issues is vulnerability to noise in various aspects, such as noisy synaptic weights, unstable integration, and noise in spike trains [10], [11]. Unlike CMOS-based neuromorphic architectures with digital signals, the emerging neuromorphic devices usually process the operations in SNNs with analog signals, such as current and voltage. Thus, it is essentially required to ensure noise robustness of the neuromorphic devices.

A lot of research has improved the robustness of SNNs to several types of noise [12]–[16]. They have introduced various methods to increase the noise robustness, but most of them focused on SNNs with rate coding. The effect of noise and their approaches have not been evaluated in the case with other neural coding methods, such as temporal coding, which can improve the efficiency of SNNs. Furthermore, their methods required to train SNNs with target noise to achieve the noise-robust model. Even if they showed the effectiveness of these approaches in shallow networks, it has not been verified that their methods can be successfully applied to deep SNNs that are difficult to train.

In this work, to overcome the aforementioned limitations, we analyzed the effect of noise on various neural coding schemes with spike deletion and jitter. With this investigation, we found that information loss due to the spike deletion had a critical impact on the performance of deep SNNs. In addition, we revealed that all-or-none activation of TTFS coding was beneficial for deletion-robust deep SNNs that were configured through DNN-to-SNN conversion approaches. For the jitter noise, we confirmed that TTFS coding was vulnerable to spike shift.

Based on the analysis, we propose noise-robust deep SNNs with temporal coding to achieve both efficiency and robustness. The proposed methods consist of weight scaling and time-to-average-spike (TTAS) coding. The weight scaling compensates the information loss effectively, which is caused by the deletion noise, but it is less efficient with all-or-none activation characteristic in TTFS. To improve the effectiveness of weight scaling, we propose TTAS coding, which utilizes both precise and average spike time. We are inspired by the\textit{phasic bursting} spike pattern [17] for the TTAS coding. To implement a neuron model that generates the phasic burst spikes, we introduce a simplified integrated-and-fire-or-burst neuron model. With the TTAS coding, we can improve the effectiveness of weight scaling and cancel out the jitter noise in spike time. The proposed methods improve the noise robustness without additional training procedures of SNNs, which is a suitable approach to DNN-to-SNN conversion methods.

*corresponding author: Sungroh Yoon (sryoon@snu.ac.kr)
contributions of this paper are summarized as follows:

- **In-depth analysis of spike noise on deep SNNs:** We analyze the impact of spike noise on the performance of deep SNNs with various neural coding methods, including rate, phase, burst, and TTFS coding.

- **Noise-robust deep SNN:** We propose a noise-robust deep SNN, which consists of weight scaling and TTAS coding to exploit temporal information under synaptic noise, including spike jitter and deletion.

## II. BACKGROUND AND RELATED WORK

### A. Spiking Neural Networks

SNNs, which are considered the third-generation artificial neural networks [18], consist of spiking neurons and synaptic weights. Integrate-and-fire (IF) neurons, which is a widely used type of spiking neurons, integrate inputs into the internal state \( u \), which is called membrane potential, as follows:

\[
d\frac{u_i(t)}{dt} = \sum_j w_{ij} z_j(t) + n_i(t) + b_i,
\]

where \( w \) is a synaptic weight, \( z \) is the post-synaptic current (PSC) that is induced by input spike, \( \eta \) is a reset function, \( b \) is a bias, \( i \) \((j)\) and \( l \) are the indices of pre-synaptic (post-synaptic) neuron and layer, respectively [19], [20]. The PSC \( z(t) \) is formulated by \((\epsilon + S_{in})(t)\), where \( S_{in} \) is an input spike train and \( \epsilon \) is a spike response kernel. The reset function \( \eta(t) \) is described as \((\mu + S_{out})(t)\), where \( S_{out} \) and \( \mu \) are output spike train and reset kernel, respectively.

The spike train \( S \), which contains binary spikes, is stated as

\[
S_i(t) = \sum_{t_i,f \in F_i} \delta(t - t_i.f),
\]

where \( \delta \) is the Dirac delta function and \( f \) is the index of spike in a set of spikes \( F \) satisfying the firing condition as follows:

\[
t_i,f : u_i(t_i.f) \geq \theta_i(t_i.f),
\]

where \( \theta \) is a threshold. Due to the features of integrate-and-fire and event-driven computing with binary spike trains (Eqs. [1] and [2]), SNNs have the potential for energy-efficient processing.

SNNs transmit information between neurons with binary spike trains. Hence, the efficiency and performance of SNNs vary depending on neural coding methods, which define how to represent the information in the form of a spike train. The neural coding schemes are mainly divided into three main types of coding: rate coding [4], phase coding [4], and burst coding [4]. As shown in Fig. 1A. Rate coding utilizes spike firing rate \( f = N/T \) to represent the information in a given time window \( T \) with the number of spikes \( N \) [2], [21], [22]. This approach is known to be less efficient because it does not exploit temporal information in spike trains [4]. An empirical threshold balancing improved the efficiency, but it did not overcome the drawback of rate coding [22].

Temporal coding is represented by phase [5], burst [1], and TTFS coding [3], which utilize global oscillator, inter-spike interval (ISI), and spike time, respectively. Among them, TTFS coding uses the least number of spikes, which results in the highest computational efficiency. The TTFS coding had a latency issue, but it was resolved in [3]. Thus, this neural coding is promising for efficient deep SNNs.

### B. Noise on Spiking Neural Networks

The types of noise associated with neuromorphic hardware can be classified mainly into external and internal noise. External noise, or input noise, originates from corruption or perturbation of the input data [23]. The reason for this kind of noise is not directly related to neuromorphic hardware itself. However, managing the input noise cannot be underestimated considering the potential usage of neuromorphic hardware with noisy data in the real world. Several works have studied the effect of this kind of noise on SNNs [12]–[16].

The internal noise can be further classified as follows: static and dynamic noise. The static noise (termed ‘fixed-pattern noise’ in [24]) is caused by manufacturing variations of neuromorphic hardware. This mismatch leads to parametric errors invariant over time. The dynamic noise (termed ‘temporal variability’ in [24]) is considered more arduous to handle. This noise can cause time-varying deviations in neuromorphic devices since it is mainly caused by thermal variations or instabilities of analog circuits. Thus, unlike static noise, circuits with dynamic noise can produce different outcomes in multiple trials even when the input remains unchanged.

In order to simulate the aforementioned noisy nature of neuromorphic hardware, SNNs can be modeled with the noise on various parameters, such as synaptic weights, thresholds, and time constants [25]–[27]. In another aspect, one can model neurons to produce noisy output spikes, which typically appear in the form of spike jitter and deletion [28]–[30]. Since the parametric errors in the former approach result in noisy spikes, we adopted the latter method for modeling the noise.

After implementing SNNs on neuromorphic hardware, errors resulting from the static noise can be mitigated by parameter correction via on-chip learning [24]. However, errors due to the dynamic noise cannot be resolved in this way. Thus, to address the noise issues, we need to design SNNs to be robust to the noise before deployment to the neuromorphic hardware.
Fig. 2: Inference accuracy and the number of spikes with spike deletion on VGG16 and CIFAR-10 dataset depending on various neural coding methods

III. ANALYSIS OF NOISE ON DEEP SNNs

In this study, we analyzed the effect of the spike noises, which are spike deletion and jitter, on deep SNNs with various neural coding methods. Based on this investigation, we propose a noise-robust deep SNN with temporal information to achieve both high efficiency and noise robustness. To utilize various neural coding approaches in deep SNNs, we adopted DNN-to-SNN conversion methods as in [1, 3, 5, 21, 22]. We implemented the spike deletion with a deletion probability p and uniformly distributed random variable between zero and one. To implement the spike jitter, we used Gaussian noise with zero mean and standard deviation σ. The jitter noise was determined by σ and quantized into an integer to add it to spike time \( t_f \). We swept the deletion probability \( p \) from 0.1 to 0.9 and the standard deviation \( \sigma \) from 0.5 to 4.0. To evaluate the effect of noise on deep SNNs, we experimented with the CIFAR-10 dataset on VGG16.

The results of deletion noise are depicted in Fig. 2. As the probability of deletion increased, the accuracy and the number of spikes decreased in all cases. When the deletion probability \( p \) was greater than 0.4, the accuracy was less than 40%. Due to information loss that was caused by the spike deletion, the overall accuracy deteriorated. With the deletion probability \( p \), the sum of PSC \( Z' \) during time window \( T \), which corresponds to activation \( A \) in DNNs, is reduced to \( (1-p)Z \) on average, where \( Z \) is the sum of PSC without deletion noise.

Among the neural coding methods, TTFS was the most robust to the deletion noise. The reasons lay in information transmission methods of neural coding and the DNN-to-SNN conversion approach. Even if the expected activation values of various neural coding methods with a deletion probability are the same with \( (1-p)A \), where \( A \) is the activation without noise, the effect of the noise on the activation appeared differently. In rate, phase, and burst coding, a number of spikes carry an activation \( A \), which is the sum of PSC. When the number of spikes is \( N \), the effective PSC of a spike is \( A/N \). Hence, each activation with the deletion noise \( A' \) lies in a range of \( [0, A] \). On the other hand, in TTFS coding, one spike carries the activation \( A \); thus, the deletion-noisy activation \( A' \) is 0 or \( A \).

These differences in the noisy activation cause the different noise robustness on various neural coding with the DNN-to-SNN conversion method. In the conversion approach, which is widely used for training deep SNNs indirectly, synaptic weights of SNNs are converted from pre-trained DNNs; thus, the weights have characteristics resulting from the training of DNNs. Dropout [31], which is used for avoiding overfitting and increasing generalization performance of DNNs, affects the noise robustness of each neural coding. It randomly makes activations zero during training DNNs, which is similar to the deletion noise in TTFS. This all-or-none property of activation strengthened the robustness to deletion noise of TTFS coding, as shown in Fig. 2.

The experimental results of spike jitter are depicted in Fig. 3. Rate coding was hardly affected by the spike jitter because it does not utilize temporal information in spike trains. The jitter noise significantly affected the results of temporal coding methods. Especially, TTFS was the most susceptible to the spike jitter because it uses only one spike per activation. The number of spikes on each neural coding did not vary considerably according to the jitter intensity. Phase and burst coding methods showed a tendency to increase in the number of spikes as jitter intensity increased. In contrast, TTFS presented a much smaller number of spikes, which left room for redundant spikes to cancel out the jitter noise.

IV. NOISE ROBUST DEEP SNNs

In the previous section, we investigated the vulnerability of temporal coding to spike deletion and jitter. Despite the high efficiency of TTFS coding in terms of the number of spikes and inference time, the information loss and spike shifting, which were caused by the spike deletion and jitter, respectively, hindered the use of temporal information in deep SNNs in noisy environments, such as emerging neuromorphic devices. To overcome such obstacles, we propose a noise-robust deep SNN, which consists of weight scaling and TTAS coding.

The weight scaling compensates for the information loss due to the deletion noise. As discussed in the previous section, the amount of activation is reduced to \( A' = (1-p)A \) on average with the deletion probability \( p \). To guarantee sufficient
information with the deletion, we scale the synaptic weight as $W' = CW$, where $C$ is a scale factor, and $W'$ is the scaled weight. As a pioneering study of the weight scaling for the noise robustness, we set $C$ proportional to the deletion probability $p$. The results of weight scaling are shown in Fig. 5. The weight scaling approach increased the noise robustness in all cases.

Although the robustness of deep SNNs was improved by the scaling, it was less effective in TTFS coding. The difference in the effectiveness of the scaling originated from the information transmission methods. In rate, phase, and burst coding, a number of spikes were required to represent an activation, which led to uniformly reduced activation $(1-p)A$ with high probability. These reduced activations were restored with the deterministic weight scaling factor $C$. On the other hand, in TTFS coding, the weight scaling resulted in an activation value of 0 or $CA$ with the probability of $p$ or $(1-p)$, respectively. Thus, it caused over activations, which led to the least improvement in deletion robustness.

To alleviate the over activation and exploit all-or-none activation property for deletion noise, we propose TTAS coding, which utilizes both precise spike time and burst spikes, as shown in Fig. 5-A. This temporal coding is inspired by phasic bursting spike pattern [17]. To implement the phasic burst spikes with negligible computational overhead, we introduce

$$\eta(t) = \begin{cases} 
0 & \text{if } (t < t_1) \\
\theta(t) & \text{if } (t \geq t_1) \text{and} (t < t_1 + t_a) \\
-\infty & \text{otherwise,}
\end{cases}$$

(4)

where $t_1$ and $t_a$ are the first spike time and target duration of burst spikes, which indicates the number of phasic burst spikes, respectively. This neuron model can be considered a simplified integrate-and-fire-or-burst neuron [32] and implemented with counter and gate operations.

TTAS coding utilizes redundant spikes in the first group of spikes to deliver information, which increases the sum of PSC as

$$\hat{Z} = \sum_{t} z(t^1 + t).$$

(5)

For accurate processing, we need to offset this increment. We set another scaling factor for TTAS coding as $C_{\eta} = z(t^1) \hat{Z}$ and integrated the scale factor to synaptic weight in order not to increase the computational overhead.

Rate, phase, and burst coding methods were vulnerable to deletion noise due to their continuously reduced activation, as depicted in Fig. 5-B. On the contrary, TTFS was prone to jitter noise because of the all-or-none activation. Each of them has its advantage; thus, it is crucial to combine the two advantages to achieve efficiency and robustness simultaneously. With the proposed TTAS coding, we can exploit the benefit of the all-or-none activation feature and information compensation by weight scaling. In particular, when TTAS is applied to the exponentially decreasing PSC kernel, as in [3], the activation distribution appears high around 0 and $A$. This distribution makes it possible to exploit both advantages of discontinuous and continuous activations for the deletion and jitter noise.

V. EXPERIMENTAL RESULTS

To evaluate the proposed methods, we experimented with various neural coding methods on MNIST, CIFAR-10, and CIFAR-100 datasets. We empirically obtained the threshold $\theta$ to reduce inference latency and improve the efficiency of SNNs as in [22]. As the threshold search results, we set $\theta$ to 0.4, 0.4,
1.2, and 0.8 for rate, burst, phase, and TTFS, respectively. We used 1,000 time steps for CIFAR-10 and CIFAR-100 datasets and 100 time steps for MNIST dataset. For TTFS, we set the time step to 108 on MNIST and CIFAR-10, and used 216 time steps on CIFAR-100.

The results of weight scaling and TTAS with deletion noise are depicted in Fig. 4. With the proposed TTAS coding, the deep SNN showed robustness to the deletion noise as the target duration of burst spikes \( t_a \) increased, which is represented in parentheses. The results of TTAS on jitter noise are illustrated in Fig. 6. As the target duration increased, TTAS surpassed TTFS in terms of robustness. In both noise cases of spike deletion and jitter, the improvements of TTAS were saturated as the target duration increased.

The comparisons with other neural coding and proposed methods are depicted in Fig. 7 and 8 for deletion and jitter noise, respectively. We found the target duration of burst spikes empirically depending on the dataset and noise type. As shown in Fig. 7 by applying the weight scaling, the robustness against deletion was significantly improved. However, when we compared the improvements between neural coding methods, TTFS showed the least improvement. Our proposed approach, which is TTAS with weight scaling, was the most robust to the deletion noise. For the jitter noise, TTAS could achieve similar robustness to that of burst coding. The detailed experimental results, including the number of spikes, are reported in Tables I and II.

VI. Conclusion

In this paper, we thoroughly analyzed the effect of spike noises, which are spike deletion and jitter, on deep SNNs. Based on the analysis, we proposed a noise-robust deep SNN with weight scaling and TTAS coding. The proposed SNN shows noise robustness while exploiting the efficiency of temporal coding without additional training procedures. We believe that our approach paves the way to efficient and robust neuromorphic computing in the near future.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) [2016M3A7B491115, 2018R1A2B3001628, 2021R1C1C2010454], the Brain Korea 21 Plus Project in 2021, AIRS Company in Hyundai Motor Company & Kia Motors Corporation through HKMC-SNU AI Consortium Fund, and Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT1901-12.

REFERENCES

[1] S. Park, S. Kim, H. Choe, and S. Yoon, “Fast and efficient information transmission with burst spikes in deep spiking neural networks,” in Design Automation Conference (DAC), 2019.
[2] S. Kim, S. Park, B. Na, and S. Yoon, “Spiking-yolo: spiking neural network for energy-efficient object detection,” in AAAI Conference on Artificial Intelligence (AAAI), 2020.
[3] S. Park, S. Kim, B. Na, and S. Yoon, “T2fsnn: deep spiking neural networks with time-to-first-spike coding,” in Design Automation Conference (DAC), 2020.
[4] J. Gautrais and S. Thorpe, “Rate coding versus temporal order coding: a theoretical approach,” Biosystems, vol. 48, no. 1-3, pp. 57–65, 1998.
[5] J. Kim, H. Kim, S. Huh, J. Lee, and K. Choi, “Deep neural networks with weighted spikes,” Neurocomputing, vol. 311, pp. 373–386, 2018.
[6] P. Merolla et al., “A million spiking-neuron integrated circuit with a scalable communication network and interface,” Science, vol. 345, no. 6197, pp. 606–673, 2014.
[7] M. Davies et al., “Loihi: A neuromorphic manycore processor with on-chip learning,” IEEE Micro, vol. 38, no. 1, pp. 82–99, 2018.
[8] K. Roy, A. Jaiswal, and P. Panda, “Towards spike-based machine intelligence with neuromorphic computing,” Nature, vol. 575, no. 7784, pp. 607–617, 2019.
[9] M. Bouvier et al., “Spiking neural networks hardware implementations and challenges: A survey,” ACM Journal on Emerging Technologies in Computing Systems, vol. 15, no. 2, pp. 1–35, 2019.
[10] D. Quertloz, O. Bichler, and C. Gnamr, “Simulation of a memristor-based spiking neural network immune to device variations,” in International Joint Conference on Neural Networks (IJCNN), 2011.
[11] D. Quertloz, O. Bichler, P. Dollfus, and C. Gnamr, “Immunity to device variations in a spiking neural network with memristive nanodevices,” IEEE Transactions on Nanotechnology, vol. 12, no. 3, pp. 288–295, 2013.
[12] X. Cheng, Y. Hao, J. Xu, and B. Xu, “Lisnn: Improving spiking neural networks with lateral interactions for robust object recognition,” in International Joint Conference on Artificial Intelligence (IJCAI), 2020.
[13] Q. Yu, R. Yan, H. Tang, K. C. Tan, and H. Li, “A spiking neural network system for robust sequence recognition,” IEEE transactions on neural networks and learning systems, vol. 27, no. 3, pp. 621–635, 2015.
[14] A. Zhang, H. Zhou, X. Li, and W. Zhu, “Fast and robust learning in spiking feed-forward neural networks based on intrinsic plasticity mechanism,” Neurocomputing, vol. 365, pp. 102–112, 2019.
[15] Y. Zheng, S. Li, R. Yan, H. Tang, and K. C. Tan, “Sparse temporal encoding of visual features for robust object recognition by spiking neurons,” IEEE transactions on neural networks and learning systems, vol. 29, no. 12, pp. 5823–5833, 2018.
| Methods | Clean | 0.2 | 0.5 | 0.8 | Avg. | Clean | 0.2 | 0.5 | 0.8 | Avg. |
|---------|-------|-----|-----|-----|------|-------|-----|-----|-----|------|
| **MNIST** |       |     |     |     |     |       |     |     |     |      |
| Rate + WS | 89.90 | 98.50 | 99.97 | 99.69 | 98.89 | 10.06 | 5.66E4 | 1.95E4 | 4.64E4 | 4.64E4 |
| Phase + WS | 10.06 | 89.50 | 99.97 | 98.89 | 98.93 | 0.49E4 | 5.66E4 | 1.95E4 | 4.64E4 | 4.64E4 |
| Burst + WS | 10.06 | 89.50 | 99.97 | 98.89 | 99.09 | 0.49E4 | 5.66E4 | 1.95E4 | 4.64E4 | 4.64E4 |
| TTFAS + WS | 10.06 | 89.50 | 99.97 | 98.89 | 99.09 | 0.49E4 | 5.66E4 | 1.95E4 | 4.64E4 | 4.64E4 |
| TTAS + WS | 10.06 | 89.50 | 99.97 | 98.89 | 99.09 | 0.49E4 | 5.66E4 | 1.95E4 | 4.64E4 | 4.64E4 |
| **CIFAR-10** |       |     |     |     |     |       |     |     |     |      |
| Rate + WS | 90.58 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| Phase + WS | 90.58 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| Burst + WS | 90.58 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| TTFAS + WS | 90.58 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| TTAS + WS | 90.58 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| **CIFAR-100** |       |     |     |     |     |       |     |     |     |      |
| Rate + WS | 89.50 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| Phase + WS | 89.50 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| Burst + WS | 89.50 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| TTFAS + WS | 89.50 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |
| TTAS + WS | 89.50 | 99.90 | 98.89 | 98.93 | 98.93 | 9.48E4 | 4.94E4 | 2.16E4 | 4.93E4 | 4.93E4 |

TABLE II: Accuracy (%) of spike jitter on deep SNNs with various neural coding methods depending on jitter intensity (σ).

| Methods | Clean | 1.0 | 2.0 | 3.0 | Avg. |
|---------|-------|-----|-----|-----|------|
| **MNIST** |       |     |     |     |      |
| Phase | 99.33 | 99.17 | 93.20 | 56.36 | 82.91 |
| Burst | 99.34 | 99.21 | 95.31 | 58.66 | 84.39 |
| TTFAS | 99.34 | 99.21 | 95.31 | 58.66 | 84.39 |
| TTAS | 99.34 | 99.21 | 95.31 | 58.66 | 84.39 |
| **CIFAR-10** |       |     |     |     |      |
| Phase | 90.58 | 99.21 | 90.61 | 37.62 | 40.62 |
| Burst | 92.21 | 90.61 | 37.62 | 40.62 | 40.62 |
| TTFAS | 92.21 | 90.61 | 37.62 | 40.62 | 40.62 |
| TTAS | 92.21 | 90.61 | 37.62 | 40.62 | 40.62 |
| **CIFAR-100** |       |     |     |     |      |
| Phase | 96.34 | 99.21 | 90.61 | 37.62 | 40.62 |
| Burst | 92.21 | 90.61 | 37.62 | 40.62 | 40.62 |
| TTFAS | 92.21 | 90.61 | 37.62 | 40.62 | 40.62 |
| TTAS | 92.21 | 90.61 | 37.62 | 40.62 | 40.62 |

[16] S. Chowdhury, C. Lee, and K. Roy, “Towards understanding the effect of leak in spiking neural networks,” arXiv preprint arXiv:2006.08761, 2020.
[17] E. M. Izhikevich, “Which model to use for cortical spiking neurons?” IEEE transactions on neural networks, vol. 15, no. 5, pp. 1063–1070, 2004.
[18] W. Maas, “Networks of spiking neurons: the third generation of neural network models,” Neural Networks, vol. 10, no. 9, pp. 1659–1671, 1997.
[19] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge University Press, 2014.
[20] W. Zhang and P. Li, “Temporal spike sequence learning via backpropagation for deep spiking neural networks,” in Conference on Neural Information Processing Systems (NeurIPS), 2020.
[21] B. Rueckauer and S.-C. Liu, “Conversion of analog to spiking neural networks using sparse temporal coding,” in International Symposium on Circuits and Systems (ISCAS), 2018.
[22] B. Han, G. Srinivasan, and K. Roy, “Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network,” in Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[23] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” in International Conference on Learning Representations (ICLR), 2019.
[24] E. Stromatias, D. Neil, M. Pfeiffer, F. Galluppi, S. B. Furber, and S.-C. Liu, “Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms,” Frontiers in neuroscience, vol. 9, p. 222, 2015.
[25] E. O. Neftci, B. U. Pedroni, S. Joshi, M. Al-Shedivat, and G. Cauwenberghs, “Stochastic synapses enable efficient brain-inspired learning machines,” Frontiers in neuroscience, vol. 12, p. 836, 2018.
[26] Q. Yu, Y. Chua, M. Zhang, H. Li, and K. C. Tan, “A spiking neural network framework for robust sound classification,” Frontiers in neuroscience, vol. 12, p. 836, 2018.
[27] Q. Yu, Y. Yao, L. Wang, H. Tang, J. Dang, and K. C. Tan, “Robust environmental sound recognition with sparse key-point encoding and efficient multispike learning,” IEEE Transactions on Neural Networks and Learning Systems, 2020.
[28] S. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.
[29] H. R. Wilson, “Simplified dynamics of human and mammalian cortical neurons,” Journal of theoretical biology, vol. 200, no. 4, pp. 375–388, 1999.