Novel Knowledge Distillation to Improve Training Accuracy of Spin-based SNN

Hanrui Li[1], Aijaz H. Lone[1], Fengshi Tian[2], Jie Yang[2], Mohamad Sawan[2], Nazek El-Atab[1]
[1]*SAMA Labs, CEMSE department, KAUST, Thuwal, Saudi Arabia, 23955-6900*
[2]*CenBRAIN Neurotech, School of Engineering, Westlake University, Hangzhou, China, 310030*
nazek.elatab@kaust.edu.sa

***Abstract*—Spintronics-based magnetic tunnel junction (MTJ) devices have shown the ability working as both synapse and spike threshold neurons, which is perfectly suitable with the hardware implementation of spike neural network (SNN). It has the inherent advantage of high energy efficiency with ultra-low operation voltage due to its small nanometric size and low depinning current densities. However, hardware-based SNNs training always suffer a significant performance loss compared with original neural networks due to variations among devices and information deficiency as the weights map with device synaptic conductance. Knowledge distillation is a model compression and acceleration method that enables transferring the learning knowledge from a large machine learning model to a smaller model with minimal loss in performance. In this paper, we propose a novel training scheme based on spike knowledge distillation which helps improve the training performance of spin-based SNN (SSNN) model via transferring knowledge from large CNN model. We propose novel distillation methodologies and demonstrate the effectiveness of the proposed method with detailed experiments on four datasets. The experimental results indicate that our proposed training scheme consistently improves the performance of SSNN model by a large margin.**

***Index Terms*—SNN, magnetic tunnel junction, knowledge distillation, transfer learning**

I. Introduction

Nowadays, many AI applications require deploying at the edge for real-time assistance and feasibility, where computation and memory sources are comparatively limited. Spike neural network (SNN) is inspired biological brain and nervous systems, which attracts the arising the attention of both academia and industry over several years [1]. As a future candidate for edge computing, the SNNs process and convey information by spike, which significantly reduces information redundancy with low resource utilization and energy-efficient information processing.

Spintronics-based magnetic tunnel junction (MTJ) devices have shown promise application in achieving brain-like SNN-based architectures [2]–[4]. Magnetic skyrmions are particle-like magnetic textures that are widely used in non-volatile data storage and computing applications [5]. MTJ devices based on skyrmions contain the basic element of SNN, including skyrmion neurons [6] and skyrmion synapses [7]. Therefore, the electric-field control of spintronic devices can be a candidate for hardware implementation of SNNs with the benefit of high data storage density [8] and low switching energy [9]. However, the hardware training and off-line mapping process always meet a significant accuracy loss due to the device stochasticity and low precision among device synaptic weights [10].

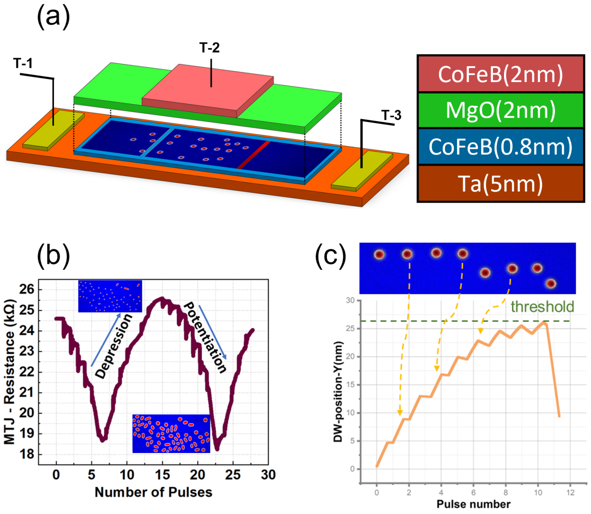

Fig. 1. (a) Skyrmion-MTJ device structure (Ta/CoFeB/MgO/CoFeB), which shows both neuron and synapse behavior. It is a three-terminal device (T-1, T-2, and T-3), which can be easily programmed by tunning the voltage of T-2. (b) MTJ-based synapse STP behaviour (potentiation and depression). (c) MTJ-based LIF neuron model, where skyrmion position changes according to the pulses.

This study utilizes an energy-efficient skyrmion-based neuromorphic MTJ (Ta/CoFeB/MgO/CoFeB) device with different neuron plasticity (potentiation and depression) [11]. It also shows Leaky Integrate and Fire (LIF) neuron behavior, which can be viewed as a biological plausible activation function. Based on that skyrmion-based MTJ device, we propose a novel knowledge distillation training scheme to increase the robustness of spin-based spike neural network (SSNN) system and training accuracy. In this paper, we describe the implementation of knowledge distillation in SSNN, which assist SSNN to learn hybrid information from convolutional neural network (CNN) and overcome insufficient training problem. In the first stage, we pre-train a CNN (teacher network) with raw data to achieve higher performance. Then, in the second stage, we introduce an SSNN (student network) with an initial state fully utilizing spin-based synapses and a spin-based LIF

This work was supported by the King Abdullah University of Science and Technology baseline fund.

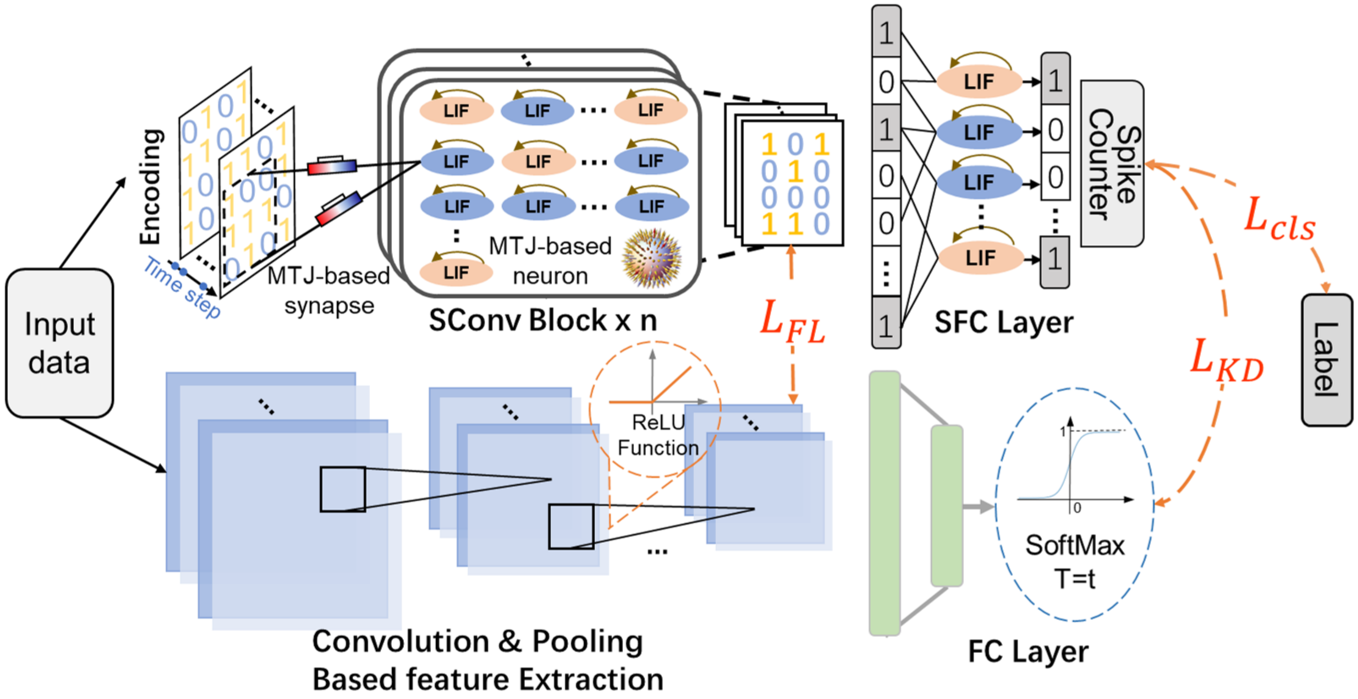

Fig. 2. Conceptual illustration of proposed training scheme, where 'SConv Block' stands for 'spike convolution block', 'SFC Layer' stands for 'spike fully connected layer', 'FC Layer' stands for 'fully connected layer'. The upper model is the spin-based student model, where MTJ-based neuron and synapses are utilized. The blow model is the pretrained teacher model, which guides the student model training. L_{KD} and L_{KD} are measured by the features map output probability distribution over student and teacher model. L_{cls} is calculated through true label and student output.

neuron model, then both the student and teacher networks are optimized under different constraints. Our training method has potent portability and generality, which does not require any alterations to the original SNN architecture.

In this paper, we describe the spike knowledge distillation approach based on SSNN, which observes a noticeable performance increase within four datasets among different time steps. The remaining sections of this paper are organized as follows. Section II introduces the spintronic device model and the algorithm details. Experimental results are shown in Section III. In Section IV we make a brief conclusion to this paper.

II. Theory of the Proposed Approach

A. MTJ-based synapse and neuron

Fig. 1(a) presents skyrmion-MTJ device structure [Ta (5nm)/CoFeB (0.8nm)/MgO (2nm)/CoFeB (2nm)], which relies on skyrmion size and density manipulation. The skyrmion-MTJ device can show mixed synaptic plasticity, which can be easily programmed by tuning the voltage [12]. When positive voltage pulses are applied, the free layer anisotropy decreases by 1%–5%, which causes the increase in skyrmion size along with the resistance decreases (potentiation). On the contrary, when removing the voltage, the skyrmion loosens to its original size brought by depression as shown in Fig. 1(b). The synaptic behavior provides the suitability to use as the weights in SNN.

Besides synapse, the LIF neuron is also an indispensable part of SNN. The LIF helps integrate spike information and transfer messages in a low-power consumption way. The simulation results indicate MTJ-based synaptic device can also work as the LIF neuron dependent on current and skyrmion position. By fitting curve of the micromagnetic simulation of the skyrmion velocity, we model the skyrmion LIF neuron by the following functions:

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = \frac{1}{k_1}(x^2 + k_2 x + \vartheta j) \tag{1}$$

$$\vartheta = -\left(\frac{\beta}{\alpha} + \frac{(\alpha-\beta)}{\alpha^3((\frac{D}{G})^2 + \alpha)}\right)\frac{P\alpha^3}{2eM_s} \tag{2}$$

where x is the position of the skyrmion with the respect to MTJ, also indicates with the membrane of spike neuron, $k_1 = 2.12 \times 10^2$, $k_2 = -27.6$, and $\vartheta = f(\alpha, \beta, G, D, a \;\; and \;\; P)m^3/C$, with α, β are parameters determined by the non-adiabatic term, G is the gyromagnetic coupling, D is the dissipative force tensor, P is the polarization, e is the elementary charge, a is the lattice constant and M_s is the saturation magnetization. The experimental LIF neuron simulation result is illustrated in Fig. 1(c). When receiving the stimulus pulses, the membrane of neuron will update and integrate information. When the membrane potential comes to the threshold voltage, it will generate a spike, and then decay to the rest. The threshold and reset function perfectly meets the ideal LIF neuron property, which indicates its potential

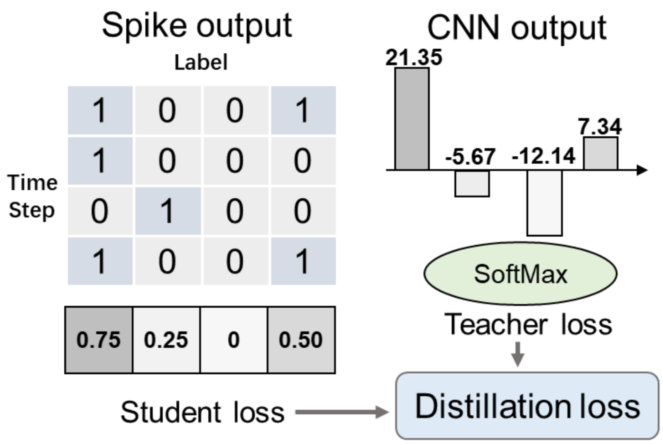

Fig. 3. The comparison of SNN and CNN output

utilization in SNN.

B. Knowledge distillation

Knowledge distillation is a transfer learning method that acts as a similar way to how human beings learn. The same as student learning from teacher, knowledge distillation was proposed to obtain a comparable performance in a small and low precision model by transferring the knowledge from a large teacher model. It is a popular model compression method, which has benefited many machine learning tasks [13]–[15].

As shown in Fig. 3, SNN and CNN have different sorts of output, which provides the potential probability to learn and extract knowledge from output response. SSNN requires to count spikes to compute the average output over all time steps as spike output, while CNN has the output of direct probability distribution after SoftMax function. Besides, with the utilization of the spin-based synaptic weight and neuron model, SSNN has less precision due to device synapse and occurs information deficiency when computing the gradient to optimize the model parameters. With the guidance of teacher model, student model could learn soft targets and directly mimic the final prediction of the teacher model. The response-based knowledge distillation is simple yet effective for model compression, which has been used in SNN training tasks [16].

Besides, feature map from the intermediate layers could be a reasonable extension of soft target based knowledge. Though SSNN and CNN have different data input and activation methods, the intermediate feature map can also guide the training of the student model by providing extracted knowledge.

Algorithm 1: The proposed training scheme

```
 Input: data, time_step,
temperature parameter: $T_0$, learning rate: $l_r$,
batch size: $B_s$, training dataset size: $N_{k+1}$
Spike generator
for t=0 to time_step do
if normalized data>Gaussian random then
  spike_seq(t)← 1;
else
  spike_seq(t)← 0;
end
 Output : Distilled student Model : $f_s$,
# Initialization
Initialize teacher model $f_t$ with pretrained weight $\Theta_t$
Initialize student model $f_s$ with device conductance $\Theta_s$
while $i = 0$; $i < Epoch$; $i$++ do
  # batch loop
  while $b = 0$; $b < [N_{k+1}/B_s]$; $b$++ do
    $\mathcal{L}_t \longleftarrow \{f_s(spike\_seq(t)), f_t(data), Target\}$
    $\Theta_t \longleftarrow \Theta_t + l_r \frac{\partial \mathcal{L}_t}{\partial \Theta_t}$
    $\mathcal{L}_s \longleftarrow \{f_s(spike\_seq(t)), f_t(data),$
    $feature(f_s), feature(f_t), Target, Temperature)\}$
    $\Theta_s \longleftarrow \Theta_s + l_r \frac{\partial \mathcal{L}_s}{\partial \Theta_s}$
  end
end
Return the SSNN model that yields better performance.
```

C. Training framework

Fig. 2 shows our training scheme. In first stage, we pretrain a CNN as the teacher model f_t parameterized Θ_t with original data and normal activation function ReLU to achieve a higher accuracy. In the second stage, with spike encoded in different time steps, we aim to train a SSNN student model f_s with initial device conductance Θ_s. To achieve better performance and transfer knowledge between models, we ask f_t and f_s to optimize at different constraints and guide f_s to "learn" knowledge from both true label and f_t. Specifically, f_t and f_s have the same model structure but f_s shows more biological property and energy efficiency, including using encoded spikes as the input and adopting the MTJ-based LIF neuron model and MTJ-based synaptic as the activation function and the weights respectively. The proposed loss function to train the student model consists of three losses; a classification loss L_{cls}; a knowledge distillation loss L_{KD}, and a feature supporting loss L_{FL}:

$$L_s = L_{cls} + \alpha L_{KD} + \beta L_{FL} \tag{3}$$

α and β represent the weights given to the losses L_{KD}, L_{FL}. When given a sample data x_n, the probabilistic distribution outputs are denoted by $q_n^t = \sigma(f_t(x_n))$ and $q_n^s = count(f_s(x_n))$, where $\sigma(\cdot)$ refers to the softmax function and $count()$ is the function to sum spike information. L_{cls} is known as mean squared error (MSE) loss and L_{KD} acts as knowledge distillation loss [17]. L_{cls} and L_{KD} could be defined below:

$$L_{cls} = \frac{1}{n}\sum_{i=1}^{n}\left(f(x_i) - \widehat{Y}_i\right)^2 \tag{4}$$

$$L_{KD}(p_p, p_c) = \sum_{i=1}^{N} p_c(x_i) \cdot \log\left(\frac{p_c(x_i)}{p_p(x_i)}\right) \tag{5}$$

where $\widehat{Y}_i$ represents true label, p_p and p_c are represented by the probabilistic distribution output q_n^s, q_n^t, giving following equation:

$$p = Softmax(\frac{q_n}{T_0}) \tag{6}$$

According to the 'temperature' T_0, it is a parameter designed for soften probability distributions. Besides probabilistic level measurement, we also consider the intermediate feature difference after convolutional layers between two networks. For feature supporting loss L_{FL}, it could be given by:

$$L_{FL} = \gamma \left\| z_t - \frac{\sum_{t=0}^{T} z_s(t)}{T} \right\|_2 \tag{7}$$

where z_t and z_s are the last convolutional feature map of two networks, which stands for the last convolutional layer before FC layers, and γ, T represent the weight parameter and time steps, respectively. For SSNN feature, it requires to integrate information of each time step and generate new feature map in average. Since the two networks have the same feature size for each layer, we can directly match the feature maps and compute the feature loss without any additional operations.

For teacher model, we also do some optimation towards the pre-trained parameters during the training with student model. The loss function of teacher model is given by:

$$L_t = L_{cls} + \delta L_{KD} \tag{8}$$

where L_{cls} contributes as the main part and δ is a small coefficient to balance two losses.

For SNN training, it usually takes more epochs to converge than CNN. During training, we update the parameters of the two models together to transfer knowledge from high accuracy CNN to SSNN. According to the training step, the input for SNN is encoded into discrete spikes towards the Gaussian random number while the CNN receives inputs from the original data. The proposed training method is shown in Algorithm 1.

III. EXPERIMENTAL RESULTS

To demonstrate the performance of the proposed training scheme, we present experimental results over four open datasets listed in Table I. Among the datasets, cifar 10 and MNIST are famous image datasets wildly used in image recognition, while MIT-BIH and UCI-HAR datasets provide time-series signal classification tasks. We adopt a simple SNN training method through back propagation as the reference baseline also acted as student model [21]. For each scenario, raw data is encoded into different time steps in the same way. With higher time steps, the baseline spin-based SNN method receives more information encoded by spike and achieves higher accuracy, but it also brings more hardware resource utilization and energy consumption. The teacher model utilizes the raw data to pretrain first and optimizes together with student model. For the experiment on every dataset, the performance of our training scheme is marked in bold text as Student + KD while teacher model is marked in grey as a comparison. As can be seen, our training scheme demonstrates the consistent improvement of all baseline methods among four datasets with different spike encoding steps. Generally, with lower time steps, the increment brought by knowledge distillation is much more significant. Learning from teacher model could directly help student model mimic the teacher output which gains better performance. For example, in the UCI-HAR dataset classification task, which is applied to classify six different human actions, the training scheme improves the performance of spin-based SCNN by 1.62% from 81.81% to 83.43% with four-time steps. This suggests that by learning knowledge from a high-performance CNN model, we can improve the SSNN accuracy by a large margin.

TABLE I
PERFORMANCE COMPARISON OF PROPOSED AND SIMPLE TRAINING SCHEMES

Dataset	Type	Method	Structure	Time step	Acc
MNIST	Teacher	ANN	784-100-10	/	97.38
	Student [18]	Spin-based SNN	784-100-10	1	91.15
	Student+KD	Spin-based SNN	784-100-10	1	**93.07**
CIFAR-10	Teacher	CNN	6Conv+2FC	/	90.31
	Student [18]	Spin-based SCNN	6SConv+2SFC	8	84.03
				4	69.73
				2	48.89
	Student+KD	Spin-based SCNN	6SConv+2SFC	8	**85.43**
				4	**72.16**
				2	**57.69**
MIT-BIH	Teacher	CNN	5Conv+2FC	/	98.23
	Student [19]	Spin-based SCNN	5SConv+2SFC	8	87.06
				4	81.81
				2	71.63
	Student+KD	Spin-based SCNN	5SConv+2SFC	8	**89.53**
				4	**83.43**
				2	**74.53**
UCI-HAR [20]	Teacher	CNN	5Conv+2FC	/	95.39
	Student	Spin-based SCNN	5SConv+2SFC	8	74.54
				4	63.66
				2	47.57
	Student+KD	Spin-based SCNN	5SConv+2SFC	8	**76.60**
				4	**65.91**
				2	**53.53**

KD: proposed knowledge distillation method; Acc: training accuracy.

IV. CONCLUSION

In this paper, we propose a novel spike knowledge distillation method to improve the performance and training accuracy of all spin spike-based deep neural network. We utilize the proposed training scheme on four datasets and compared training results with or without our training scheme on small spin-based SCNN student model with different time steps. Results show that the proposed training scheme can guide knowledge learning to SNN and consistently improves the accuracy of all methods in any time step. This indicates the training scheme can be applied to SSNN based training method to improve training performance without any modifications with network structure.

ACKNOWLEDGMENT

This work was supported by the King Abdullah University of Science and Technology baseline fund.

References

[1] 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, 2019.

[2] M.-C. Chen, A. Sengupta, and K. Roy, “Magnetic skyrmion as a spintronic deep learning spiking neuron processor,” *IEEE Transactions on Magnetics*, vol. 54, no. 8, pp. 1–7, 2018.

[3] M.-H. Wu, M.-S. Huang, Z. Zhu, F.-X. Liang, M.-C. Hong, J. Deng, J.-H. Wei, S.-S. Sheu, C.-I. Wu, G. Liang, *et al.*, “Compact probabilistic poisson neuron based on back-hopping oscillation in stt-mram for all-spin deep spiking neural network,” in *2020 IEEE Symposium on VLSI Technology*, pp. 1–2, IEEE, 2020.

[4] F.-X. Liang, I.-T. Wang, and T.-H. Hou, “Progress and benchmark of spiking neuron devices and circuits,” *Advanced Intelligent Systems*, vol. 3, no. 8, p. 2100007, 2021.

[5] Y. Tokura and N. Kanazawa, “Magnetic skyrmion materials,” *Chemical Reviews*, vol. 121, no. 5, pp. 2857–2897, 2020.

[6] S. Li, W. Kang, X. Chen, J. Bai, B. Pan, Y. Zhang, and W. Zhao, “Emerging neuromorphic computing paradigms exploring magnetic skyrmions,” in *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 539–544, IEEE, 2018.

[7] K. M. Song, J.-S. Jeong, B. Pan, X. Zhang, J. Xia, S. Cha, T.-E. Park, K. Kim, S. Finizio, J. Raabe, *et al.*, “Skyrmion-based artificial synapses for neuromorphic computing,” *Nature Electronics*, vol. 3, no. 3, pp. 148–155, 2020.

[8] F. Tan, W. Gan, C. Ang, G. Wong, H. Liu, F. Poh, and W. Lew, “High velocity domain wall propagation using voltage controlled magnetic anisotropy,” *Scientific reports*, vol. 9, no. 1, pp. 1–6, 2019.

[9] Z. Zhang, Y. Zhu, Y. Zhang, K. Zhang, J. Nan, Z. Zheng, Y. Zhang, and W. Zhao, “Skyrmion-based ultra-low power electric-field-controlled reconfigurable (super) logic gate,” *IEEE Electron Device Letters*, vol. 40, no. 12, pp. 1984–1987, 2019.

[10] Y. Zhang, Z. Wang, J. Zhu, Y. Yang, M. Rao, W. Song, Y. Zhuo, X. Zhang, M. Cui, L. Shen, *et al.*, “Brain-inspired computing with memristors: Challenges in devices, circuits, and systems,” *Applied Physics Reviews*, vol. 7, no. 1, p. 011308, 2020.

[11] A. H. Lone and H. Fariborzi, “Skyrmion-magnetic tunnel junction synapse with long-term and short-term plasticity for neuromorphic computing,” *IEEE Transactions on Electron Devices*, 2022.

[12] A. H. Lone, H. Li, N. El-Atab, X. Li, and H. Fariborzi, “Voltage gated domain wall magnetic tunnel junction-based spiking convolutional neural network,” *arXiv preprint arXiv:2212.09444*, 2022.

[13] D. Wu, J. Yang, and M. Sawan, “Bridging the gap between patient-specific and patient-independent seizure prediction via knowledge distillation,” *Journal of Neural Engineering*, vol. 19, no. 3, p. 036035, 2022.

[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” *arXiv preprint arXiv:1910.01108*, 2019.

[15] S. Sun, Y. Cheng, Z. Gan, and J. Liu, “Patient knowledge distillation for bert model compression,” *arXiv preprint arXiv:1908.09355*, 2019.

[16] R. K. Kushawaha, S. Kumar, B. Banerjee, and R. Velmurugan, “Distilling spikes: Knowledge distillation in spiking neural networks,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4536–4543, 2021.

[17] G. Hinton, O. Vinyals, J. Dean, *et al.*, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[18] Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi, “Direct training for spiking neural networks: Faster, larger, better,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1311–1318, 2019.

[19] J. Jiang, F. Tian, J. Liang, Z. Shen, Y. Liu, J. Zheng, H. Wu, Z. Zhang, C. Fang, Y. Zhao, *et al.*, “Mspan: A memristive spike-based computing engine with adaptive neuron for edge arrhythmia detection,” *Frontiers in Neuroscience*, p. 1707, 2021.

[20] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, “A public domain dataset for human activity recognition using smartphones,” in *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, pp. 437–442, 2013.

[21] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, “Spatio-temporal backpropagation for training high-performance spiking neural networks,” *Frontiers in neuroscience*, vol. 12, p. 331, 2018.