Exploring the impact of random telegraph noise-induced accuracy loss in Resistive RAM-based deep neural network

Yide Du, Linglin Jing, Hui Fang, Haibao Chen, Yimao Cai, Runsheng Wang, Jianfu Zhang and Zhigang Ji *Member, IEEE*

***Abstract*— For Resistive RAM (RRAM)-based deep neural network, Random telegraph noise (RTN) causes accuracy loss during inference. In this work, we systematically investigated the impact of RTN on the complex deep neural networks (DNNs) with different datasets. By using 8 mainstream DNNs and 4 datasets, we explored the origin that caused the RTN-induced accuracy loss. Based on the understanding, for the first time, we proposed a new method to estimate the accuracy loss. The method was verified with other 10 DNN/dataset combinations that were not used for establishing the method. Finally, we discussed its potential adoption for the co-optimization of the DNN architecture and the RRAM technology, paving ways to RTN-induced accuracy loss mitigation for future neuromorphic hardware systems.**

***Index Terms*—Time-dependent variability, Random Telegraph Noise, RTN, RRAM, Neuromorphic computing.**

I. Introduction

Operating deep neural networks (DNN) in low-power mode for Artificial Intelligence (AI) has become the critical driver for edge computing, which is crucial to solve the latency issues for future internet-of-thing applications **[1]**. By performing matrix-vector multiplication in cross-bar arrays, resistive-switching memories (RRAM) have successfully lowered down the power consumption to nW level **[2-3]**. Therefore, the study of the interaction between non-ideal characteristics of RRAMs and the inference accuracy becomes essential and has attracted attention from both industry and academic in recent years **[4-5]**. Among all the non-ideal characteristics, the RRAM cell variation and the Random Telegraph Noise (RTN) can cause the deviation of the weights away from their pre-set values. The intrinsic cell variability is time-independent. After the weight values are obtained during the offline training, they can be mapped to RRAMs with the write-and-verify scheme **[6]** to ensure the accurate conductance values are assigned. However, RTN introduces time-dependent weight variations. Even the conductance value of each RRAM has been accurately mapped, it can still vary with time during the inference and thus lead to accuracy loss **[7]**. With further scaling of RRAM technology **[8]**, RTN is expected to be a big concern and thus needs special attention. The binary-based neuromorphic network has been proposed for RTN mitigation. However, this does not apply for the analog system, which is required for future high-accuracy applications. Several pioneer works **[4, 9-11]** has investigated the RTN-induced accuracy loss. However, they only assessed the simple perceptron network with simple datasets such as MNIST. The practical applications, such as pattern recognition and enhancement **[12]**, require complex deep neural networks (DNN). There is a lack of study regarding the RTN impact on DNN-based analog neuromorphic network.

This work is to fill this knowledge gap. By deploying GPU-based parallel computing, we systematically investigated the RTN impact on the accuracy loss for 8 mainstream complex networks with stacked convolutional layers and 4 major datasets. It is found that RTN-induced accuracy loss depends on both the dataset and the network structure and cannot be suppressed by using longer pulse width or strengthening a certain layer in the DNN structure. Moreover, the distribution of DIFF value, which is a figure of merit we defined in this work and can be extracted from any DNN with any dataset, exhibits a strong correlation with the RTN-induced accuracy loss. Based on this understanding, we proposed a new fast method for assessing the RTN-induced accuracy loss of mainstream DNN/dataset under any RTN levels. Finally, we show the potential use of this method for RTN mitigation through co-design between DNN architecture and RRAM technology.

This work was supported by National Key R&D Program of China under the grant no. 2019YFB2205000 and National Natural Science Foundation of China under the grant no. 61927901.

Yide Du, Linglin Jing, Haibao Chen and Zhigang Ji are with National Key Laboratory of Science and Technology on Micro/Nano Fabrication, Shanghai Jiaotong University, Shanghai, 200240, P. R. China (email: zhigangji@sjtu.edu.cn). Yide Du is also with Department of Micro/Nano Electronics, Shanghai Jiao Tong university.

Hui Fang is with Computer Science department in Loughborough University, Haslegrave Building, LE11 3TU.

Jianfu Zhang is with the Department of Electronics and Electrical Engineering, Liverpool John Moores University, Liverpool L3 3AF, U.K.

Yimao Cai, and Runsheng Wang are with Peking University. Beijing, China.

II. Simulation for RTN-induced accuracy loss

A. Empirical model for RTN simulation

10μm x 10μm bipolar-switch RRAMs with 5nm Ta_2O_5 dielectric and TiN metal electrodes are used. With different reset voltage, the conductance can be adjusted gradually, as shown in **Fig.1a**. Due to the stochastic nature, RTN introduces conductance fluctuation (δg). Recently, we showed that δg in RRAMs could be modelled in analogy to modelling RTN in nano-scaled FETs **[9]**: δg is the conductance fluctuation caused by charging-discharging of traps. Each trap induces a RTN amplitude of δI and there are n traps in each RRAM. The device-to-device variation is modelled by assuming δI and n

following the Exponential and Poisson distributions, respectively. As shown in **Fig.1b**, this simple model agrees well with the measured distributions.

To take the stochastic trapping and detrapping processes into consideration, we carried out the RTN measurement under read voltage of 0.1V on multiple devices with the speed of 100μs/point. The trap time constants are extracted with the Factorial Hidden Markov Model to ensure the devices with multi-trap can be analyzed **[13]**. As shown in **Fig.2**, both τc & τe follows similar lognormal distribution **[14]**. The fitting parameters of this distribution will be used in this work.

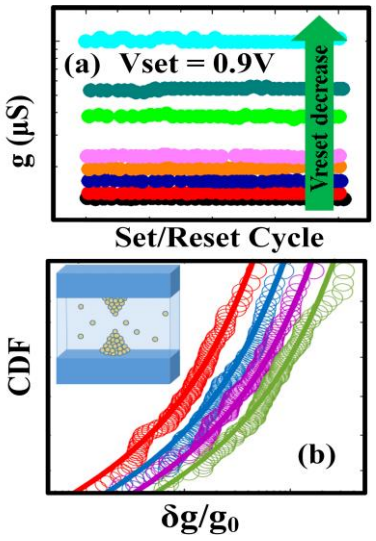

Fig.1 (a) Conductance with multiple levels by using different Vreset. Vset is fixed at 0.9V with forming Icc = 300μA. **(b)** CDF of relative RTN-induced conductance variation from the measurement (points) and lines (model). The inset shows the schematic of the device structure.

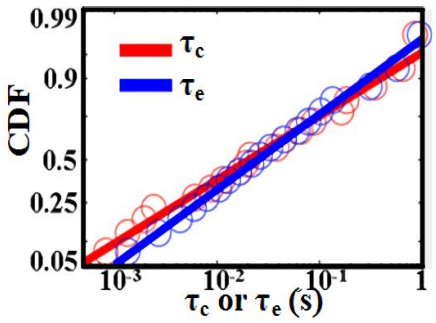

Fig.2 The distribution of capture, emission time under reading voltage of 0.1V.

For RRAM-based network, the weight of each synapse can be obtained through offline training and then mapped to the RRAM array. The procedure on the integration of RTN into each RRAM is summarized in **Fig. 3**: For each conductance g0, the trap number, n and the corresponding conductance fluctuation δg are firstly obtained using the method in **[9]**. For each trap, τc and τe are generated randomly from their lognormal distribution. Considering the typical running speed of 100ns **[15]**, we can calculate the filling probability, P_f. The total conductance with fluctuation, g_1, can then be obtained by summing up all the traps within one RRAM. Because of the stochastic nature of trapping/detrapping, g_1 varies with time, leading to time-dependent weight variation.

B. Acceleration for large-scale DNN simulation

The complex neural network, such as AlexNet, contains over 30 million synapses, which is ~ 400 times larger than the simple perceptron network (MLP). Since RTN is time-dependent, its impact on the weights varies when different images are inputted at different time. To reflect this in the simulation, the accuracy needs to be assessed on an image-by-image basis. As shown in **Fig.4**, for one input image, it takes ~10s to introduce RTN-induced fluctuations into all the synapses of MLP (red line). This time scales up with the size of DNN. DNNs with practical interests, such as AlexNET and VGG19, usually have a large number of synapses and thus the simulation time for one input image can take over 1000s. Considering the accuracy assessment with 1000 input images, the total simulation time becomes too long to afford. Since the introduction of RTN is an independent process for each synapse, the parallel computation can be deployed using multithreading either in CPUs or GPUs. The comparison is shown in **Fig.4**. The GPU-based parallel method can be ~70 times faster than the one without acceleration. This laid the foundation for us to assess RTN impact on various complex neural networks with different datasets and will be applied hereafter.

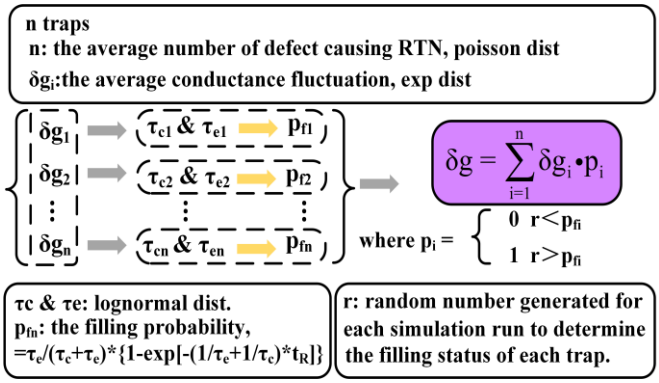

Fig.3 Procedure for introducing RTN into conductance fluctuation.

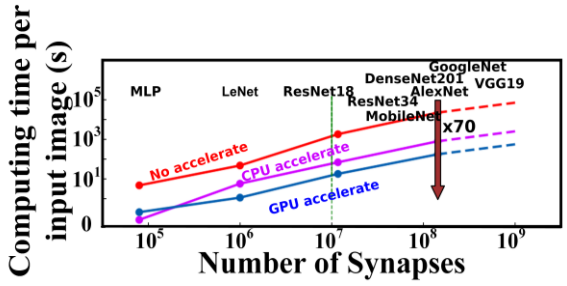

Fig.4 Comparison of the simulation speed for introducing RTN into RRAMs without acceleration and with acceleration using CPU- or GPU- based parallel computation. The specification of the PC is CPU i7-9700K, 3.60GHz, 32G, GPU: NVIDIA GeForce RTX 2080Ti.

III. RTN-INDUCED ACCURACY LOSS FOR COMPLEX DNNS

To investigate the impact of RTN on the complex networks

with stacked convolutional layers, we constructed 7 mainstream DNNs which has been widely used in recent years including LeNet, AlexNet, VGG16, VGG19, GoogleNet, MobileNet, and ResNet34 **[16-20]**. The construction follows their standard structure. The 3-layer perceptron network (MLP) was also constructed for comparison purpose **[9]**. For all DNNs, the rectified linear unit (ReLU) is used as the activation function after every convolutional and fully-connected layer, which allow us to scale the conductance of the RRAMs to represent synaptic weights. Four different datasets are also used, including MNIST **[16]**, fashion MNIST **[21],** Cifar10 **[22]**, and ImageNet **[20]**.

The input can be encoded with either the pulse amplitude **[23]** or the pulse number **[24-28]**. The amplitude-encoding method suffers from the I-V nonlinearity problem and also can potentially trigger unexpected SET operations. Therefore, the number-encoding method is widely used. In this work, we adopt the 8-bit pulse number encoding for the input. DNNs were trained with the gradient descent backpropagation until the accuracy reached a level similar to their reported value. To carry out the simulation, the well-trained weights in each layer were mapped to two simulated RRAM arrays which handle the positive and negative weights separately **[4]**. We use conductance between 1.25μS (800kΩ) to 12.5 μS (80kΩ), which is the range for our measured data. For each RRAM, RTN is introduced by following the procedure in **Fig.3**. During inference, 1000 images were used to evaluate the recognition accuracy of the network.

A. Impact of the DNN size

It is well known that increasing the size of DNNs to increase redundancy can reduce the RTN-induced accuracy loss. As shown in **Fig. 5a&b**, for a given combination of DNN structure and the dataset, this is indeed the case, where the inference accuracy increases when more synapses in used in each layer. This is not always the case when comparing among different DNNs and datasets. In **Fig.6a,** the accuracy loss for randomly-chosen 8 DNNs (different marker style) and 4 datasets (different colour) was assessed. The accuracy loss shows no clear correlation with the total number of synapse. For the same dataset, such as ImageNet, The VGG19 has the highest loss, although it has the highest number of synapse.

For the same DNNs, the loss depends on the datasets. **Fig.6b** shows the RTN-induced accuracy loss for three different DNNs with datasets of MNIST, fashion MNIST, and Cifar10. For each DNN, the MNIST dataset works well and the maximum accuracy loss is no more than 3%, which is comparable to the typical reported value. This also confirmed that the RTN we introduced into the simulation is not far away from practice. However, With the same level of RTN, Cifar10 and Fashion datasets show over 30% losses with the same DNNs, which is intolerable in practice. Therefore, we conclude that the RTN-induced accuracy loss depends on both DNN and dataset and should be assessed for each DNN and dataset combination. It is highly desirable to have a fast assessment method, therefore. This will be discussed further in section IV.

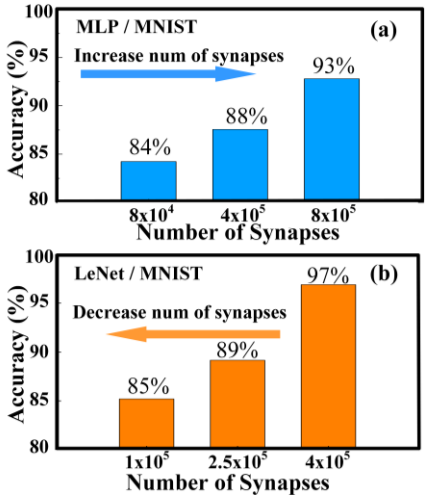

Fig. 5 Impact of the number of synapses on inference accuracy for (a) MLP and (b) LeNET.

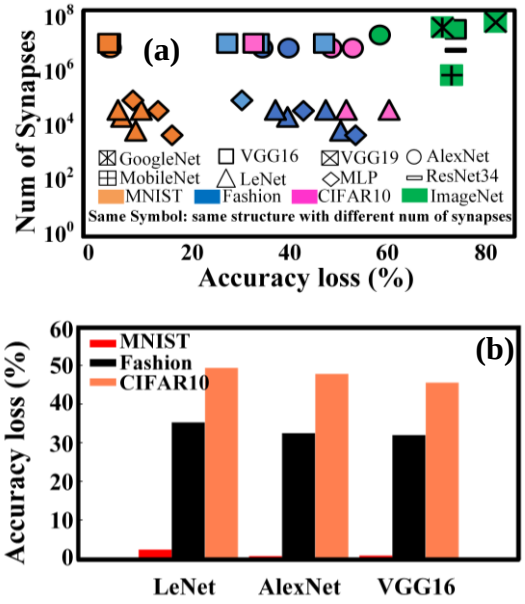

Fig.6 (a) Relationship between total synapse number and accuracy loss for 7 DNNs and 4 datasets. Different DNNs were represented with different marker style, and the datasets were with a different colour. **(b)** RTN-induced inference accuracy loss for three different DNNs and three different datasets. The accuracy loss is obtained from 1000 images**.**

B. Impact of the pulse width

In the circuit level, the realization of the analog matrix-vector multiplication calculation relies on the currents to be integrated within a certain time before triggering the neurons to respond. Therefore, reducing the DNN operating speed by using longer pulse width for input encoding is expected to suppress the RTN-induced accuracy loss through averaging effect. We compared the inference accuracy with different pulse width on different DNNs. As shown in **Fig.7**, the accuracy does not improve until reaching the millisecond region, which is already out of the practical-use domain. What is worth noting is that this simulation is based on the RTN we measured with slow

measurement in which only the relatively slow RTNs were captured. However, this limitation does not affect our conclusion because taking both the fast and slow traps into consideration can only increase the accuracy loss because the slow traps will never be averaged out. Therefore, we conclude that averaging with a longer pulse width is not effective to mitigate RTN-induce accuracy loss.

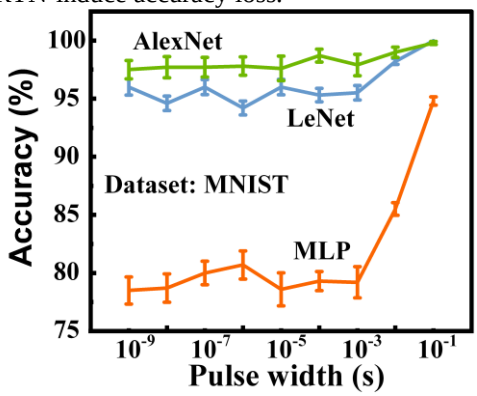

Fig.7 The relationship between inference accuracy of three different DNNs and the pulse width.

C. *Impact of the different layers in the DNN*

There is speculation that the RTN-induced accuracy loss is dominated by one layer of a DNN. We now investigate the impact of RTN from a specific layer on the accuracy. In **Fig. 8**, we randomly picked three different DNN/dataset combinations. To check the layer sensitivity, we only removed the RTN noise from one layer at a time. For the AlexNet DNN/ Fashion MNIST, the largest improvement in accuracy occurs when RTN is removed from the first convolutional layer (C1). The improvement reduces when moving further into the network. Similar trend is observed for AlexNet DNN/CIFAR10. For ResNet18/MNIST, the improvement becomes not obvious starting from the C2 layer. In all cases, the accuracy does not reach the 'Ideal' level and the extent of the improvement depends on both DNN and datasets. Therefore, mitigating RTN in one specific layer cannot be a general solution for the RTN-induced accuracy loss problem.

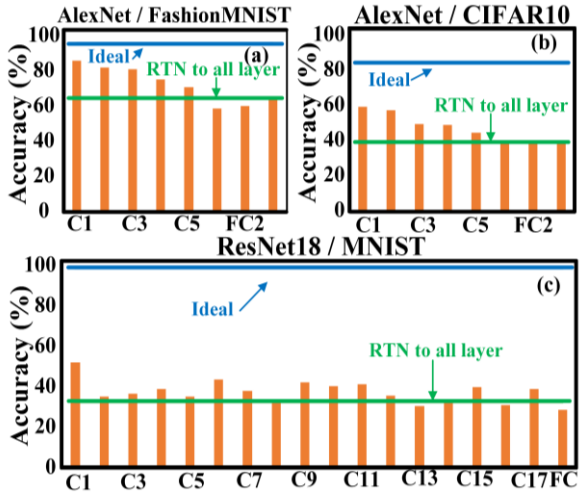

Fig. 8 Layer sensitivity for three different DNN/datasets.

IV. Fast Assessment Method on the Accuracy Loss

The assessment of the RTN-induced accuracy loss is important for its hardware implementation. In this section, we proposed a new fast assessment method.

A. *Origin for the RTN-induced accuracy loss in DNNs*

We first explore the key factor that controls the RTN immunity. LeNet with MNIST dataset is used for illustration purpose. By randomly picking one image "5" as input, ten outputs can be obtained. The blue line in **Fig. 9a** represents the 10 outputs from an ideal DNN without RTN. The largest output occurs in node 5 suggests the correct recognition. However, the difference between node 5 and node 8 is small. When RTN is taken into consideration, the output curve can vary for every inference even with the same input image (shown as grey lines). Sometimes, node 8 can exceed node 5 and thus cause the wrong recognition. Obviously, if this difference is small, RTN-induced conductance fluctuation can easily lift up the 2nd highest node and cause failure in pattern recognition. Therefore, the larger difference between the nodes with the highest and 2nd highest values should exhibit less chance for wrong recognition.

Base on this idea, we define the difference between the highest and the 2nd highest nodes extracted from the ideal DNN, as DIFF. We can get one DIFF value with each input in the given dataset, and distribution of DIFF can be obtained for each DNN/dataset combination. **Fig. 9b** compared the distributions of DIFF from three DNNs. Wherein, VGG16/MNIST includes more DIFF of large values, and this explains its small accuracy loss of 0.78% compared with 13.61% for MLP/MNIST. Therefore, it is expected that the distribution of DIFF with larger mean value, μ, and narrower variation, σ, should exhibit less accuracy loss.

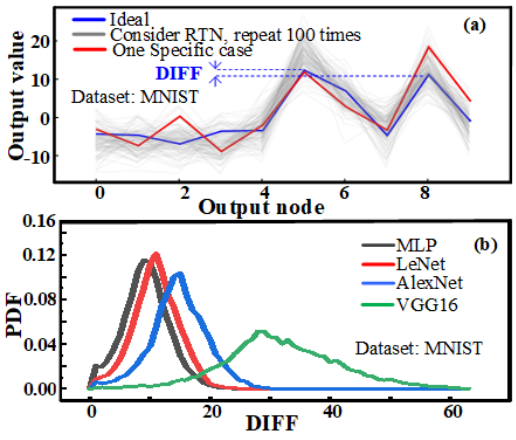

Fig.9 (a) Outputs of LeNet with given input image '5'. Blue line shows the ideal output. Difference between highest and 2nd highest nodes, DIFF, reflects RTN tolerance. The grey lines represent outputs with RTN for 100 repeats. Wherein, one case is marked in red to highlight the RTN-induced recognition failure. **(b)** PDF of DIFF for different DNNs using MNIST datasets.

B. *Method for the fast assessment*

Fig.10a plotted μ(DIFF)/σ(DIFF) from all DNNs/datasets against the corresponding accuracy loss. A clear correlation can

be obtained. A higher μ(DIFF)/σ(DIFF) represent a tighter statistical distribution, which is less vulnerable to RTN. Therefore, μ(DIFF)/σ(DIFF) can be used as a figure of merit to describe RTN-immunity of DNNs/datasets. This trend can be well described as a logarithmical relationship with Eqn (1). Wherein, a and b are the two fitting parameters.

$$\frac{\mu(DIFF)}{\sigma(DIFF)} = -a\ln(accuracy\_loss) + b \qquad (1)$$

RTN in RRAMs can vary with the quality of the fabrication process, which in turn affects the average number of defects causing RTN, n, and the average conductance fluctuation, δg, **[9]**. By using different n and δg, the accuracy loss for all the DNNs/dataset can be re-assessed. The previous work revealed that the accuracy loss only depends on the smallest conductance used in DNNs when the ratio between the largest and smallest conductance is higher than 10. Therefore, there exists a unique relationship between the parameter a&b and the n* δg from the smallest conductance that is to be used as DNN weight, as show in **Fig.10b**. Based on this, a simple solution for accuracy loss estimation can be established: After determining the range of conductance to be used to map the synapse weight, n* δg can be extracted by using the procedure described in ref.8. The parameters a & b can be determined which establishes the relationship between accuracy loss and μ(DIFF)/σ(DIFF). For any target DNN and dataset, we can extract μ(DIFF)/σ(DIFF), which is from the ideal case and no RTN is involved. Based on this extracted μ(DIFF)/σ(DIFF), the corresponding accuracy loss can be obtained from the accuracy loss~ μ(DIFF)/σ(DIFF) relationship.

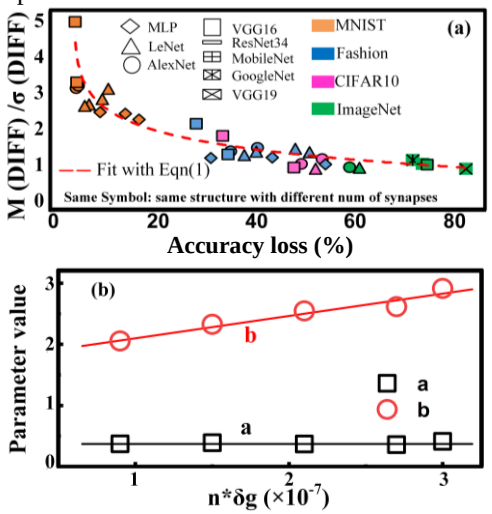

Fig. 10 (a) Relationship between the proposed figure-of-merit, μ/σ, and their corresponding accuracy loss using different datasets. **(b)** The relationship between the parameters a and b defined in Eqn (1) and n*δg, which correspond to the smallest conductance used for the whole DNN and represent the quality of RRAM.

B. Method validation

We further checked the validity of the proposed method. Wherein, we purposely selected four DNNs/datasets that were not used to establish our method. We also assume a better RRAM technology in which the average number of defects is reduced by half. The values predicted by the proposed fast method are compared with the values predicted by the tedious conventional RTN-simulation. The result is shown in **Fig.11** and the good agreement can be achieved. This supports that the proposed method has reasonable accuracy in assessing the accuracy loss.

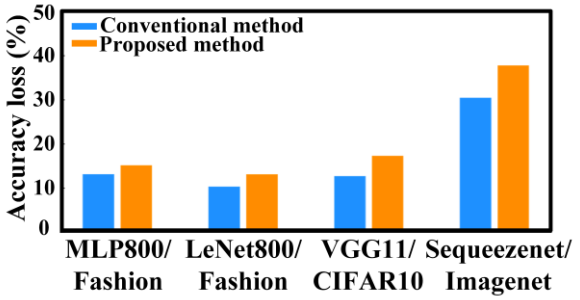

Fig. 11 Comparison between the prediction from the proposed method and the conventional method. For RTN setting, the average number of defects n is set to 1.2. The capture and emission time follow a logarithmic distribution, as shown in Fig.2. These four DNNs/dataset combinations were not used when establishing a fast assessment method.

Since our fast assessment method is established using the RTN results from the slow measurement, it is also important to check the impact of the fast traps on the assessment accuracy. Without loss of generality, we assigned a wider lognormal distribution for both capture and emission time which spans from 100ns to 1s. We also increase n from 2.4 to 3.6 to reflect that more traps are now contributing to the RTN. Then we assessed the RTN-induced accuracy loss using the conventional assessment method and compared with our proposed method. The results are shown in **Fig.12**. Overall, a good agreement has been achieved, which further confirms the validity of our proposed method.

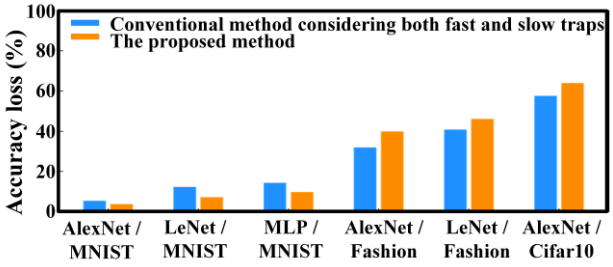

Fig. 12 Comparison between the prediction from the proposed method and conventional method when considering fast traps. For the RTN setting, the average number of defects n is set to 3.6. The capture and emission time follow logarithmic distribution spanning from 100ns to 1s.

Similar to IC industry in which Device/Circuit co-design has become the key root for reliability-aware design methodology **[29-30]**, the future design for the RTN-immune hardware can also be achieved through co-design between software-level DNN architecture and hardware-level RRAM technology. One illustration is given in **Fig.13** by adopting the proposed method: one can improve the technology to reduce RTN and thus move the μ(DIFF)/σ(DIFF) ~accuracy loss relationship (the red curve) to the left direction and also improve the DNN architecture for higher μ(DIFF)/σ(DIFF) on the curve. Therefore, controlling the RTN-induced accuracy loss within a certain range can be

achieved through the co-design between software-based DNN architecture and hardware-based RRAM technology.

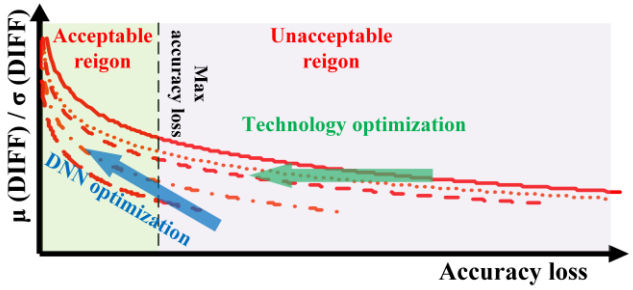

Fig.13 Illustration for the algorithm/device co-design for accuracy loss mitigation.

V. CONCLUSIONS

We investigated the impact of RTN on the inference accuracy for complex deep neural networks. The main contributions of this work are: (1) It is found that the DNN accuracy can be affected by both the dataset and the network structure. In addition, they cannot be suppressed by using longer pulse width or strengthening a certain layer in the DNN structure. (2) We proposed a figure-of-merit to assess the RTN-tolerance. Based on this, a simple method in assessing the accuracy loss of any DNNs and dataset is proposed and validated. We show such method can potentially be used for algorithm/device co-optimization, which can be useful for future RTN-immune DNN design.

REFERENCES

1. D. Ielmini and H. S. P. Wong, "In-memory computing with resistive switching devices," *Nat. Electron.*, vol.1, no. 6, pp.333-343, 2018, doi:org/10.1038/s41928-018-0092-2.
2. S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang and H. S. P. Wong, "A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: experimental characterization and large-scale modeling," *Proc. IEEE IEDM Tech.*, pp.10.4.1-10.4.4, 2012, doi: 10.1109/IEDM.2012.6479018.
3. B. Gao, Y. Bi, H. Y. Chen, R. Liu, P. Huang, B. Chen, L. Liu, X. Liu, S. Yu, H. S. P. Wong, and J. Kang, "Ultra-low-energy three-dimensional oxide-based electronic synapses for implementation of robust high-accuracy neuromorphic computation system", ACS Nano., vol. 8, no. 7, pp.6998–7004, 2014, doi: 10.1021/nn501824r.
4. Z. Chai, P. Freitas, W. Zhang, F. Hatem, J. Zhang, J. Marsland, B. Govoreanu, L. Goux, and G. S. Kar, "Impact of RTN on Pattern Recognition Accuracy of RRAM-Based Synaptic Neural Network", *IEEE Electron Device Letters*, vol.39, no. 11, pp. 1652–1655, 2018, doi: 10.1109/LED.2018.2869072.
5. S. Yu, "Neuro-inspired computing with emerging nonvolatile memory", *Proc. IEEE*, vol. 106, no. 2, pp. 260-285, 2018, doi: 10.1109/JPROC.2018.2790840
6. H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby, and G. W. Burr, "Recent progress in analog memory-based accelerators for deep learning," J. Phys. D. Appl. Phys., vol. 51, no. 28, 2018. doi: 10.1088/1361-6463/aac8a5
7. S. Yu, P. Y. Chen, Y. Cao, L. Xia, Y. Wang and H. Wu, "Scaling-up resistive synaptic arrays for neuro-inspired architecture: Challenges and prospect," *Proc. IEEE IEDM Tech.*, pp.17.3.1-17.3.4, 2015, doi: 10.1109/IEDM.2015.7409718.
8. G. C. Adam, A. Khiat and T. Prodromakis, "Challenges hindering memristive neuromorphic hardware from going mainstream," *Nat. Commun.*, vol.9, no.1, pp. 5267, 2018. doi: org/10.1038/s41467-018-07565-4.
9. J. Kang, Z. Yu, L. Wu, Y. Fang, Z. Wang, Y. Cai, Z. Ji and J. Zhang, "Time-dependent variability in RRAM-based analog neuromorphic system for pattern recognition," *Proc. IEEE IEDM Tech.*, pp.6.4.1-6.4.4, 2017, doi: 10.1109/IEDM.2017.8268340.
10. Z. Dong, Z. Zhou, Z. Li, C. Liu, P. Huang, L. Liu, X. Liu, J. Kang, "Convolutional Neural networks for image recognition and online learning tasks based on rram devices," *IEEE Trans. Electron Devices*, vol. 66, no. 1, pp. 793-801, 2017. doi: 10.1109/TED.2018.2882779.
11. D. Joksas, P. Freitas, Z. Chai, W. H. Ng, M. Buckwell, W. D. Zhang, A. J. Kenyon and A. Mehonic, "Committee Machines—A Universal Method to Deal with Non-Idealities in RRAM-Based Neural Networks", arXiv preprint arXiv:1909.06658, 2019.
12. Z. Dong, Z. Zhou, Z. Li, C. Liu, P. Huang, L. Liu, X. Liu, J. Kang, "Convolutional Neural networks for image recognition and online learning tasks based on rram devices," IEEE Trans. Electron Devices, vol. 66, no. 1, pp. 793-801, 2017. doi: 10.1109/TED.2018.2882779
13. F. Puglisi, "Random Telegraph Noise Analysis as a Tool to link Physical Device features to Electrical Reliability in Nanoscale Devices," *IEEE International Integrated Reliability Workshop (IIRW)*, pp.13-17, 2016, doi: 10.1109/IIRW.2016.7904891.
14. F. Puglisi, L. Larcher, A. Padovani and P. Pavan, "A Complete statistical Investigation of RTN in HfO2-Based RRAM in High Resistive State," *IEEE Trans. Electron Devices*, vol. 62, no. 8, pp. 2606-2613, 2015, doi: 10.1109/TED.2015.2439812.
15. Y. Jiang, J. Kang and X. Wang, "RRAM-based Parallel computing architecture using k-nearest neighbour classification for pattern recognition," Sci. Rep. 7, 45233, 2017, doi: 10.1038/srep45233.
16. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, 1998, doi: 10.1109/5.726791.
17. A Krizhevsky, I Sutskever, G E Hinton, "ImageNet classification with deep convolutional neural networks", *Advances in neural information processing systems*, pp. 1097-1105, 2012, doi: 10.1145/3065386.
18.G. Huang, Z. Liu, L. van der Maaten, K. Weinberger, "Densely connected convolutional networks", *Proc. CVPR*, pp. 2261-2269, 2017, doi:10.1109/CVPR.2017.243.
19. A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, K. Keutzer, "Squeezenext: Hardware-aware neural network design", *CVPR Workshops*, 2018, doi: 10.1109/CVPRW.2018.00215.
20. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition", *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, pp. 770-778, 2016, doi: 10.1109/CVPR.2016.90.
21. Y. LeCun, The MNIST database of handwritten digits [Online]. Available: http://yann.lecun.com/exdb/mnist
22. A. Krizhevsky, V. Nair, G. Hinton, The CIFAR-10 dataset, 2014, [online] Available: http://www.cs.toronto.edu/kriz/cifar.html.
23. Lashkare, S., Chouhan, S., Chavan, T., Bhat, A., Kumbhare, P., & Ganguly, U. (2018). PCMO RRAM for integrate-and-fire neuron in spiking neural networks. IEEE Electron Device Letters, 39(4), 484-487, doi:10.1109/LED.2018.2805822.
24. P. Y. Chen, X. Peng, and S. Yu, "System-level benchmark of synaptic device characteristics for neuro-inspired computing," 2017 IEEE SOI-3D-Subthreshold Microelectron. Unified Conf. S3S, 2017, vol. 2018-March, pp. 1–2, 2018, doi: 10.1109/S3S.2017.8309197.
25.P. Y. Chen and S. Yu, "Reliability perspective of resistive synaptic devices on the neuromorphic system performance," IEEE Int. Reliab. Phys. Symp. Proc., vol. 2018-March, pp. 5C.41-5C.44, 2018, doi: 10.1109/IRPS.2018.8353615.
26. Z. Li, P. Y. Chen, H. Xu, and S. Yu, "Design of Ternary Neural Network with 3-D Vertical RRAM Array," IEEE Trans. Electron Devices, vol. 64, no. 6, pp. 2721–2727, 2017, doi: 10.1109/TED.2017.2697361.
27.Y. C. Xiang et al., "Analog deep neural network based on nor flash computing array for high speed/energy efficiency computation," Proc. - IEEE Int. Symp. Circuits Syst., vol. 2019-May, pp. 7–10, 2019, doi: 10.1109/ISCAS.2019.8702401.
28. K. Moon et al., "RRAM-based synapse devices for neuromorphic systems," Faraday Discuss., vol. 213, pp. 421–451, 2019, doi: 10.1039/C8FD00127H.
29. S. Yu, Z. Li, P. Y. Chen, H. Wu, B. Gao, D. Wang, W. Wu and H. Qian, "Binary neural network with 16 Mb RRAM macro chip for classification and online training," *Proc. IEEE IEDM Tech.*, pp.16.2.1-16.2.4, 2016, doi: 0.1109/IEDM.2016.7838429.
30. Z. Ji, H. Chen and X. Li, "Design for reliability with the advanced integrated circuit (IC) technology: challenges and opportunities". *Sci. China Inf. Sci*. 62, 226401, 2019. doi:10.1007/s11432-019-2643-5.