Supplementary Information (SI) for Nanoscale.

Parylene-MoO_x Crossbar Memristors as Volatile Reservoir and Non-volatile Readout: A Homogeneous Reservoir Computing System. Supplementary materials.

Margarita A. Ryabova[1,2], Anna N. Matsukatova[1,3], Andrey V. Emelyanov[1,2,*], Alexander A. Nesmelov[1], Timofey D. Patsaev[1], Vyacheslav A. Demin[1]

[1] *National Research Center Kurchatov Institute, Moscow 123182, Russia*

[2] *Moscow Institute of Physics and Technology (National Research University), Moscow 141701, Russia*

[3] *Lomonosov Moscow State University, Moscow 119991, Russia*

[*] *Emelyanov_AV@nrcki.ru*

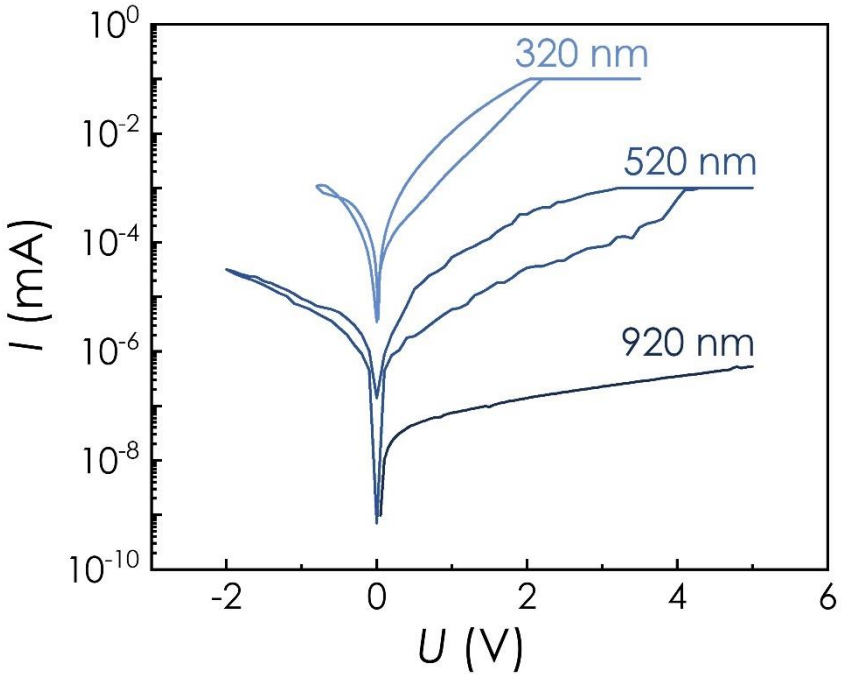

Figure S1. The median *I-V* curves for three crossbar samples with different thicknesses of the active nanocomposite PPX-MoO_x layer, specified in the graph.

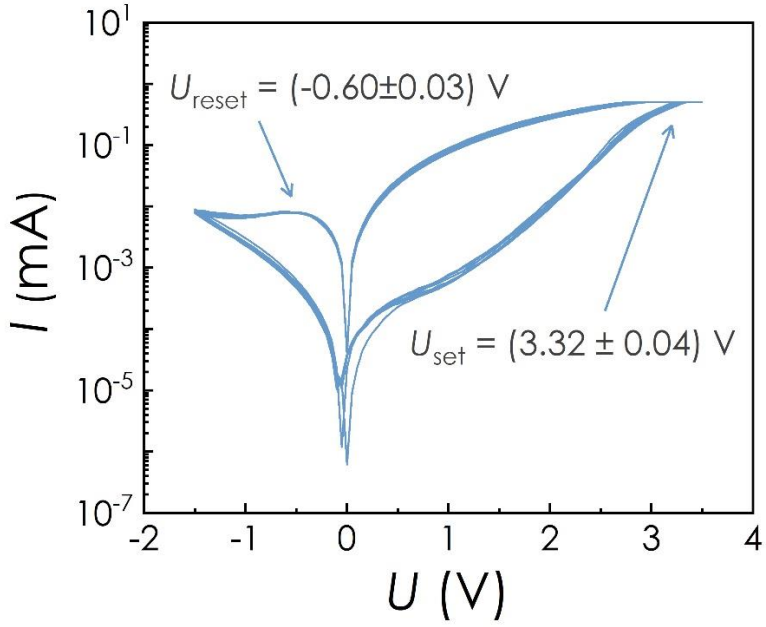

Figure S2. 10 *I-V* cycles of the PPX-MoO_x memristor, I_{cc} = 0.5 mA.

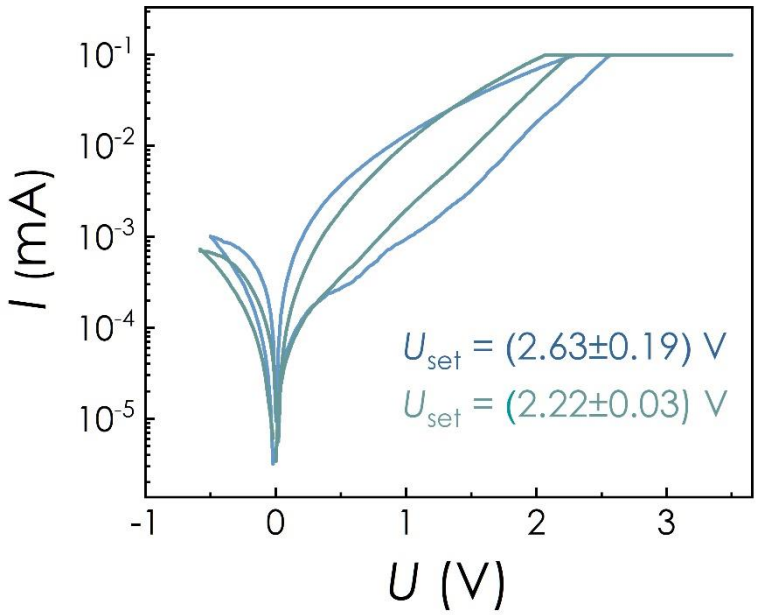

Figure S3. The *I-V* median curves of 2 different devices in the PPX-MoO_x based crossbar array, I_{cc} = 0.1 mA.

Supplementary Note 1

Stuck devices may occur in different situations: as a result of the set/reset operation (i.e., endurance failure [1]) or as an initial memristive fault (our case). Stuck-on failures may occur during the set process and are often linked to over-diffusion of the active electrode in the active layer [2]. In our case, stuck-off failures can be attributed to defects in the active layer or inaccuracy of the fabrication technology [3]. Latter is referred to as device yield or production yield problem in the literature. It is especially important for polymer-based memristors because of the structural inhomogeneity of most polymers, making a 90% production yield of polymer memristors a high one [4]. While we will strive to minimize such faults in our future work, endurance faults may still occur after a considerable number of resistive switching cycles, so the problem of stuck devices is of high importance.

References

[1] Yoon J.H., Song Y.-W., Ham W., Park J.-M., Kwon J.-Y. A review on device requirements of resistive random access memory (RRAM)-based neuromorphic computing // APL Materials. AIP Publishing, LLC, 2023. Vol. 11.no. 9. https://doi.org/10.1063/5.0149393
[2] Banerjee W., Kashir A., Kamba S. Hafnium Oxide (HfO2) – A Multifunctional Oxide: A Review on the Prospect and Challenges of Hafnium Oxide in Resistive Switching and Ferroelectric Memories // Small. John Wiley and Sons Inc, 2022. Vol. 18.no. 23. https://doi.org/10.1002/smll.202107575
[3] Liu C., Hu M., Strachan J.P., Li H. (Helen). Rescuing Memristor-based Neuromorphic Design with High Defects // Proceedings of the 54th Annual Design Automation Conference 2017. New York, NY, USA: ACM, 2017. P. 87. https://doi.org/10.1145/3061639.3062310
[4] Zhang B. et al. 90% yield production of polymer nano-memristor for in-memory computing // Nature Communications. 2021. Vol. 12. no. 1. P. 1984. https://doi.org/10.1038/s41467-021-22243-8

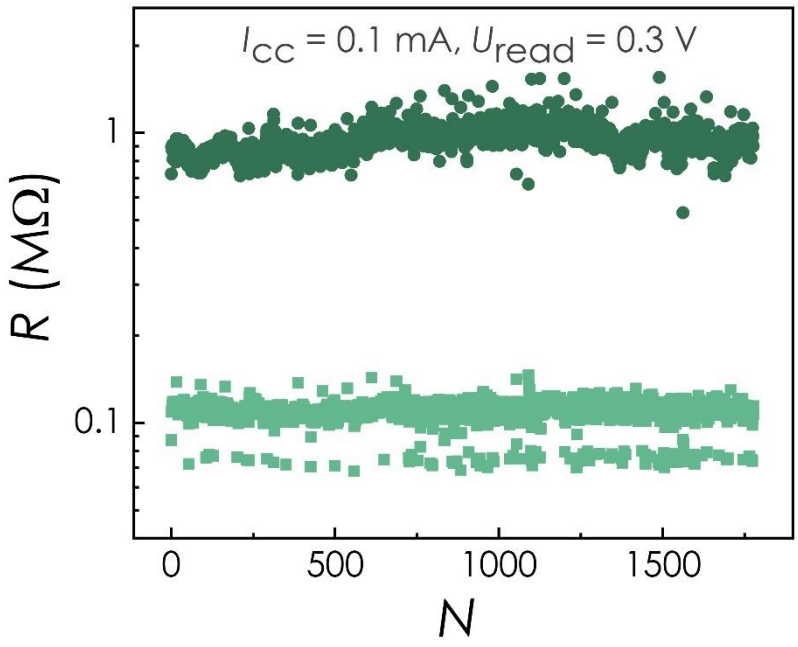

Figure S4. The endurance of the PPX-MoO_x memristor with decreased U_{read}.

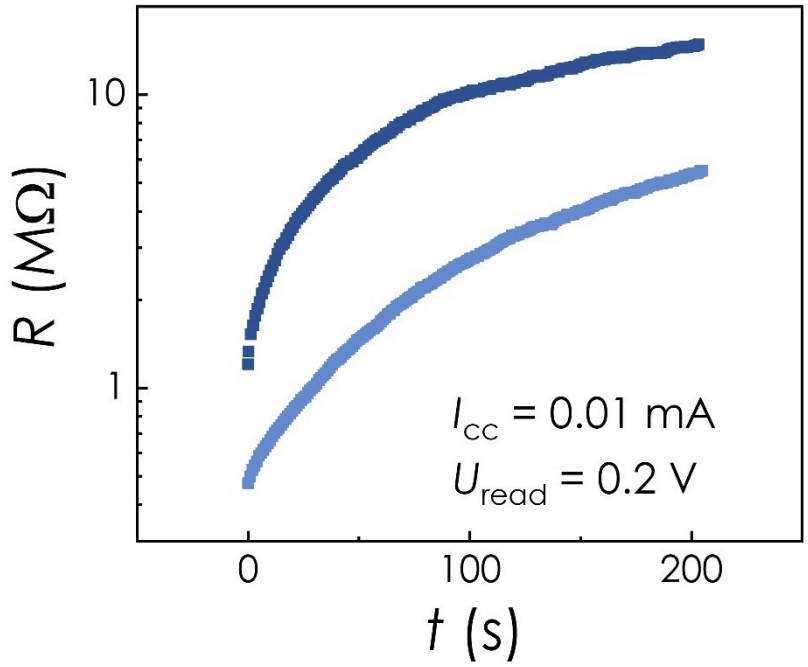

Figure S5. The retention of two different states, obtained in the volatile regime.

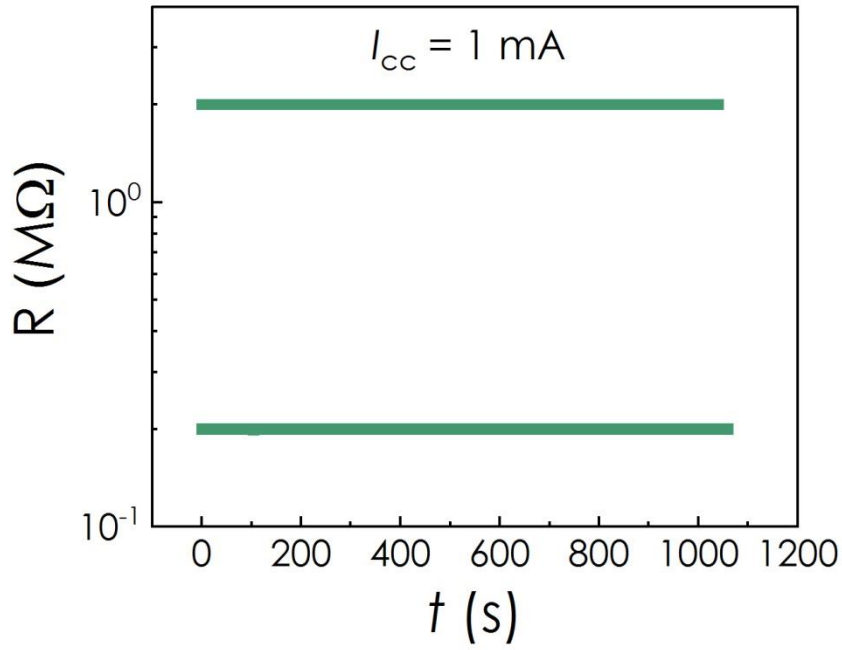

Figure S6. The long-term retention of two different states. LRS was set at compliance current 1 mA.

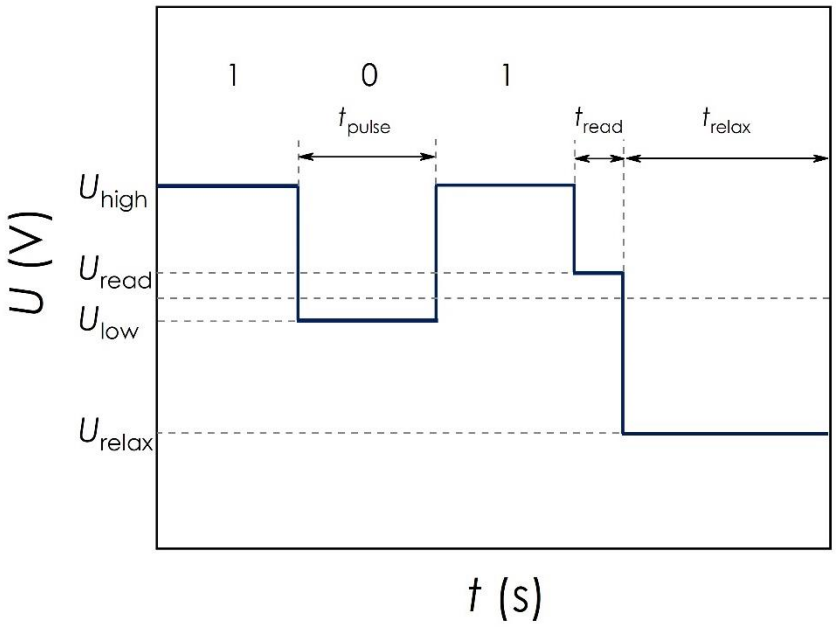

Figure S7. An example of the voltage pulse sequence, representing the '101' pattern. The fitted experimental parameters: U_{high} = 4 V, U_{read} = 1.4 V, U_{low} = -100 mV, U_{relax} = -0.7 mV, t_{pulse} = 25 ms, t_{read} = 25 ms, t_{relax} = 400 ms.

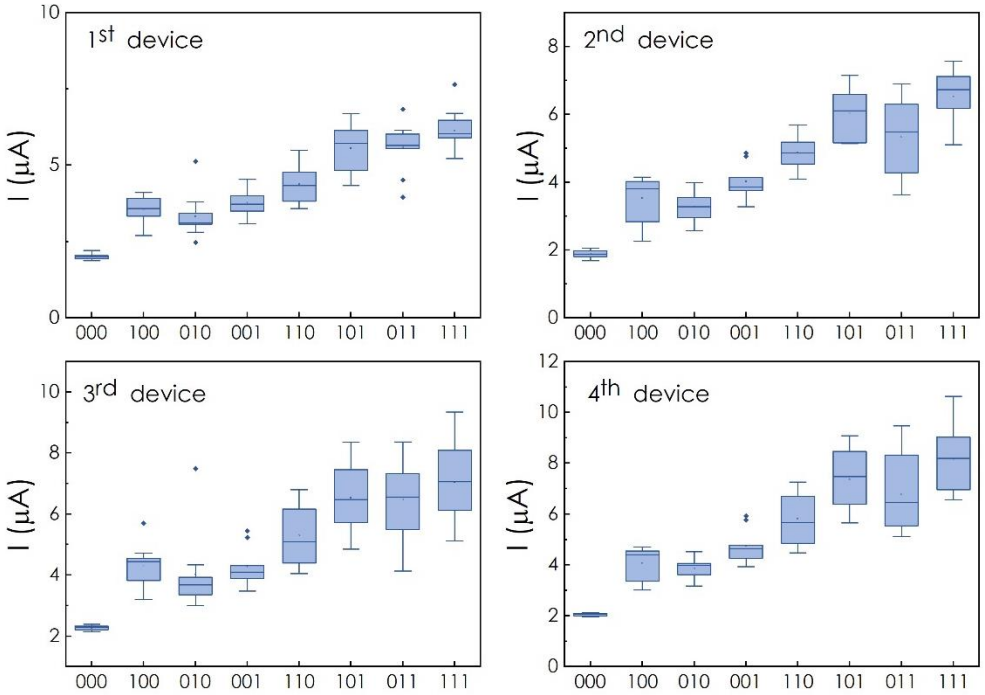

Figure S8. The distributions of the output currents for different patterns obtained with I_{cc} = 0.05 mA presented separately for 4 different PPX-MoO_x memristors, each graph demonstrates solely the C2C variations.

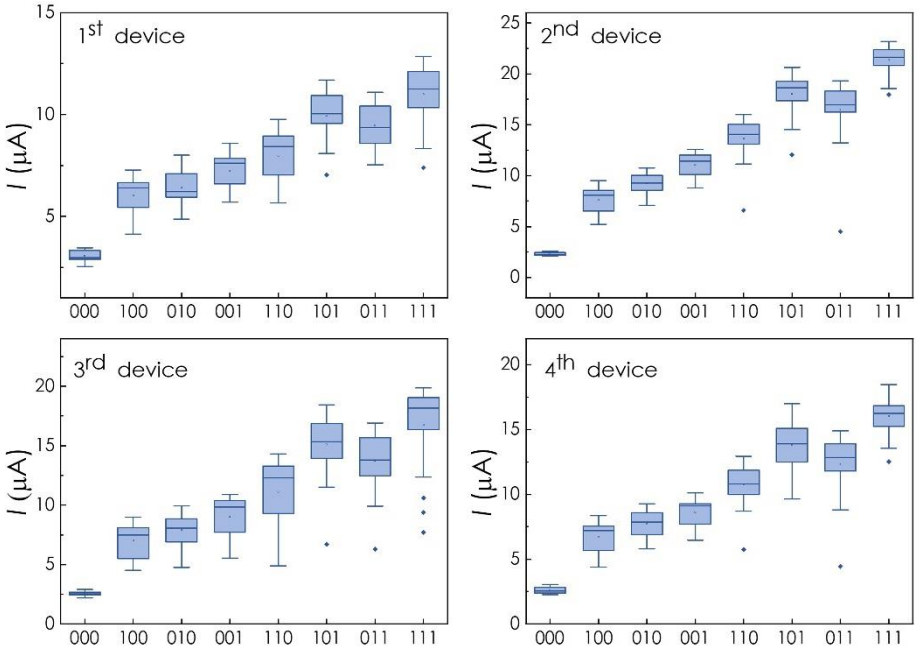

Figure S9. The distributions of the output currents for different patterns obtained with I_{cc} = 0.1 mA presented separately for 4 different PPX-MoO_x memristors, each graph demonstrates solely the C2C variations.

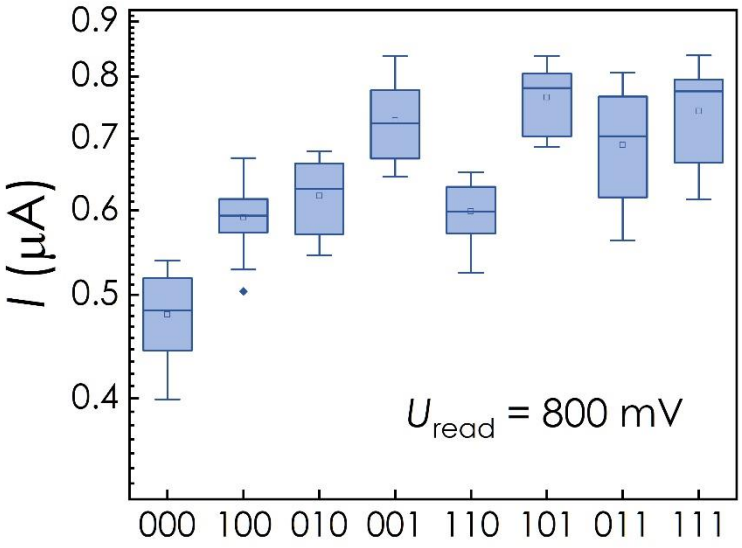

Figure S10. The distributions of the output currents for different patterns obtained with I_{cc} = 0.1 mA and U_{read} = 800 mV.

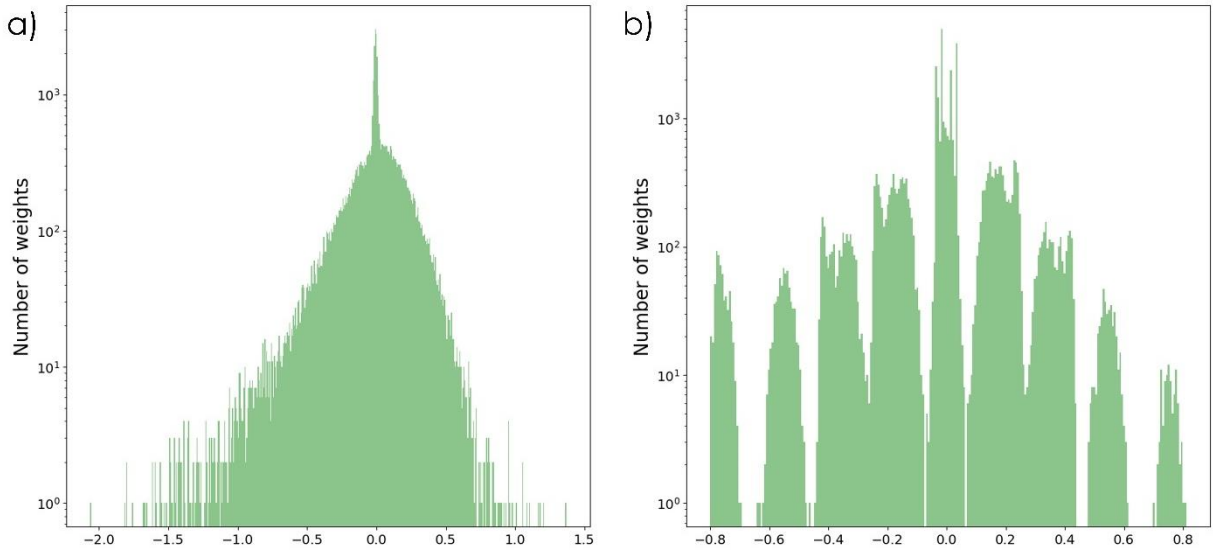

Figure S11. The network weights a) full-precision in the software readout layers, b) discretized in the memristive readout layers.

Supplementary Note 2

First, we specify what we mean by on-chip and off-chip training. On-chip training implies that the training process is performed directly via memristive devices. During training, each weight update takes place in hardware, i.e., all memristive states are updated accordingly, classification accuracies during training are tested in hardware. In order to simulate on-chip training, one has to introduce memristive variations, weight discretization, and other hardware memristive data for each weight update. In contrast, off-chip training means that the network is first trained in software with no memristive data input. Once training is completed, the states of the memristive devices are updated in correspondence with the software weights. We should note that we obtained some rough approximations of the resulting classification accuracy, and hardly any simulation could account for all memristive non-idealities in both cases.

Some devices (~ 10 %) in the crossbar were stuck in high-resistance state regardless of the input voltage. To simulate stuck devices, two random Boolean matrices (one with the reservoir dimensions, another one with readout dimensions) were created with a fixed ratio of true/false values (10% true, 90% false). The on-chip simulation was performed (Figure S12a). The weights were discretized according to the experimental data for each training epoch. The discretization process was slightly altered. The lowest possible value was attributed to each weight for which the corresponding Boolean matrix value was true, simulating the memristor stuck in high-resistance state. In this case, off-chip training led to low accuracies. In off-chip training, the weights are discretized in the last step, and the training process is not adjusted to the locations of the stuck devices. Accuracy is strongly dependent on the random locations of the stuck devices. Alternatively, on-chip training adjusts to the stuck devices during training and achieves higher accuracy despite higher impact of the memristive variations on the results.

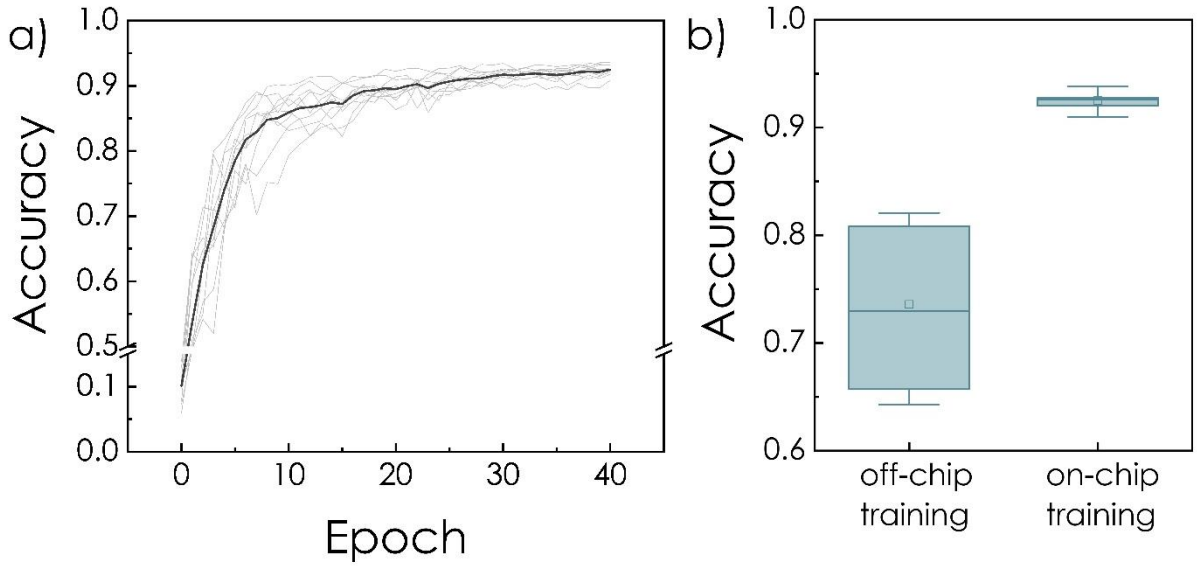

Figure S12. (a) The classification accuracy on the validation dataset during the on-chip training. (b) The classification accuracy on the test dataset for the off-chip and on-chip trained model.