LJMU Research Online

Joksas, D, Freitas, P, Chai, Z, Ng, WH, Buckwell, M, Li, C, Zhang, WD, Xia, QF, Kenyon, AJ and Mehonic, A

Committee Machines—A Universal Method to Deal with Non-Idealities in Memristor-Based Neural Networks

http://researchonline.ljmu.ac.uk/id/eprint/13408/

Article

Citation (please note it is advisable to refer to the publisher's version if you intend to cite from this work)

Joksas, D, Freitas, P, Chai, Z, Ng, WH, Buckwell, M, Li, C, Zhang, WD, Xia, QF, Kenyon, AJ and Mehonic, A Committee Machines—A Universal Method to Deal with Non-Idealities in Memristor-Based Neural Networks. Nature Communications. ISSN 2041-1723 (Accepted)

LJMU has developed **LJMU Research Online** for users to access the research output of the University more effectively. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Users may download and/or print one copy of any article(s) in LJMU Research Online to facilitate their private study or for non-commercial research. You may not engage in further distribution of the material or use it for any profit-making activities or any commercial gain.

The version presented here may differ from the published version or from the version of the record. Please see the repository URL above for details on accessing the published version and note that access may require a subscription.

For more information please contact researchonline@ljmu.ac.uk

http://researchonline.ljmu.ac.uk/

Committee Machines—A Universal Method to Deal with Non-Idealities in Memristor-Based Neural Networks

D. Joksas[1], P. Freitas[2], Z. Chai[2], W. H. Ng[1], M. Buckwell[1],
C. Li[3], W. D. Zhang[2], Q. Xia[3], A. J. Kenyon[1], and A. Mehonic[1]

[1]*Department of Electronic and Electrical Engineering,*
University College London, London (United Kingdom)

[2]*Department of Electronics and Electrical Engineering,*
Liverpool John Moores University, Liverpool (United Kingdom)

[3]*Department of Electrical and Computer Engineering,*
University of Massachusetts Amherst (United States of America)

Abstract

Artificial neural networks are notoriously power- and time-consuming when implemented on conventional von Neumann computing systems. Consequently, recent years have seen an emergence of research in machine learning hardware that strives to bring memory and computing closer together. A popular approach is to realise artificial neural networks in hardware by implementing their synaptic weights using memristive devices. However, various device- and system-level nonidealities usually prevent these physical implementations from achieving high inference accuracy. We suggest applying a well-known concept in computer science—committee machines—in the context of memristor-based neural networks. Using simulations and experimental data from three different types of memristive devices, we show that committee machines employing ensemble averaging can successfully increase inference accuracy in physically implemented neural networks that suffer from faulty devices, device-to-device variability, random telegraph noise and line resistance. Importantly, we demonstrate that the accuracy can be improved even without increasing the total number of memristors.

I. INTRODUCTION

Artificial neural networks (ANNs), with all of their variants, are now the main tools in machine learning tasks, such as classification. The vast amounts of data being constantly produced have enabled successful training and operation of ANNs. However, to achieve high inference accuracy, it is usually necessary for neural networks to have a large number of parameters. This results in both training [1] and inference [2] stages being time- and power-consuming. This is largely caused by the need to transfer data from memory to computing units—physical separation of memory and computing is the essence of any von Neumann system.

One of the most promising solutions to these problems is the paradigm of non-von Neumann computing and, specifically, analogue implementations of synapses (weights) in physical ANNs. Because there are many more synapses than there are neurons in ANNs, the matrix-vector multiplications, in which the synaptic weight values are used, are the costliest operations in these networks, both in terms of power and time. Computing directly in memory would minimise data transfers from off-chip memory, thus the most popular approach is using analogue memory devices as proxies for synaptic weights of ANNs (both fully connected and their variants [3, 4]). A common technique is to arrange such devices in a structure, called crossbar array, in which every device (or a pair of devices) is used to represent a single synaptic weight or, more generally, an entry in a matrix [5]. Memristive devices, such as phase-change memories (PCMs) [6, 7] or resistive random-access memories (RRAMs) [8, 9], have been considered as candidates for such tasks. Although here we focus on ex-situ training, such systems have been successfully utilised for in-situ training too [10, 11].

In memristive implementations of ANNs, the main concern is that various non-idealities associated with these devices can prevent these systems from achieving high accuracy [12, 13]. Examples of non-idealities affecting inference accuracy include, but are not limited to, devices not being able to electroform, devices stuck in one of the resistance states after electroforming, device-to-device (D2D) variability and random telegraph noise (RTN). When training analogue systems in-situ, limited endurance and non-linear resistance modulation too have to be taken into account. To mitigate the effects of these device non-idealities, it is often necessary to modify device structure [9], to use more advanced programming schemes

[14] or to use additional circuitry [15] or high-precision processing units [16] in conjunction with memristive elements. On the system level, there is an issue of line resistance which affects the distribution of currents and thus decreases the accuracy. These line resistance effects can be partially compensated for algorithmically [17] or partially mitigated by using multiple smaller crossbar arrays [18]. Examples of past efforts at dealing with these and other non-idealities of memristive devices and systems are listed in Table I; most of these non-idealities are still the main focus of the research in the neuromorphic community.

We propose a simple way to mitigate the effects of all types of non-idealities during inference. We suggest combining several non-ideal memristor-based neural networks into committees to achieve better accuracy. The committee machine (CM) method we propose significantly increases the inference accuracy and does not increase the computation time because memristive ANNs in such committees work in parallel.

In this work, we firstly explain the simulation setup—what networks were trained, how they were simulated and how they were combined into CMs. After that, follows the experimental part. We investigate three different types of memristor technology—tantalum/hafnium oxide-based (Ta/HfO_2), tantalum oxide-based (Ta_2O_5), and amorphous vacancy modulated conductive oxide-based (aVMCO) devices. By exploring their non-idealities relevant to inference—faulty devices, D2D variability, RTN, and line resistance—we use the experimental data to simulate memristive ANNs working individually and in committees.

II. RESULTS

A. Simulation setup

Fully connected ANNs were trained in software to recognise handwritten digits (using MNIST data base [19]). Architectures with one hidden layer were explored. Unless stated otherwise, the simulations used networks with 25 hidden neurons. However, networks with 50, 100 and 200 hidden neurons were additionally employed to evaluate the effectiveness of the proposed method while controlling for the total number of memristors required. Following training, weights of ANNs were mapped onto pairs of conductances using proportional mapping scheme (see [20]) to simulate memristor-based ANNs. Finally, these memristive

networks were disturbed using experimental data to reflect the effect of device- and system-level non-idealities.

After simulating physical non-idealities, the networks were combined into CMs that employed ensemble averaging (EA) [21]. The principle of EA is shown in Figure 1A—several networks are combined in parallel and then their outputs are averaged. After that, the prediction is made using the averaged vector—the prediction is the label corresponding to the largest entry in the vector.

CM methods are frequently used even with conventional ANNs. Methods, such as EA, often produce better accuracy than that of the best individual network in a committee [22]. Although there are other types of CMs besides EA, they often rely on training additional gating networks or boosting networks during the training stage. Using a gating network in this scenario would produce additional problems—to avoid it acting as a performance bottleneck, it too would have to be implemented on crossbar arrays. Various non-idealities would decrease the effectiveness of this gating network which is responsible for making the decisions about the whole committee of ANNs. Likewise, we speculate that boosting of networks would not be feasible in ex-situ training because it requires information about where individual ANNs perform poorly—this cannot be known precisely until they are implemented physically on crossbar arrays and the non-idealities manifest themselves. To authors' best knowledge, the application of boosting in the context of memristive neural networks seems to have been explored only once before [23]; as expected, it requires training each memristive implementation differently because non-idealities manifest themselves differently in different crossbar arrays.

There exist modifications of EA algorithm that could potentially perform better. One example of this is generalized ensemble method (GEM) which, instead of using equal weightings for each network during averaging (as in EA), uses a different one for each network [21]. These weightings are analytically determined by considering correlation of errors between different networks. But because [21] only considered networks with mean square error loss function (while our networks used cross-entropy loss function), this work does not explore GEM. Instead, we investigated whether it is possible to achieve a better performance by optimising the weightings numerically. This method, like GEM and others previously mentioned, might be impractical because, firstly, these weightings could be determined only after the ANNs are physically implemented on crossbars, and, secondly, the devices could change

117 throughout their lifetimes thus affecting the optimal weightings.

118 Even with the assumption that the devices would have perfect retention, we found that 119 optimisation of weightings achieves effectively the same performance. Because of these rea-120 sons, we focus only on EA in the main text, but present our results of optimising weightings 121 in Supplementary Figure S5. We stress that we are open to the idea that other CM methods 122 besides EA could be utilised successfully for ex-situ training in the context of memristive 123 ANNs. However, in this work we focus on demonstrating that CMs can be used to improve 124 the accuracy of memristor-based ANNs in general.

125 With EA, we find that even when the memristive ANNs, which go into a committee, all 126 use the same digital weights that are mapped onto crossbar arrays (see Figure 1B), committee 127 of memristor-based networks can still achieve higher accuracy than just a single non-ideal 128 network. Although all networks have the same *digital* weights before mapping, their physical 129 implementations (which we call "disturbances" in Figures 1B, C because they can usually 130 be represented by the modification of individual weights) will be different. For example, in 131 one crossbar array, a certain set of devices will be faulty, while in the other crossbar array, it 132 will be a different set. This will result in different physical implementations having slightly 133 different learned representations of the data set, or, to paraphrase, different networks will 134 be "damaged" differently by the non-idealities. This means that these committees will be 135 able to combine different representations, and thus achieve higher accuracy. However, by 136 definition, such approach would almost certainly not yield a committee accuracy that is 137 higher than the accuracy of a single digitally implemented network.

138 A better approach is to use different digital networks for different physical implementa-139 tions that go into a committee (see Figure 1C). This approach much more resembles the 140 conventional application of EA in computer science. In the context of memristive crossbar 141 arrays, it would not only help to mitigate the effects of the non-idealities (as in the case 142 of Figure 1B), but would also allow to combine the representations of digital networks that 143 were different even before the mapping stage. Most importantly, this method allows for a 144 committee to achieve higher accuracy which is sometimes even higher than that of individual 145 networks with digitally implemented weights. We thus used this method in this analysis. 146 An example comparison of these two approaches is presented in Supplementary Figure S8.

147 In this work, any given committee used only one network architecture but each network 148 was initialised differently before training, thus trained networks had different sets of weights.

149 Although it was not explored in this work, combining different network architectures in a
150 committee of memristor-based networks might be advantageous. Furthermore, in this work
151 we focus on fully connected ANNs but CMs could be applied to other variants of neural
152 networks as well. Due to the simplicity of EA, it could, for example, be employed in con-
153 volutional neural networks (CNNs) [24], which are often used for image classification. This
154 might be of interest as CNNs have been successfully implemented using crossbar arrays re-
155 cently [25]. However, crossbar implementations are naturally more suited to fully connected
156 networks, therefore we limit ourselves to this architecture but are open to exploring the
157 effectiveness of EA with memristive CNNs in the future.

158 ### B. Ta/HfO_2 RRAM

159 With array-level data available, Ta/HfO_2 experiments provide the most complete pic-
160 ture of device- and system-level non-idealities. In this subsection, we present not only the
161 analysis of faulty devices and D2D variability, but also careful consideration of the line resis-
162 tance effects. Ta/HfO_2 memristors do not exhibit apparent RTN and overall have excellent
163 retention properties [26], and thus are perfect candidates for inference application.

164 #### *1. Faulty devices and device-to-device variability*

165 The most energy-efficient procedure to modulate the conductance of memristors is by
166 the application of voltage pulses. In an ideal scenario, one would apply identical pulses
167 and observe constant increases in conductance with each pulse. This is rarely the case
168 in practise, but, fortunately, this type of behaviour is more relevant for in-situ training
169 where it is necessary to ensure linear adjustment of ANN's weights [27]. In ex-situ training,
170 conductance verification schemes can be used to program the devices precisely. Because the
171 devices would have to be programmed only once, one can spend additional resources to do so
172 accurately by applying SET (potentiation) and RESET (depression) pulses until a desirable
173 conductance state is achieved.

174 Even with this approach, there remain two obstacles—faulty devices and D2D variability.
175 It is observed in most memristor technologies that at least a small fraction of the devices
176 tends to get stuck in a particular conductance state. Additionally, even if not stuck, different

177 devices might behave differently; for example, they might have different conductance ranges.
178 Figure 2A shows conductance changes in Ta/HfO_2 RRAM devices (in a 128×64 crossbar
179 array) when they are applied with voltage pulses. We can see from the median values
180 that overall the devices' conductance tends to increase as more SET pulses are applied.
181 However, the wider bottom regions of the violin plots indicate that some devices are stuck
182 around high resistance state (HRS) and cannot set entirely no matter how many voltage
183 pulses are applied. There also exist devices that are stuck in low resistance state (LRS), or
184 simply do not span the full conductance range.

185 Figure 2A combines data from multiple SET cycles for each of the memristors, thus it
186 is important to understand how do these devices behave individually. Figures 2B-F show
187 conductance of 5 (out of 8,192) devices over 11 SET and RESET cycles. In the five dia-
188 grams, the radial component represents the conductance (in mS) and the angular component
189 represents the number of applied pulses. Figure 2B shows an example of preferable (and
190 typical) device behaviour—conductance changes in a continuous fashion and spans a wide
191 range of conductance values, from $\sim$0.1 ms to $\sim$1.0 ms. Although RESET cycles tend to
192 feature abrupt decreases in conductance, one can always repeat a cycle and exploit the more
193 predictable behaviour of SET cycles.

194 When encoding continuous numbers into crossbar devices' conductances, it is often prefer-
195 able to choose a large enough conductance range. Using data from Figure 2A, one could,
196 for example, choose the range between the first and the last median points (from $\sim$0.1 mS
197 to $\sim$1.0 mS). Device, whose behaviour is presented in Figure 2B, could be easily set to any
198 conductance within that range, as we have seen before. On the other hand, device, whose
199 behaviour is presented in Figure 2C, although operating in a predictable fashion, has smaller
200 conductance range. We can see that in all cycles, its conductance does not exceed 0.8 mS.
201 This is an example of D2D variability that can make it difficult to choose optimal operating
202 range and set the conductance of all devices precisely.

203 Device, whose behaviour is presented in Figure 2D, shows high cycle-to-cycle variability.
204 Although that could prove to be a problem in some applications, this specific device might
205 perfectly serve its purpose in ex-situ training of ANNs. We can observe that this device
206 spans the same conductance range as device from Figure 2B, even if in an unpredictable
207 manner. Because all states in the full range are, in theory, achievable, one can cycle the
208 device multiple times until it is set to the required conductance level.

209 Lastly, we have devices whose negative effect is most difficult to mitigate—faulty devices.
210 Figure 2E shows behaviour of a device stuck at high conductance values, while Figure 2F
211 shows behaviour of a device stuck at low conductance values. No matter how many pulses
212 the devices are applied with or how many times they are cycled, they exhibit almost no
213 conductance variation and thus, in most cases, cannot be used to encode information.

214 Knowing that some devices perform like the ones whose behaviour is shown in Fig-
215 ures 2C,E,F, it is important to minimise their negative effect. If the conductance that a
216 device has to be set to is outside that device's range, it is sensible to set it to the closest
217 achievable conductance. Although there is little that can be done about fully stuck memris-
218 tors, it is possible to optimise the behaviour of devices like the one in Figure 2C that simply
219 have smaller conductance range. For example, if such a device has to be set to 0.9 mS, one
220 would set it to the highest achievable conductance ($\sim$0.8 mS). In the following simulations
221 involving faulty devices and D2D variability, operating range between the first and the last
222 median points was used, the devices were chosen randomly from the 128×64 crossbar and
223 set to the most desirable states, as described in this paragraph.

224 *2. Line resistance*

225 The effect of line resistance can be extremely detrimental in many crossbar-based im-
226 plementations of ANNs. That is especially the case if the crossbars used are large and the
227 resistance of the interconnects is high (compared to memristors' resistance). Because in a
228 neural network many of the inputs are non-zero at any given time, a lot of current accumu-
229 lates in the bit lines which results in significant voltage drops across the interconnects, and
230 thus the current distribution across the crossbar is affected in a major way.

231 The Ta/HfO_2 crossbar has shape 128×64 and so this shape was chosen for all the simula-
232 tions involving line resistance. Even relatively small ANNs of architecture 784(+1):25(+1):10
233 would need $2 \times (785 \times 25 + 26 \times 10) = 39,770$ memristors to be implemented. Even if not
234 all the inputs were used at any given time, it would not be possible to fit all the memristors
235 onto a single crossbar of shape 128×64. To overcome this, we decided to simulate multiple
236 crossbars, each of which would implement a subset of the synaptic weights, but, for a given
237 synaptic layer, would all compute in parallel. Because $\lceil 785/128 \rceil = 7$, seven crossbars were
238 used to implement the first synaptic layer; the first crossbar utilized bottom 113 word lines,

while the other six crossbars used bottom 112 word lines because $113 + 6 \times 112 = 785$. The second synaptic layer was implemented using eighth crossbar utilizing its bottom 26 word lines.

Figure 3A shows an example of how the first synaptic layer of 784(+1):25(+1):10 neural network could be implemented. Specifically, it shows how the first subset of weights would be implemented using one of the crossbars. Because we use proportional mapping scheme, positive and negative weights would be implemented in different bit lines. In Figure 3A, memristors designated to implement positive weights are coloured in blue, memristors designated to implement negative weights are coloured in orange and unelectroformed memristors are coloured in black. Because simulations were constrained by experimental data, some of the devices were left unused and assumed to be unelectroformed. In practise, the crossbars could be manufactured to fit the geometry of the ANNs.

In each synaptic layer, the corresponding output currents from each of the crossbars would be added together. Additionally, output currents at the bit lines implementing negative weights would be subtracted from the output currents at the neighbouring bit lines (to their left) implementing positive weights. For example, in the example configuration of Figure 3A, output current at the 2nd bit line would be subtracted from the output current at the 1st bit line, etc.

Unfortunately, even when using multiple smaller crossbars, the interconnects can significantly disturb current distribution in the crossbar. Average output current decreases due to line resistance in all seven crossbars of Ta/HfO_2 devices (whose resistance ranges from $\sim 1\,\mathrm{k\Omega}$ to $\sim 11\,\mathrm{k\Omega}$, and their interconnect resistance is $0.35\,\Omega$ and $0.32\,\Omega$ in the word and bit lines, respectively), are shown in the heatmap in Figure 3B. We can see that the current decreases can range from $\sim 12\%$ at the outputs nearest to the applied voltages to $\sim 16\%$ at the outputs in the rightmost bit lines that are used. In the supplementary information, we provide a possible strategy of mitigating line resistance effects in supervised learning. This scheme was not employed in the simulations described in the main text because we wanted to find out how well the CM method would deal with noticeable line resistance effects.

267 *3. Inference accuracy*

268 Figure 4 shows the accuracy of individual networks, as well as of their committees; mem-
269 ristive ANNs were simulated by taking into account three non-idealities of Ta/HfO_2 crossbar
270 explored earlier—faulty devices, D2D variability and line resistance. As indicated by the
271 yellow box plot in Figure 4, individual networks implemented digitally achieve ∼95.9% me-
272 dian accuracy. Networks disturbed to reflect the effect of non-idealities achieve ∼91.0%
273 median accuracy, as indicated by the vermilion box plot. Although that is a substantial
274 drop in accuracy, we see that as more networks are added to the committee, the more the
275 accuracy increases. When 5 networks are used in a committee, median accuracy increases
276 up to ∼95.7%, as indicated by the rightmost green box plot.

277 C. Ta_2O_5 RRAM

278 In order to explore the effectiveness of minimising adverse effects of RTN, we use another
279 memristor technology based on Ta_2O_5. To investigate RTN, measurements from a single
280 device were considered. To simulate line resistance effects, interconnect resistance from
281 Ta/HfO_2 was used and the same crossbar shape was assumed.

282 *1. Random telegraph noise*

283 Memristors often suffer from RTN resulting in a different accuracy at any given instant
284 in time. Ta_2O_5 device was characterised by measuring the current of 8 resistance states
285 multiple times. Figure 5 shows the cumulative probability plots for those resistance states,
286 together with lognormal fits modelling the nature of RTN. One of the things that the figure
287 reveals is that higher resistance states suffer from higher degree of RTN. Fits for every
288 resistance state, together with occurrence rates (see Supplementary Table SII), were used
289 to disturb the weights of ANNs in order to reproduce the effect of RTN.

290 *2. Inference accuracy*

291 The results combining RTN and line resistance effects for Ta_2O_5 device are shown in
292 Figure 6. From the difference in median accuracy between yellow and blue box plots, we can

notice that there is a significant drop in accuracy simply due to mapping of weights onto conductances. That is not surprising given that only 8 states were available for mapping. One can also observe that further drop in median accuracy due to non-idealities is not as severe—it drops to $\sim$94.1%. The RTN disturbance magnitude is limited to $<$100% in most cases, which possibly contributes to its smaller effect on accuracy. Additionally, Ta_2O_5 device has much higher resistance (ranging from $25\,\mathrm{k\Omega}$ to $200\,\mathrm{k\Omega}$), thus line resistance is also less of a concern. When non-ideal networks are combined into committees of 5, the median accuracy jumps to $\sim$96.5%—even higher than the software baseline of individual networks. This reveals additional trend seen in all the simulations performed—the higher the accuracy of the individual non-ideal memristive networks, the higher the accuracy of the committees that they are part of.

D. aVMCO RRAM

Further, we consider a third memristor technology—one based on aVCMO materials. We test the effects of RTN by considering measurements from a single device. Line resistance effects were simulated by using interconnect resistance and shape of Ta/HfO_2 crossbar array.

1. Random telegraph noise

Figure 7 shows the cumulative probability plots for 8 resistance states of an aVMCO device suffering from RTN. Like in Ta_2O_5, we observe that higher resistance states experience RTN of higher magnitude. However, compared to Ta_2O_5, the RTN magnitude is much more predictable. Fits for each of the 8 resistance states, together with occurrence rates (see Supplementary Table SIII), were used to simulate the effect of RTN in aVMCO-based neural networks.

2. Inference accuracy

The results combining RTN and line resistance are shown in Figure 8. As with Ta_2O_5, we see a large drop due to mapping onto conductances—consequence of very few states available for mapping. More interestingly, the accuracy of individual memristor-based networks with

319 and without non-idealities is almost identical. That is because the occurrence rate of RTN
320 in aVMCO device is small and there is a much smaller probability of RTN having large
321 magnitude. Additionally, resistance of aVMCO device is even higher than that of Ta_2O_5
322 device—it ranges from $1\,M\Omega$ to $7.5\,M\Omega$. Therefore, line resistance has even a smaller effect
323 in a hypothetical array of aVMCO devices. Due to median accuracy of individual non-ideal
324 memristor-based networks being higher ($\sim$94.6%), the median accuracy of committees is
325 higher too—in committees of size 5 it increases to $\sim$96.7%.

326 III. DISCUSSION

327 The results from the previous section suggest that the method of using committee ma-
328 chines to improve the accuracy of memristive neural networks is technology- and non-ideality-
329 agnostic. CMs can mitigate the effects of faulty devices, D2D variability, RTN and line
330 resistance in combination with each other. Although CM method is slightly less effective
331 with large line resistance (see discussion in the supplementary information), in all cases, we
332 observe that the accuracy of individual non-ideal networks largely determines the accuracy
333 of committees. That is consequential because it means that although committees always
334 increase the accuracy, there is still an incentive to optimise the devices and systems that
335 implement these networks—the higher the accuracy of individual networks, the higher the
336 accuracy of the committees.

337 It is also important to consider whether using larger networks, instead of committees of
338 smaller networks, would yield the same results if the same number of synapses (or mem-
339 ristors) was used in the large network as in the committee of smaller networks. In our
340 previous work we found that the accuracy of networks before disturbance (which we call
341 "starting accuracy") has a huge effect on the robustness to non-idealities—the larger the
342 starting accuracy, the more robust the networks become [20]. One way to achieve higher
343 starting accuracy is to have larger networks, e.g. if we have a network with one hidden layer,
344 we might increase the number of neurons in that hidden layer, which would likely result in
345 higher accuracy after training and thus higher robustness.

346 Figure 9 shows a comparison of CMs of memristor-based networks disturbed using faulty
347 devices and D2D variability data from Ta/HfO_2 crossbar, when controlled for the total
348 number of memristors that is required to implement them (line resistance was not taken

into account due to long time required to simulate it in large networks). We can observe that committees of two networks, each with 25 hidden neurons, (leftmost data point of the orange curve) achieve ∼0.9% higher median accuracy than individual networks with 50 hidden neurons (second data point from the left in the vermilion curve), despite both requiring almost identical total number of memristors. Committees of two networks, each with 100 hidden neurons, (third data point from the left in the orange curve) achieve ∼1.1% higher median accuracy than individual networks with 200 hidden neurons (rightmost data point in the vermilion curve), even though both require almost the same total number of memristors. Even larger improvement is gained when committees of four networks, each with 50 hidden neurons, (second data point from the left in the blue curve) are used instead—then the accuracy is improved by ∼1.5%, with almost the exact total number of memristors used.

For different non-idealities and even different training schemes of the ANNs, the equivalents of Figure 9 might be different, but there are a few common characteristics in all of them. In all cases, for a given total number of memristors used, there is an optimal number of networks that should be used in a committee. Additionally, we observe that the more severe a non-ideality is, the more apparent the effectiveness of committees becomes. Finally, sometimes the committees (for a fixed total number of memristors) might achieve lower accuracy than individual networks but only if the networks that they replace are very small and the non-ideality is not very detrimental. If the networks that are being replaced with committees of smaller networks, are sufficiently large, the committees will achieve higher accuracy. An example of that is shown in Supplementary Figure S7 where aVMCO device is minimally affected by the non-idealities and so the advantage of committees becomes apparent only when replacing larger networks.

The reason why committees work in the context of non-ideal implementations and why they work best when they are used to replace large networks might, to some extent, lie in their training. When it comes to training fully connected networks, their accuracy tends to saturate as more parameters are added. Supplementary Figure S4 shows that networks with 50 hidden neurons can be trained to achieve significantly higher accuracy than networks with 25 hidden neurons. However, networks with 200 hidden neurons achieve only slightly higher accuracy than networks with 100 hidden neurons. This also means that networks with 200 hidden neurons will be only slightly more robust to non-idealities than networks with 100

381 hidden neurons. When such networks are affected by non-idealities, their accuracy drops
382 to similar values but the smaller network can work in a committee with other networks,
383 totalling almost the same number of memristors as the large network, but achieving higher
384 accuracy overall. This is the most likely reason why the committees of smaller networks are
385 effective at dealing with non-idealities, especially when replacing large networks.

386 In addition to the accuracy improvements, committees can provide flexibility in mem-
387 ristive implementations of neural networks. Digital implementations of ANNs have very
388 predictable behaviour due to the precision of digital logic. Analogue implementations, on
389 the other hand, can vary greatly even if they use the same weights before the mapping
390 onto conductances—that is a result of the stochastic nature of memristors that implement
391 these ANNs. The parallel and modular nature of committee machines makes memristive
392 systems much more flexible. For example, if the verification accuracy of one of the ANNs in
393 a memristor-based CM deteriorates below acceptable levels, its outputs could be disabled
394 to ensure higher accuracy of the rest of the committee.

395 Importantly, this introduced parallelism comes at almost no extra cost. For a fixed total
396 number of memristors, a committee of smaller networks, compared to a large individual
397 network, would only require a few additional output and bias neurons, and an averaging
398 functionality, which could potentially be implemented in hardware. For example, an ANN
399 with 50 hidden neurons would require 846 neurons in total, while a committee of two ANNs,
400 each with 25 hidden neurons (and thus requiring almost the same total number of memris-
401 tors), would require 857 neurons in total.

402 In summary, our simulations employing experimental data from three different types of
403 memristive devices show that committee machines employing ensemble averaging can be used
404 to mitigate the effects of device- and system-level non-idealities in memristor-based neural
405 networks. EA allows to achieve higher inference accuracy in physically implemented neural
406 networks that suffer from faulty devices, device-to-device variability, random telegraph noise,
407 and even line resistance. This method is a universal way to deal with the most common
408 non-idealities and is straightforward to implement during the fabrication stage. Increased
409 modularity of these memristive neural network systems will increase not only their inference
410 accuracy, but also their robustness and flexibility, even without the need to sacrifice area.
411 Although some level of non-idealities in memristors is unavoidable, CM method allows us
412 to deal with these on the system level and is agnostic to a particular technology or, to some

413 degree, type of the non-ideality.

414 ## METHODS

415 ### Experiments

416 Ta/HfO_2 RRAM 1T1R array consists of NMOS transistors fabricated in a commercial
417 fab (feature size of 2 µm) and Pt/HfO_2/Ta devices. The bottom electrode was deposited by
418 evaporation of 20 nm Pt layer on top of a 2 nm tantalum (Ta) adhesive layer; the electrode
419 was patterned by photolitography and a lift-off process. A 5 nm HfO_2 switching layer was
420 deposited by atomic layer deposition using water and tetrakis(dimethylamido)hafnium as
421 precursors at 250 °C. Sputter-deposited Ta of 50 nm thickness followed by 10 nm Pd was
422 used in a lift-off process to serve as the top electrode. The filamentary based Ta_2O_5 device
423 consists of a TiN/4nm stoichiometric Ta_2O_5/20 nm nonstoichiometric TaO_x/10 nm TaN/TiN
424 stack with a cross-sectional area of 75 nm × 75 nm, while the non-filamentary-based aVMCO
425 has a cross-sectional area of 135 nm × 135 nm and is composed of a TiN/8 nm amorphous-
426 Si/8 nm anatase TiO_2/TiN stack. Ta_2O_5 and aVMCO devices were fabricated by imec. The
427 detailed fabrication process parameters can be found in references [11, 28, 29] for Ta/HfO_2,
428 Ta_2O_5 and aVMCO RRAMs respectively.

429 The conductance of Ta/HfO_2 devices was modulated by applying SET pulses (500 µs @
430 2.5 V and gate voltage increasing from 0.6 V to 1.6 V). After each of the 11 cycles, RESET
431 pulses were applied (5 µs @ 0.9 V increasing to 2.2 V and gate voltage of 5 V). The voltage
432 was being increased linearly throughout the 100 pulses. All electrical tests for Ta_2O_5 and
433 aVMCO devices were done with a Keysight B1500A. The RTN data is extracted by switching
434 the device into 8 uniformly distributed resistance levels between 25 kΩ and 200 kΩ, and 8
435 nearly uniformly distributed resistance levels between 1 MΩ and 7.5 MΩ with incremental
436 RESET DC sweeps [30] for Ta_2O_5 and aVMCO respectively. RTN measurement is then
437 carried out at each resistance level at a 0.1 V and 3 V read-out for Ta_2O_5 and aVMCO
438 respectively, with a sampling time of 2 ms/point and 10,000 sampling point per resistance
439 level for an RTN measurement period of 20 s.

Simulations

In this work, feed-forward ANNs with fully connected layers and continuous weights were trained to recognise handwritten digits using the MNIST data base. All 60,000 MNIST training images were used during the training stage; training set consisted of 50,000 images and verification set consisted of 10,000 images. All 10,000 test images were used to evaluate the inference accuracy of ANNs. Networks used 784 input neurons representing pixel intensities of MNIST images of 28×28 pixel size, as well as one bias neuron. 10 output neurons were used; they represented the ANNs' predictions of 10 handwritten digits. Hidden layers used sigmoid activation function, while the output layer used softmax activation function. Weights were optimised by minimising cross-entropy error function using stochastic gradient descent. Learning rate of 0.01 and patience of 25 epochs were used. 25 networks were trained for each architecture explored by initialising them differently. When numerically optimising ANNs' weightings, optimisation was performed by employing verification set, while the performance was evaluated using the test set. The code was implemented in Python.

Weights were mapped onto pairs of memristors' conductances using proportional mapping scheme—synaptic weights were made proportional to one of the conductances in the pair, while the other was left unelectroformed. The zero weight was interpreted as given—in practise, it would be implemented by not electroforming the device, thus resulting in its negligible conductance. Although aVMCO devices do not have electroforming stage, for consistency we assumed that additional insulating circuit elements could be used to implement the zero weight. Negative weights would be implemented by placing certain memristors in dedicated bit lines of the crossbars whose outputs would be subtracted from the outputs at the corresponding bit lines implementing positive weights. Maximum weights after mapping were optimised separately for each set of network architecture and conductance levels; in each case this was done by excluding a certain proportion, p_{L}, of weights with largest absolute values. What p_{L} values were used for each simulation is summarised in Supplementary Table SI. More details on the mapping procedure can be found in our past work [20].

All non-idealities, except for line resistance, were simulated by disturbing the individual conductances of memristor-based ANNs. To investigate line resistance, nodal analysis was employed. By setting up simultaneous linear equations using Ohm's law and Kirchhoff's current law, those were solved in sparse matrix representation using Python's library scipy.

471 After simulating memristor non-idealities, committees of different ANNs were composed. 472 Committees used EA, i.e. the outputs of individual networks in a committee were averaged 473 to produce a single output vector. In EA, the output vectors of individual networks can 474 simply be added together (if the weightings of different networks are the same, as we assume 475 in the main text); the label corresponding to the entry with the highest value would be 476 the prediction of the committee. This addition can be performed either in software, or, if 477 the activation function of the last neuronal layer can be implemented physically, it can be 478 performed by adding corresponding currents produced by the circuitry of this activation 479 function.

480 In the simulations, neural networks that go into a committee were chosen randomly. 481 This was done to reflect the most convenient strategy when manufacturing such systems— 482 because one does not need to selectively choose the networks, manufactured crossbars can be 483 easily programmed without the need to replace them if they perform poorly when working 484 individually (unless their effect is so detrimental that they have to be ignored which can 485 be made possible with this technique). Besides, devices might change over time, thus these 486 simulations, which show what happens when one does not selectively choose the networks, 487 are valuable to investigate conditions where it is not possible to replace the networks.

488 In the simulations, 25 base networks were used (each having different set of weights) for 489 each of the architectures. Then all of their weights were mapped onto pairs of conductances 490 using HRS/LRS values extracted from experiments. Finally, to reflect the effect of each of 491 the non-idealities, all networks were disturbed multiple times. In each disturbance iteration, 492 multiple combinations of networks were chosen and their performance as a committee of 493 certain size was evaluated. In total, for most simulations, 10,000 data points were recorded 494 for a committee of every size—these data captured the variations of base networks, their 495 combinations and different disturbance iterations. Only simulations involving line resistance 496 or numerical optimisation of weights had fewer data points for some committee sizes (due 497 to long simulation times).

498 DATA AVAILABILITY

499 The data that support the findings of this study are available from the corresponding 500 author upon reasonable request.

AUTHOR CONTRIBUTIONS

A.M. and D.J. conceived the idea and designed the study. A.M., P.F. and Z.C. performed the experimental measurements. D.J. performed the simulations and analysed the experimental and simulation results. C.L. and Q.X. provided the experimental data of the programming of a Ta/HfO_2 1T1R RRAM array. A.M., W.D.Z. and A.J.K. supervised the research. D.J. wrote the initial manuscript. All authors contributed to the discussions of the results and improved the text.

COMPETING INTERESTS STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

FUNDING

A.M. acknowledges funding from the Royal Academy of Engineering under the Research Fellowship scheme, A.J.K. acknowledges funding from the Engineering and Physical Sciences Research Council (EP/P013503/1) and the Leverhulme Trust (RPG-2016-135), W.D.Z. acknowledges funding from the Engineering and Physical Sciences Research Council (EP/S000259/1).

---

[1] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," *arXiv preprint arXiv:1906.02243*, 2019.

[2] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *International Conference on Learning Representations*, 2016, San Juan (Puerto Rico), arXiv preprint arXiv:1510.00149.

[3] C. Li, Z. Wang, M. Rao, D. Belkin, W. Song, H. Jiang, P. Yan, Y. Li, P. Lin, M. Hu, N. Ge, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Long short-term memory networks in memristor crossbar arrays," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 49–57, 2019, doi: 10.1038/s42256-018-0001-4.

[4] Z. Wang, C. Li, W. Song, M. Rao, D. Belkin, Y. Li, P. Yan, H. Jiang, P. Lin, M. Hu, J. P. Strachan, N. Ge, M. Barnell, Q. Wu, A. G. Barto, Q. Qiu, R. S. Williams, Q. Xia, and J. J. Yang, "Reinforcement learning with analogue memristor arrays," *Nature Electronics*, vol. 2, no. 3, p. 115, 2019, doi: 10.1038/s41928-019-0221-6.

[5] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, and D. Ielmini, "Solving matrix equations in one step with cross-point resistive arrays," *Proceedings of the National Academy of Sciences*, vol. 116, no. 10, pp. 4123–4128, 2019, doi: 10.1073/pnas.1815682116.

[6] S. R. Nandakumar, M. Le Gallo, I. Boybat, B. Rajendran, A. Sebastian, and E. Eleftheriou, "A phase-change memory model for neuromorphic computing," *Journal of Applied Physics*, vol. 124, no. 15, p. 152135, 2018, doi: 10.1063/1.5042408.

[7] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. D. Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. P. Farinha, B. Killeen, C. Cheng, Y. Jaoudi, and G. W. Burr, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, 2018, doi: 10.1038/s41586-018-0180-5.

[8] 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," in *International Electron Devices Meeting*. IEEE, 2016, San Francisco (United States), doi: 10.1109/IEDM.2016.7838429.

[9] J. Woo, K. Moon, J. Song, S. Lee, M. Kwak, J. Park, and H. Hwang, "Improved synaptic behavior under identical pulses using AlO_x/HfO_2 bilayer RRAM array for neuromor-

546 phic systems," *IEEE Electron Device Letters*, vol. 37, no. 8, pp. 994–997, 2016, doi:
547 10.1109/LED.2016.2582859.

548 [10] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B.
549 Strukov, "Training and operation of an integrated neuromorphic network based on metal-
550 oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015, doi: 10.1038/nature14441.

551 [11] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang,
552 W. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Ef-
553 ficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature*
554 *communications*, vol. 9, no. 1, p. 2385, 2018, doi: 10.1038/s41467-018-04484-2.

555 [12] A. Chen and M. R. Lin, "Variability of resistive switching memories and its impact on cross-
556 bar array performance," in *2011 International Reliability Physics Symposium*. IEEE, 2011,
557 Monterey (United States), doi: 10.1109/IRPS.2011.5784590.

558 [13] J. Kang, Z. Yu, L. Wu, Y. Fang, Z. Wang, Y. Cai, Z. Ji, J. Zhang, R. Wang, and Y. Yang,
559 "Time-dependent variability in RRAM-based analog neuromorphic system for pattern recogni-
560 tion," in *International Electron Devices Meeting*. IEEE, 2017, San Francisco (United States),
561 doi: 10.1109/IEDM.2017.8268340.

562 [14] L. Xia, W. Huangfu, T. Tang, X. Yin, K. Chakrabarty, Y. Xie, Y. Wang, and H. Yang,
563 "Stuck-at fault tolerance in RRAM computing systems," *IEEE Journal on Emerging and*
564 *Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 102–115, 2017, doi: 10.1109/JET-
565 CAS.2017.2776980.

566 [15] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E.
567 Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, S. Williams, J. Yang,
568 and Q. Xia, "Analogue signal and image processing with large memristor crossbars," *Nature*
569 *Electronics*, vol. 1, no. 1, pp. 52–59, 2018, doi: 10.1038/s41928-017-0002-z.

570 [16] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni,
571 and E. Eleftheriou, "Mixed-precision in-memory computing," *Nature Electronics*, vol. 1, no. 4,
572 p. 246, 2018, doi: 10.1038/s41928-018-0054-8.

573 [17] M. Hu, J. P. Strachan, Z. Li, and S. R. William, "Dot-product engine as computing mem-
574 ory to accelerate machine learning algorithms," in *17th International Symposium on Quality*
575 *Electronic Design*, 2016, Santa Clara (United States), doi: 10.1109/ISQED.2016.7479230.

576 [18] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nature*
577 *materials*, vol. 18, no. 4, p. 309, 2019, doi: 10.1038/s41563-019-0291-x.
578 [19] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits,"
579 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist
580 [20] A. Mehonic, D. Joksas, W. H. Ng, M. Buckwell, and A. J. Kenyon, "Simulation of inference
581 accuracy using realistic RRAM devices," *Frontiers in Neuroscience*, vol. 13, p. 593, 2019, doi:
582 10.3389/fnins.2019.00593.
583 [21] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid
584 neural networks," in *Artificial Neural Networks for Speech and Vision*. Chapman and Hall,
585 1993, pp. 126–142.
586 [22] S. Hashem and B. Schmeiser, "Improving model accuracy using optimal linear combinations of
587 trained neural networks," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 792–794,
588 1995, doi: 10.1109/72.377990.
589 [23] B. Li, L. Xia, P. Gu, Y. Wang, and H. Yang, "Merging the interface: Power, area and accuracy
590 co-optimization for RRAM crossbar-based mixed-signal computing system," in *Proceedings of*
591 *the 52nd Annual Design Automation Conference*, 2015, San Francisco (United States), doi:
592 10.1145/2744769.2744870.
593 [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional
594 neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105,
595 Lake Tahoe (United States), doi: 10.1145/3065386.
596 [25] Z. Wang, C. Li, P. Lin, M. Rao, Y. Nie, W. Song, Q. Qiu, Y. Li, P. Yan, J. P. Strachan,
597 N. Ge, N. McDonald, Q. Wu, M. Hu, H. Wu, R. S. Williams, Q. Xia, and J. J. Yang, "In situ
598 training of feed-forward and recurrent convolutional memristor networks," *Nature Machine*
599 *Intelligence*, vol. 1, no. 9, pp. 434–442, 2019, doi: 10.1038/s42256-019-0089-1.
600 [26] H. Jiang, L. Han, P. Lin, Z. Wang, M. H. Jang, Q. Wu, M. Barnell, J. J. Yang, H. L. Xin, and
601 Q. Xia, "Sub-10 nm ta channel responsible for superior performance of a HfO_2 memristor,"
602 *Scientific reports*, vol. 6, p. 28525, 2016, doi: 10.1038/srep28525.
603 [27] G. W. Burr, R. M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat, R. S. Shenoy,
604 P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. Hwang, "Experimen-
605 tal demonstration and tolerancing of a large-scale neural network (165 000 synapses) using
606 phase-change memory as the synaptic weight element," *IEEE Transactions on Electron De-*

607 *vices*, vol. 62, no. 11, pp. 3498–3507, 2015, doi: 10.1109/TED.2015.2439635.

608 [28] Y. Fan, L. Zhang, D. Crotti, T. Witters, M. Jurczak, and B. Govoreanu, "Direct evidence
609 of the overshoot suppression in Ta_2O_5-based resistive switching memory with an integrated
610 access resistor," *IEEE Electron Device Letters*, vol. 36, no. 10, pp. 1027–1029, 2015, doi:
611 10.1109/LED.2015.2470081.

612 [29] B. Govoreanu, D. Crotti, S. Subhechha, L. Zhang, Y. Chen, S. Clima, V. Paraschiv, H. Hody,
613 C. Adelmann, M. Popovici, O. Richard, and M. Jurczak, "A-VMCO: A novel forming-free, self-
614 rectifying, analog memory cell with low-current operation, nonfilamentary switching and excel-
615 lent variability," in *Symposium on VLSI Technology*, 2015, Kyoto (Japan), doi: 10.1109/VL-
616 SIT.2015.7223717.

617 [30] Z. Chai, W. Zhang, P. Freitas, F. Hatem, J. F. Zhang, J. Marsland, B. Govoreanu, L. Goux,
618 G. S. Kar, S. Hall, P. Chalker, and J. Robertson, "The over-reset phenomenon in Ta_2O_5
619 RRAM device investigated by the RTN-based defect probing technique," *IEEE Electron Device*
620 *Letters*, vol. 39, no. 7, pp. 955–958, 2018, doi: 10.1109/LED.2018.2833149.

621 [31] C. Sung, S. Lim, H. Kim, T. Kim, K. Moon, J. Song, J.-J. Kim, and H. Hwang, "Effect
622 of conductance linearity and multi-level cell characteristics of TaO_x-based synapse device on
623 pattern recognition accuracy of neuromorphic system," *Nanotechnology*, vol. 29, no. 11, p.
624 115203, 2018, doi: 10.1088/1361-6528/aaa733.

625 [32] Y. Fang, Z. Yu, Z. Wang, T. Zhang, Y. Yang, Y. Cai, and R. Huang, "Improvement of HfO_x-
626 based RRAM device variation by inserting ALD TiN buffer layer," *IEEE Electron Device*
627 *Letters*, vol. 39, no. 6, pp. 819–822, 2018, doi: 10.1109/LED.2018.2831698.

628 [33] B. Govoreanu, A. Redolfi, L. Zhang, C. Adelmann, M. Popovici, S. Clima, H. Hody,
629 V. Paraschiv, I. Radu, A. Franquet, J. C. Liu, J. Swerts, O. Richard, H. Bender, L. Altimime,
630 and M. Jurczak, "Vacancy-modulated conductive oxide resistive RAM (VMCO-RRAM): An
631 area-scalable switching current, self-compliant, highly nonlinear and wide on/off-window re-
632 sistive switching cell," in *International Electron Devices Meeting*. IEEE, 2013, Washington
633 (United States), doi: 10.1109/IEDM.2013.6724599.

634 [34] A. J. Kenyon, M. S. Munde, W. H. Ng, M. Buckwell, D. Joksas, and A. Mehonic, "The
635 interplay between structure and function in redox-based resistance switching," *Faraday Dis-*
636 *cussions*, vol. 213, pp. 151–163, 2019, doi: 10.1039/C8FD00118A.

637 [35] W. Wu, H. Wu, B. Gao, P. Yao, X. Zhang, X. Peng, S. Yu, and H. Qian, “A methodology
638 to improve linearity of analog RRAM for neuromorphic computing,” in *Symposium on VLSI*
639 *Technology*. IEEE, 2018, Honolulu (United States), doi: 10.1109/VLSIT.2018.8510690.
640 [36] Z. Chai, P. Freitas, W. Zhang, F. Hatem, J. F. Zhang, J. Marsland, B. Govoreanu, L. Goux,
641 and G. S. Kar, “Impact of RTN on pattern recognition accuracy of RRAM-based synaptic
642 neural network,” *IEEE Electron Device Letters*, vol. 39, no. 11, pp. 1652–1655, 2018, doi:
643 10.1109/LED.2018.2869072.

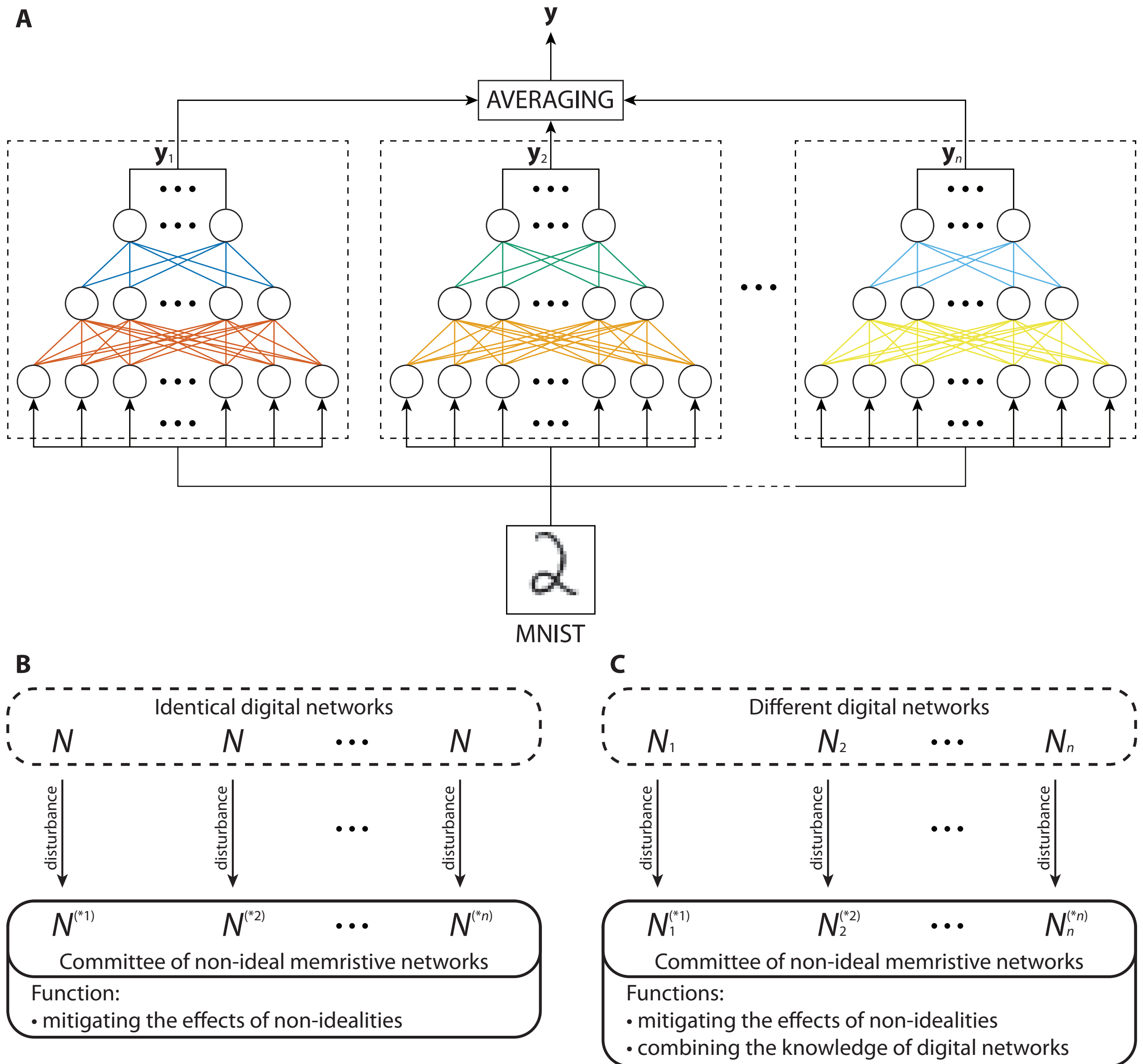

Figure 1. Using multiple neural networks to improve inference accuracy. **A**) The principle of EA. **B**) Using identical digital networks when implementing committees of memristive neural networks only helps to deal with the damage to the networks caused by the non-idealities. **C**) Using different digital networks when implementing committees of memristive neural networks both helps to deal with the damage to the networks caused by the non-idealities and allows to combine the knowledge about the data set acquired by individual digital networks.

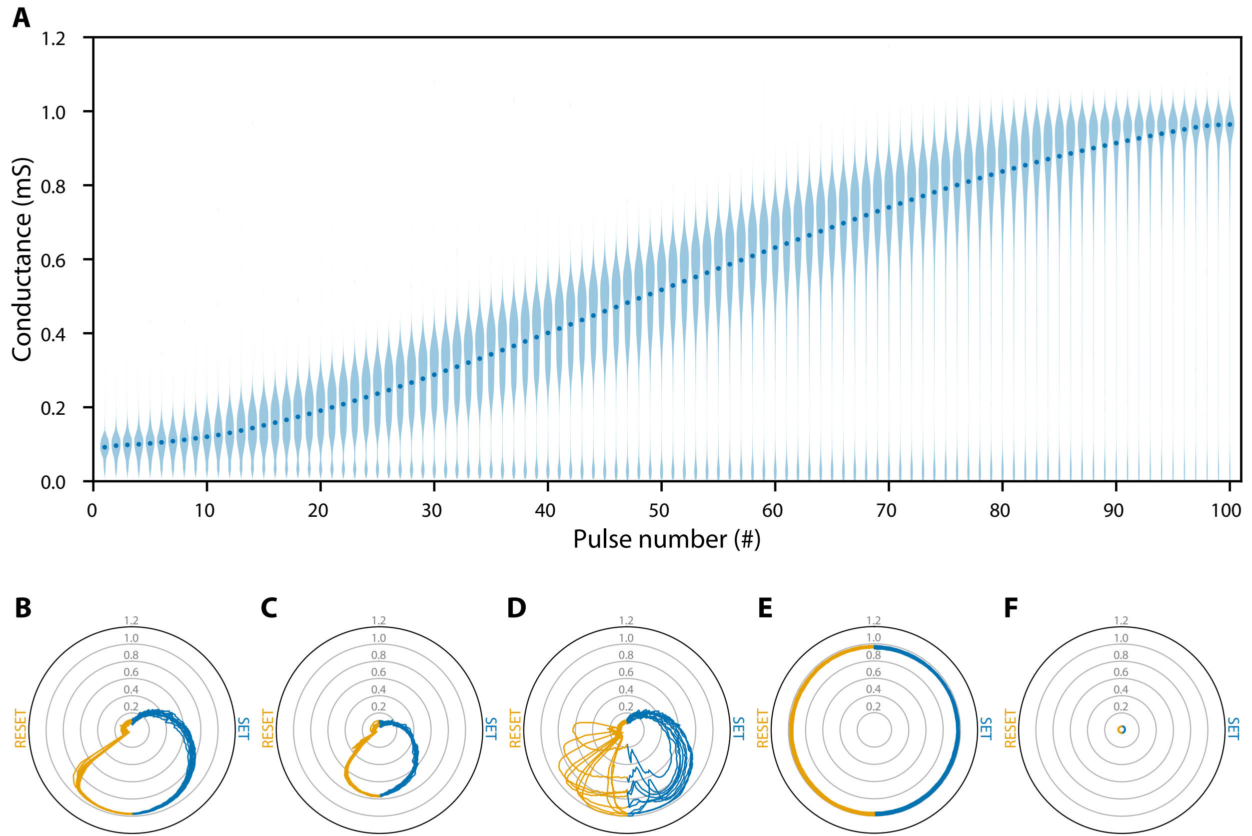

Figure 2. Experimental data of Ta/HfO_2 RRAM crossbar array of shape 128×64. **A**) Modulation of devices' conductance over 11 SET cycles, each consisting of a 100 potentiating pulses. Violin plots of gradual conductance changes are shown for all Ta/HfO_2 devices, with dots representing median conductance after a certain number of pulses. 100 points were used for Gaussian kernel density estimation. All violin plots have their maximum widths normalised. **B**-**F**) Examples of devices with their conductance (in mS) **B**) spanning the full range, **C**) spanning part of the full range, **D**) exhibiting cycle-to-cycle variability, **E**) stuck at high values, **F**) stuck at low values. These diagrams show conductance of five devices from Ta/HfO_2 crossbar array over 11 SET and RESET cycles. The radial component represents the conductance, while the angular component represents the number of applied pulses. The first SET cycle starts at the top of each of the diagrams. The conductance (in blue) over 100 SET pulses is displayed in a clockwise fashion across the right half of each of the diagrams. Following that, conductance (in orange) over 100 RESET pulses (starting at the bottom) is displayed across the left half of each of the diagrams, after which the next cycle is displayed. Cartesian version of these plots is shown in Supplementary Figure S9.

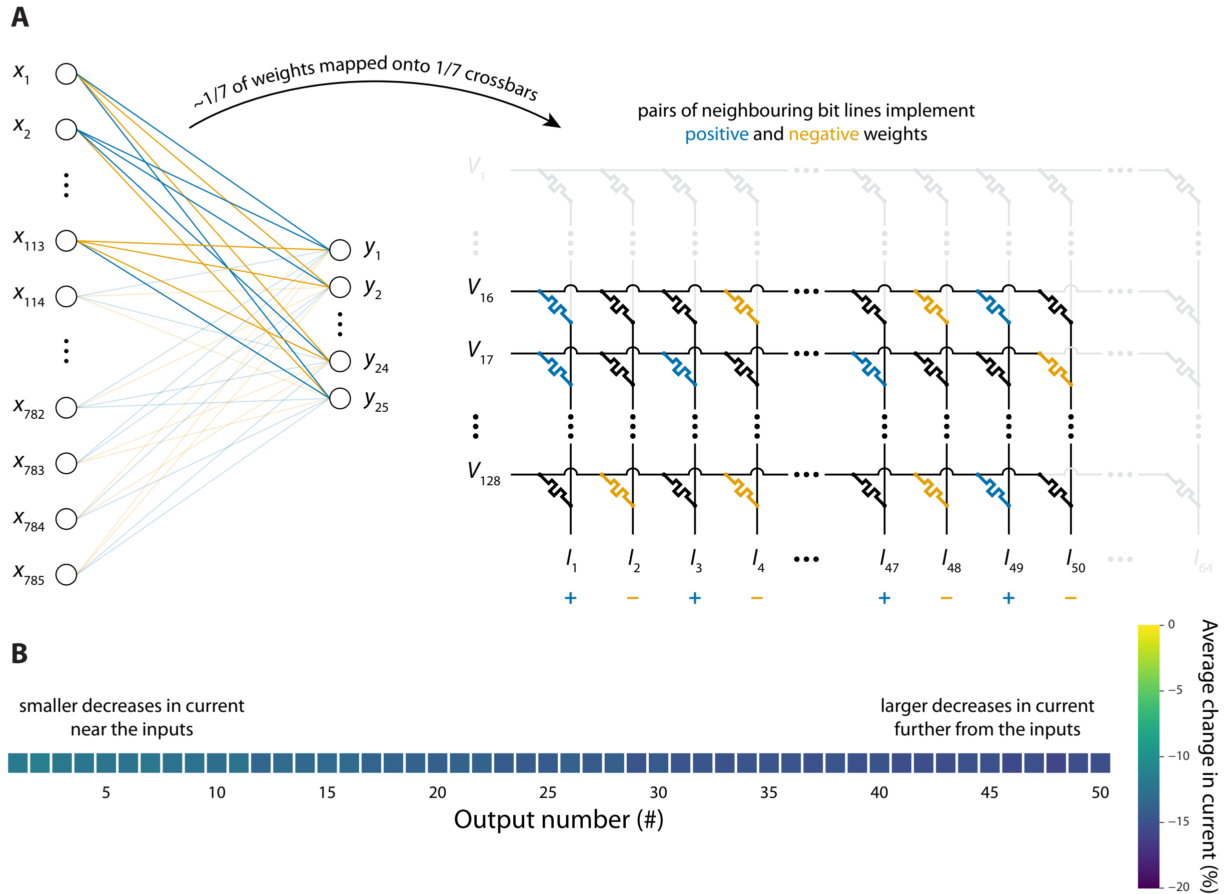

Figure 3. Theoretical implementation of a synaptic layer of shape 785 × 25 using crossbars of shape 128 × 64. **A**) Mapping the first subset of weights onto one of the seven crossbars used to implement the whole synaptic layer. Positive weights and negative weights are mapped onto memristors in different bit lines. **B**) Heatmap of average changes in output currents due to line resistance (in all seven Ta/HfO_2 crossbars). For this particular simulation, it was assumed that Ta/HfO_2 devices can be programmed perfectly.

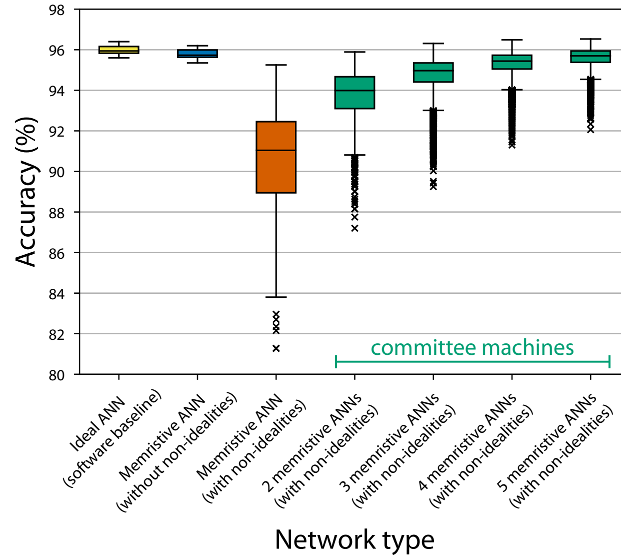

Figure 4. Accuracy achieved by individual networks and their committees when faulty devices, D2D variability data and line resistance of Ta/HfO_2 crossbar are taken into account. The maximum whisker length is set to 1.5 × IQR.

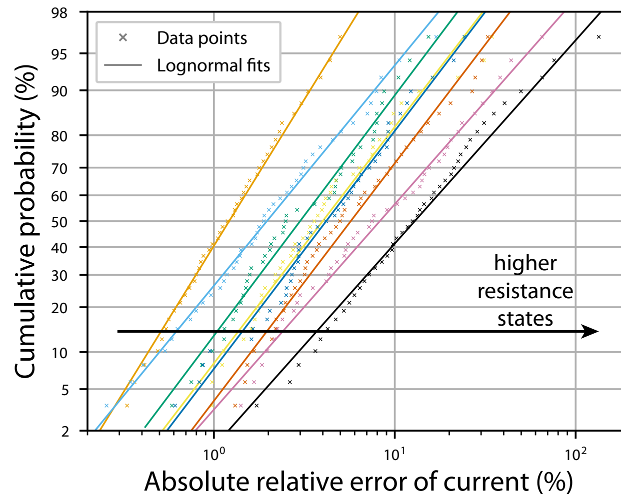

Figure 5. Cumulative probability plots of RTN-induced relative current deviations for all 8 resistance states of a Ta_2O_5 RRAM device. Lognormal fits are shown for each resistance state.

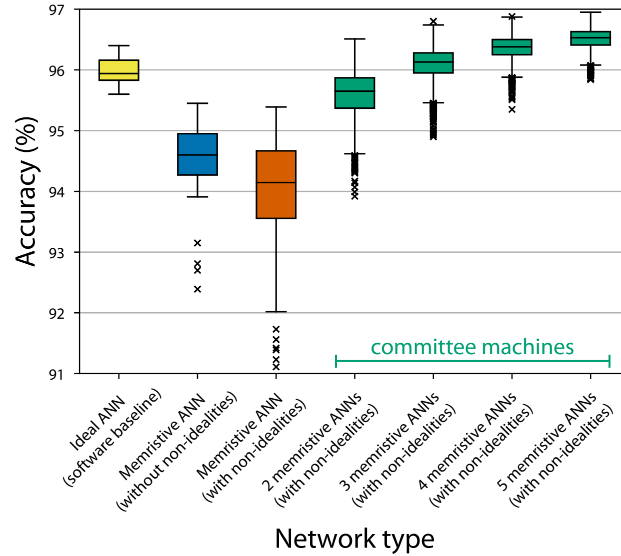

Figure 6. Accuracy achieved by individual networks and their committees when RTN data of a Ta_2O_5 device are taken into account. Additionally, interconnect resistance of $0.35\,\Omega$ and $0.32\,\Omega$ in the word and bit lines, respectively, (from Ta/HfO_2 array) was used to include line resistance effects. The maximum whisker length is set to $1.5 \times$ IQR.

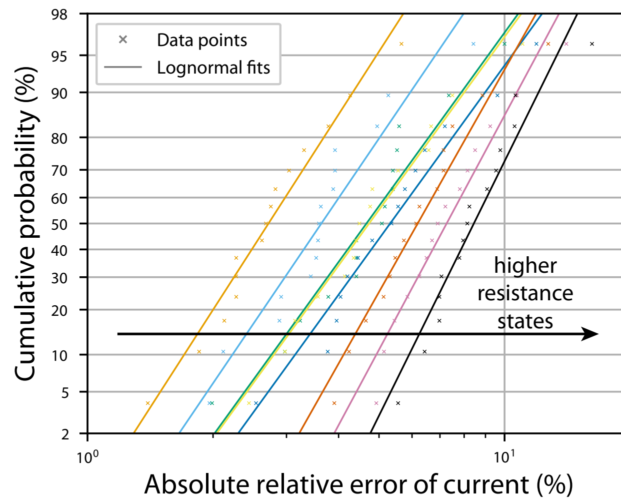

Figure 7. Cumulative probability plots of RTN-induced relative current deviations for all 8 resistance states of aVMCO RRAM device. Lognormal fits are shown for each resistance state.

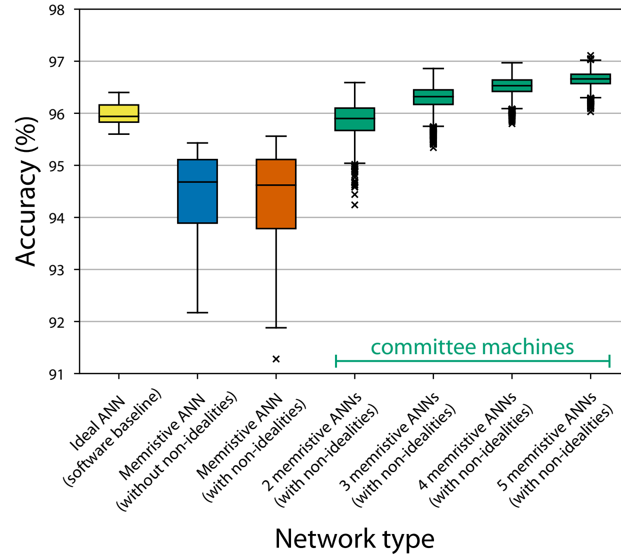

Figure 8. Accuracy achieved by individual networks and their committees when RTN data of an aVMCO device are taken into account. Additionally, interconnect resistance of $0.35\,\Omega$ and $0.32\,\Omega$ in the word and bit lines, respectively, (from Ta/HfO_2 array) was used to include line resistance effects. The maximum whisker length is set to $1.5 \times$ IQR.

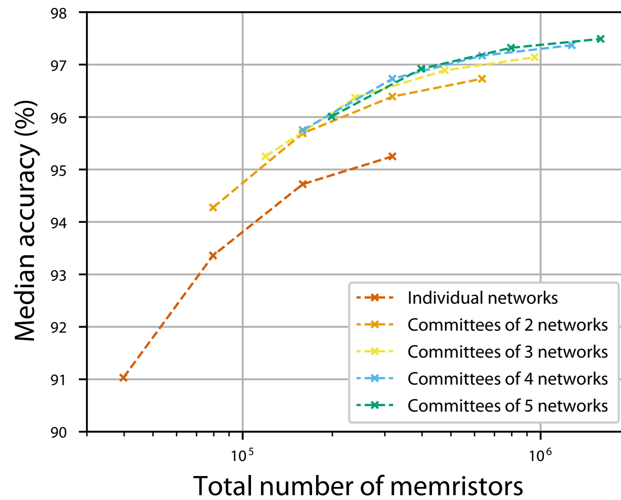

Figure 9. Median accuracy achieved by individual one-hidden-layer memristor-based networks and their committees, when controlled for total number of memristors required. The networks contained 25, 50, 100 or 200 hidden neurons and were disturbed using faulty devices and D2D variability data from Ta/HfO_2 crossbar.

645 **TABLES**

First author (year)	Non-ideality	Device type	Proposed solution
C. Sung (2018) [31]	Current/voltage non-linearity	TaO_x RRAM	Hot-forming step is adopted
C. Li (2018) [15]	Current/voltage non-linearity	Ta/HfO_2 RRAM	1T1R architecture is adopted
Y. Fang (2018) [32]	Device-to-device variability	HfO_x RRAM	Ultra-thin ALD-TiN buffer layer is introduced
B. Govoreanu (2013) [33]	Device-to-device variability	Al_2O_3/TiO_2 (VMCO) RRAM	Non-filamentary RRAM is adopted
A. J. Kenyon (2019) [34]	Device-to-device variability	SiO_x RRAM	The roughness of bottom electrodes is increased
L. Xia (2017) [14]	Faulty devices	-	A modified mapping algorithm and redundancy schemes are used
S. Ambrogio (2018) [7]	Limited dynamic range	PCM	Two pairs of conductance of varying significance for every synaptic weight are used
M. Hu (2016) [17]	Line resistance	-	Advanced mapping algorithms are used to compensate for line resistance effects
W. Wu (2018) [35]	Programming non-linearity	HfO_x RRAM	Electro-thermal modulation layer is deposited on the switching layer
J. Woo (2016) [9]	Programming non-linearity	HfO_2 RRAM	Bilayer structure is adopted
S. Ambrogio (2018) [7]	Programming non-linearity	PCM	PCM devices are used together with CMOS transistors
Z. Chai (2018) [36]	Random telegraph noise	TiO_2/a-Si (aVMCO) RRAM	Non-filamentary RRAM is adopted

Table I. Examples of past efforts at dealing with non-idealities of memristive devices and their systems.

Committee Machines—A Universal Method to Deal with Non-Idealities in Memristor-Based Neural Networks

D. Joksas[1], P. Freitas[2], Z. Chai[2], W. H. Ng[1], M. Buckwell[1],
C. Li[3], W. D. Zhang[2], Q. Xia[3], A. J. Kenyon[1], and A. Mehonic[1]

[1]*Department of Electronic and Electrical Engineering,*
University College London, London (United Kingdom)

[2]*Department of Electronics and Electrical Engineering,*
Liverpool John Moores University, Liverpool (United Kingdom)

[3]*Department of Electrical and Computer Engineering,*
University of Massachusetts Amherst (United States of America)

Abstract

Artificial neural networks are notoriously power- and time-consuming when implemented on conventional von Neumann computing systems. ~~Recent~~ Consequently, recent years have seen an emergence of research in machine learning hardware that strives to ~~break the bottleneck of von Neumann architecture and optimise the data flow, namely, to~~ bring memory and computing closer together. ~~One of the most often suggested solutions is the physical implementation of~~ A popular approach is to realise artificial neural networks in ~~which~~ hardware by implementing their synaptic weights ~~are realised with memristive devices, such as resistive random-access memory~~using memristive devices. However, various device- and system-level non-idealities usually prevent these physical implementations from achieving high inference accuracy. We suggest applying a well-known concept in computer science—committee ~~machine—in~~ machines—in the context of memristor-based neural networks. Using simulations and experimental data from three different types of memristive devices, we show that committee machines employing ensemble averaging can successfully increase inference accuracy in physically implemented neural networks that suffer from faulty devices, device-to-device variability, random telegraph noise and line resistance. Importantly, we ~~show~~ demonstrate that the accuracy can be improved even without increasing the total number of memristors.

I. INTRODUCTION

Artificial neural networks (ANNs), with all of their variants, are now the main tools in machine learning tasks, such as classification. The vast amounts of data being constantly produced have enabled successful training and operation of ANNs. However, to achieve high inference accuracy, it is usually necessary for neural networks to have a large number of parameters. This results in both training [1] and inference [2] stages being time- and power-consuming. This is largely caused by the need to transfer data from memory to computing units—physical separation of memory and computing is the essence of any von Neumann system.

One of the most promising solutions to these problems is the paradigm of non-von Neumann computing and, specifically, analogue implementations of synapses (weights) in physical ANNs. Because there are many more synapses than there are neurons in ANNs, the matrix-vector multiplications, in which the synaptic weight values are used, are the costliest operations in these networks, both in terms of power and time. Computing directly in memory would minimise ~~costly~~ data transfers from off-chip memory, thus the most popular approach is using analogue memory devices as proxies for synaptic weights of ANNs (both fully connected and their variants [3, 4]). A common technique is to arrange such devices in a structure, called crossbar array, in which every device (or a pair of devices) is used to represent a single synaptic weight or, more generally, an entry in a matrix [5]. Memristive devices, such as phase-change memories (PCMs) [6, 7] or resistive random-access memories (RRAMs) [8, 9], have been considered as candidates for such tasks. Although here we focus on ex-situ training, such systems have been successfully utilised for in-situ training too [10, 11].

In memristive implementations of ANNs, the main concern is that various non-idealities associated with these devices can prevent these systems from achieving high accuracy [12, 13]. Examples of non-idealities affecting inference accuracy include, but are not limited to, devices not being able to electroform, devices stuck in one of the resistance states after electroforming, device-to-device (D2D) variability and random telegraph noise (RTN). When training analogue systems in-situ, limited endurance and non-linear resistance modulation too have to be taken into account. To mitigate the effects of these device non-idealities, it is often necessary to modify device structure [9], to use more advanced programming schemes

59 [14] or to use additional circuitry [15] or high-precision processing units [16] in conjunction
60 with memristive elements. On the system level, there is an issue of line resistance which
61 affects the distribution of currents and thus decreases the accuracy. These line resistance
62 effects can be partially compensated for algorithmically [17] or partially mitigated by using
63 multiple smaller crossbar arrays [18]. Examples of past efforts at dealing with these and
64 other non-idealities of memristive devices and systems are listed in Table I; most of these
65 non-idealities are still the main focus of the research in the neuromorphic community.

66 We propose a simple way to mitigate the effects of all types of non-idealities during
67 inference. We suggest combining several non-ideal memristor-based neural networks into
68 committees to achieve better accuracy. The committee machine (CM) method we propose
69 significantly increases the inference accuracy and does not increase the computation time
70 because memristive ANNs in such committees work in parallel.

71 In this work, we firstly explain the simulation setup—what networks were trained,
72 how they were simulated and how they were combined into CMs. After that, follows
73 the experimental part. We investigate three different types of memristor technology—
74 tantalum/hafnium oxide-based (Ta/HfO_2), tantalum oxide-based (Ta_2O_5), and amorphous
75 vacancy modulated conductive oxide-based (aVMCO) devices. By exploring their non-
76 idealities relevant to inference—faulty devices, D2D variability, RTN, and line resistance—
77 we use the experimental data to simulate memristive ANNs working individually and in
78 committees.

79 II. RESULTS

80 A. Simulation setup

81 Fully connected ANNs were trained in software to recognise handwritten digits (using
82 MNIST data base [19]). Architectures with one hidden layer were explored. Unless stated
83 otherwise, the simulations used networks with 25 hidden neurons. However, networks with
84 50, 100 and 200 hidden neurons were additionally employed to evaluate the effectiveness of
85 the proposed method while controlling for the total number of memristors required. Follow-
86 ing training, weights of ANNs were mapped onto pairs of conductances using proportional
87 mapping scheme (see [20]) to simulate memristor-based ANNs. Finally, these memristive

networks were disturbed using experimental data to reflect the effect of device- and system-level non-idealities.

After simulating physical non-idealities, the networks were combined into CMs that employed ensemble averaging (EA) [21]. The principle of EA is shown in Figure 1A—several networks are combined in parallel and then their outputs are averaged. After that, the prediction is made using the averaged vector—the prediction is the label corresponding to the largest entry in the vector.

CM methods are frequently used even with conventional ANNs. Methods, such as EA, often produce better accuracy than that of the best individual network in a committee [22]. Although there are other types of CMs besides EA, they often rely on training additional gating networks or boosting networks during the training stage. Using a gating network in this scenario would produce additional problems—to avoid it acting as a performance bottleneck, it too would have to be implemented on crossbar arrays. Various non-idealities would decrease the effectiveness of this gating network which is responsible for making the decisions about the whole committee of ANNs. Likewise, we speculate that boosting of networks would not be feasible in ex-situ training because it requires information about where individual ANNs perform poorly—this cannot be known precisely until they are implemented physically on crossbar arrays and the non-idealities manifest themselves. To authors' best knowledge, the application of boosting in the context of memristive neural networks seems to have been explored only once before [23]; as expected, it requires training each memristive implementation differently because non-idealities manifest themselves differently in different crossbar arrays.

There exist modifications of EA algorithm that could potentially perform better. One example of this is generalized ensemble method (GEM) which, instead of using equal weightings for each network during averaging (as in EA), uses a different one for each network [21]. These weightings are analytically determined by considering correlation of errors between different networks. But because [21] only considered networks with mean square error loss function (while our networks used cross-entropy loss function), this work does not explore GEM. Instead, we investigated whether it is possible to achieve a better performance by optimising the weightings numerically. This method, like GEM and others previously mentioned, might be impractical because, firstly, these weightings could be determined only after the ANNs are physically implemented on crossbars, and, secondly, the devices could change

120 throughout their lifetimes thus affecting the optimal weightings.

121 Even with the assumption that the devices would have perfect retention, we found that 122 optimisation of weightings achieves effectively the same performance. Because of these rea- 123 sons, we focus only on EA in the main text, but present our results of optimising weightings 124 in Supplementary Figure S3S5. We stress that we are open to the idea that other CM meth- 125 ods besides EA could be utilised successfully for ex-situ training in the context of memristive 126 ANNs. However, in this work we focus on demonstrating that CMs can be used to improve 127 the accuracy of memristor-based ANNs in general.

128 With EA, we find that even when the memristive ANNs, which go into a committee, all 129 use the same digitally implemented digital weights that are mapped onto crossbar arrays 130 (see Figure 1B), committee of memristor-based networks can still achieve higher accuracy 131 than just a single non-ideal network. Although all networks have the same *digital* weights 132 before mapping, their physical implementations (which we call "disturbances" in Figures 1B, 133 C because they can usually be represented by the modification of individual weights) will 134 be different. For example, in one crossbar array, a certain set of devices will be faulty, while 135 in the other crossbar array, it will be a different set. This will result in different physical 136 implementations having slightly different learned representations of the data set, or, to 137 paraphrase, different networks will be "damaged" differently by the non-idealities. This 138 means that these committees will be able to combine different representations, and thus 139 achieve higher accuracy. However, by definition, such approach would almost certainly not 140 yield a committee accuracy that is higher than the accuracy of a single digitally implemented 141 network.

142 A better approach is to use different digital networks for different physical implementa- 143 tions that go into a committee (see Figure 1C). This approach much more resembles the 144 conventional application of EA in computer science. In the context of memristive crossbar 145 arrays, it would not only help to mitigate the effects of the non-idealities (as in the case 146 of Figure 1B), but would also allow to combine the representations of digital networks that 147 were different even before the mapping stage. Most importantly, this method allows for a 148 committee to achieve higher accuracy which is sometimes even higher than that of individual 149 networks with digitally implemented weights. We thus used this method in this analysis. 150 An example comparison of these two approaches is presented in Supplementary Figure S8.

151 In this work, any given committee used only one network architecture but each network

was initialised differently before training, thus trained networks had different sets of weights. Although it was not explored in this work, combining different network architectures in a committee of memristor-based networks might be advantageous. Furthermore, in this work we focus on fully connected ANNs but CMs could be applied to other variants of neural networks as well. Due to the simplicity of EA, it could, for example, be employed in convolutional neural networks (CNNs) [24], which are often used for image classification. This might be of interest as CNNs have been successfully implemented using crossbar arrays recently [25]. However, crossbar implementations are naturally more suited to fully connected networks, therefore we limit ourselves to this architecture but are open to exploring the effectiveness of EA with memristive CNNs in the future.

B. Ta/HfO_2 RRAM

With array-level data available, Ta/HfO_2 experiments provide the most complete picture of device- and system-level non-idealities. In this subsection, we present not only the analysis of faulty devices and D2D variability, but also careful consideration of the line resistance effects. Ta/HfO_2 memristors do not exhibit apparent RTN and overall have excellent retention properties [26], and thus are perfect candidates for inference application.

1. Faulty devices and device-to-device variability

The most energy-efficient procedure to modulate the conductance of memristors is by the application of voltage pulses. In an ideal scenario, one would apply identical pulses and observe constant increases in conductance with each pulse. This is rarely the case in practise, but, fortunately, this type of behaviour is more relevant for in-situ training where it is necessary to ensure linear adjustment of ANN's weights [27]. In ex-situ training, conductance verification schemes can be used to program the devices precisely. Because the devices would have to be programmed only once, one can spend additional resources to do so accurately by applying SET (potentiation) and RESET (depression) pulses until a desirable conductance state is achieved.

Even with this approach, there remain two obstacles—faulty devices and D2D variability. It is observed in most memristor technologies that at least a small fraction of the devices

tends to get stuck in a particular conductance state. Additionally, even if not stuck, different devices might behave differently; for example, they might have different conductance ranges. Figure 2A shows conductance changes in Ta/HfO_2 RRAM devices (in a 128×64 crossbar array) when they are applied with voltage pulses. We can see from the median values that overall the devices' conductance tends to increase as more SET pulses are applied. However, the wider bottom regions of the violin plots indicate that some devices are stuck around high resistance state (HRS) and cannot set entirely no matter how many voltage pulses are applied. There also exist devices that are stuck in low resistance state (LRS), or simply do not span the full conductance range.

Figure 2A combines data from multiple SET cycles for each of the memristors, thus it is important to understand how do these devices behave individually. Figures 2B-F show conductance of 5 (out of 8,192) devices over 11 SET and RESET cycles. In the five diagrams, the radial component represents the conductance (in mS) and the angular component represents the number of applied pulses. Figure 2B shows an example of preferable (and typical) device behaviour—conductance changes in a continuous fashion and spans a wide range of conductance values, from $\sim$0.1 ms to $\sim$1.0 ms. Although RESET cycles tend to feature abrupt decreases in conductance, one can always repeat a cycle and exploit the more predictable behaviour of SET cycles.

When encoding continuous numbers into crossbar devices' conductances, it is often preferable to choose a large enough conductance range. Using data from Figure 2A, one could, for example, choose the range between the first and the last median points (from $\sim$0.1 mS to $\sim$1.0 mS). Device, whose behaviour is presented in Figure 2B, could be easily set to any conductance within that range, as we have seen before. On the other hand, device, whose behaviour is presented in Figure 2C, although operating in a predictable fashion, has smaller conductance range. We can see that in all cycles, its conductance does not exceed 0.8 mS. This is an example of D2D variability that can make it difficult to choose optimal operating range and set the conductance of all devices precisely.

Device, whose behaviour is presented in Figure 2D, shows high cycle-to-cycle variability. Although that could prove to be a problem in some applications, this specific device might perfectly serve its purpose in ex-situ training of ANNs. We can observe that this device spans the same conductance range as device from Figure 2B, even if in an unpredictable manner. Because all states in the full range are, in theory, achievable, one can cycle the

device multiple times until it is set to the required conductance level.

Lastly, we have devices whose negative effect is most difficult to mitigate—faulty devices. Figure 2E shows behaviour of a device stuck at high conductance values, while Figure 2F shows behaviour of a device stuck at low conductance values. No matter how many pulses the devices are applied with or how many times they are cycled, they exhibit almost no conductance variation and thus, in most cases, cannot be used to encode information.

Knowing that some devices perform like the ones whose behaviour is shown in Figures 2C,E,F, it is important to minimise their negative effect. If the conductance that a device has to be set to is outside that device's range, it is sensible to set it to the closest achievable conductance. Although there is little that can be done about fully stuck memristors, it is possible to optimise the behaviour of devices like the one in Figure 2C that simply have smaller conductance range. For example, if such a device has to be set to 0.9 mS, one would set it to the highest achievable conductance ($\sim$0.8 mS). In the following simulations involving faulty devices and D2D variability, operating range between the first and the last median points was used, the devices were chosen randomly from the 128×64 crossbar and set to the most desirable states, as described in this paragraph.

2. Line resistance

The effect of line resistance can be extremely detrimental in many crossbar-based implementations of ANNs. That is especially the case if the crossbars used are large and the resistance of the interconnects ~~are large~~ is high (compared to memristors' resistance). Because in a neural network many of the inputs are non-zero at any given time, a lot of current accumulates in the bit lines which results in significant voltage drops across the interconnects, and thus the current distribution across the crossbar is affected in a major way.

~~Although there are many possible options for how to map synaptic weights onto crossbar arrays, the choice can determine the role of line resistance. It is often the case that synaptic layers of ANNs are large in size. However, that does not mean that the weights in those layers have to be mapped onto crossbars of equivalent shape; not only is that sometimes impossible, but it can also amplify the effect of line resistance. For example, if a synaptic layer with 785 input neurons (as is the case with the first layer of our ANNs) was mapped~~

~~onto a crossbar with 785 word lines, massive amounts of current would accumulate in the bit lines.~~

The Ta/HfO_2 crossbar has shape 128×64 and so this shape was chosen for all the simulations involving line resistance. Even relatively small ANNs of architecture 784(+1):25(+1):10 would need $2 \times (785 \times 25 + 26 \times 10) = 39,770$ memristors to be implemented. Even if not all the inputs were used at any given time, it would not be possible to fit all the memristors onto a single crossbar of shape 128×64. To overcome this, we decided to simulate multiple crossbars, each of which would implement a subset of the synaptic weights, but, for a given synaptic layer, would all compute in parallel. Because $\lceil 785/128 \rceil = 7$, seven crossbars were used to implement the first synaptic layer; the first ~~six crossbars utilised all 128~~ crossbar utilized bottom 113 word lines, while the ~~last one used only the bottom 17~~ other six crossbars used bottom 112 word lines because ~~$785 - 6 \times 128 = 17$~~$113 + 6 \times 112 = 785$. The second synaptic layer was implemented using eighth crossbar ~~utilising~~ utilizing its bottom 26 word lines.

Figure 3A shows an example of how the first synaptic layer of 784(+1):25(+1):10 neural network could be implemented. Specifically, it shows how the first subset of weights would be implemented using one of the crossbars. Because we use proportional mapping scheme, positive and negative weights would be implemented in different bit lines. In Figure 3A, memristors designated to implement positive weights are coloured in blue, memristors designated to implement negative weights are coloured in orange and unelectroformed memristors are coloured in black. Because simulations were constrained by experimental data, ~~the rightmost bit lines are~~ some of the devices were left unused and assumed to ~~contain only unelectroformeddevices~~be unelectroformed. In practise, the crossbars could be manufactured to fit the geometry of the ANNs.

In each synaptic layer, the corresponding output currents from each of the crossbars would be added together. Additionally, output currents at the bit lines implementing negative weights would be subtracted from the output currents at the ~~corresponding bit lines~~ neighbouring bit lines (to their left) implementing positive weights. For example, in the example configuration of Figure 3A, output current at the ~~26th~~ 2nd bit line would be subtracted from the output current at the 1st bit line, etc.

Unfortunately, even when using multiple smaller crossbars, the interconnects can significantly disturb current distribution in the crossbar. Average output current decreases due

to line resistance in all seven crossbars of Ta/HfO_2 devices (whose resistance ranges from ∼1 kΩ to ∼11 kΩ, and their interconnect resistance is ~~0.3 Ω~~0.35 Ω and 0.32 Ω in the word and bit lines, respectively), are shown in the ~~top heatmap of~~ heatmap in Figure 3B. We can see that the current decreases can range from ∼~~15~~12% at the outputs nearest to the applied voltages to ∼~~18~~16% at the outputs in the rightmost bit lines that are used. ~~Such large current decreases often result from large input voltages that are applied at the top part of the crossbar, far away from the outputs. Such inputs generate large amounts of current that flow through large portions of the bit lines and, with voltage drops across interconnects, disturb the overall current distribution in a major way.~~

~~In some applications, such as supervised learning, it might be possible to strategically map certain inputs to certain word lines, so that the effect of line resistance is minimised. We propose intensity-aware reordering of ANN's inputs in which we record the average input intensities over training and verification sets, and then map inputs with highest average intensities to the word lines closest to the outputs of a crossbar. This makes it so that most of the current is generated near the outputs, while the currents in the top parts of the bit lines are disturbed minimally. Bottom heatmap in Figure 3B shows average current decreases when using such a scheme with an unseen test set—we observe significantly smaller decreases. Additionally, to make the influence of positive and negative weights (which are affected very differently in the naive mapping of Figure 3A) more equal and to increase the variability between different ANNs in a committee, we suggest random reordering of inputs and outputs. Both intensity-aware and random reordering were used in all the following simulations involving line resistance . The implementation of these methods individually and in combination with each other is explained in more detail in the supplementary information~~In the supplementary information, we provide a possible strategy of mitigating line resistance effects in supervised learning. This scheme was not employed in the simulations described in the main text because we wanted to find out how well the CM method would deal with noticeable line resistance effects.

3. Inference accuracy

Figure 4 shows the accuracy of individual networks, as well as of their committees; memristive ANNs were simulated by taking into account three non-idealities of Ta/HfO_2 crossbar

explored earlier—faulty devices, D2D variability and line resistance. As indicated by the yellow box plot in Figure 4, individual networks implemented digitally achieve ∼95.9% median accuracy. Networks disturbed to reflect the effect of non-idealities achieve ∼~~90.8~~91.0% median accuracy, as indicated by the vermilion box plot. Although that is a substantial drop in accuracy, we see that as more networks are added to the committee, the more the accuracy increases. When 5 networks are used in a committee, median accuracy increases up to ∼~~95.8~~95.7%, as indicated by the rightmost green box plot.

C. Ta_2O_5 RRAM

In order to explore the effectiveness of minimising adverse effects of RTN, we use another memristor technology based on Ta_2O_5. To investigate RTN, measurements from a single device were considered. To simulate line resistance effects, interconnect resistance from Ta/HfO_2 was used and the same crossbar shape was assumed.

1. Random telegraph noise

Memristors often suffer from RTN resulting in a different accuracy at any given instant in time. Ta_2O_5 device was characterised by measuring the current of 8 resistance states multiple times. Figure 5 shows the cumulative probability plots for those resistance states, together with lognormal fits modelling the nature of RTN. One of the things that the figure reveals is that higher resistance states suffer from higher degree of RTN. Fits for every resistance state, together with occurrence rates (see Supplementary Table SII), were used to disturb the weights of ANNs in order to reproduce the effect of RTN.

2. Inference accuracy

The results combining RTN and line resistance effects for Ta_2O_5 device are shown in Figure 6. From the difference in median accuracy between yellow and blue box plots, we can notice that there is a significant drop in accuracy simply due to mapping of weights onto conductances. That is not surprising given that only 8 states were available for mapping. One can also observe that further drop in median accuracy due to non-idealities is not as

severe—it drops to $\sim$~~94.2~~94.1%. The RTN disturbance magnitude is limited to $<100\%$ in most cases, which possibly contributes to its smaller effect on accuracy. Additionally, Ta_2O_5 device has much higher resistance (ranging from 25 kΩ to 200 kΩ), thus line resistance is also less of a concern. When non-ideal networks are combined into committees of 5, the median accuracy jumps to $\sim$96.5%—even higher than the software baseline of individual networks. This reveals additional trend seen in all the simulations performed—the higher the accuracy of the individual non-ideal memristive networks, the higher the accuracy of the committees that they are part of.

D. aVMCO RRAM

Further, we consider a third memristor technology—one based on aVCMO materials. We test the effects of RTN by considering measurements from a single device. Line resistance effects were simulated by using interconnect resistance and shape of Ta/HfO_2 crossbar array.

1. Random telegraph noise

Figure 7 shows the cumulative probability plots for 8 resistance states of an aVMCO device suffering from RTN. Like in Ta_2O_5, we observe that higher resistance states experience RTN of higher magnitude. However, compared to Ta_2O_5, the RTN magnitude is much more predictable. Fits for each of the 8 resistance states, together with occurrence rates (see Supplementary Table SIII), were used to simulate the effect of RTN in aVMCO-based neural networks.

2. Inference accuracy

The results combining RTN and line resistance are shown in Figure 8. As with Ta_2O_5, we see a large drop due to mapping onto conductances—consequence of very few states available for mapping. More interestingly, the accuracy of individual memristor-based networks with and without non-idealities is almost identical. That is because the occurrence rate of RTN in aVMCO device is small and there is a much smaller probability of RTN having large magnitude. Additionally, resistance of aVMCO device is even higher than that of Ta_2O_5

device—it ranges from $1\,\mathrm{M\Omega}$ to $7.5\,\mathrm{M\Omega}$. Therefore, line resistance has even a smaller effect in a hypothetical array of aVMCO devices. Due to median accuracy of individual non-ideal memristor-based networks being higher ($\sim$~~94.7~~94.6%), the median accuracy of committees is higher too—in committees of size 5 it increases to $\sim$~~96.6~~96.7%.

III. DISCUSSION

The results from the previous section suggest that the method of using committee machines to improve the accuracy of memristive neural networks is ~~technology-agnostic~~technology- and non-ideality-agnostic. CMs can mitigate the effects of faulty devices, D2D variability, RTN and line resistance in combination with each other. Although ~~line resistance is more difficult to deal with using committees due to the similar way in which all crossbars of different networks get affected, using random reordering can increase the effectiveness of ensembles of non-ideal memristive networks. In~~ CM method is slightly less effective with large line resistance (see discussion in the supplementary information), in all cases, we observe that the accuracy of individual non-ideal networks largely determines the accuracy of committees. That is consequential because it means that although committees always increase the accuracy, there is still an incentive to optimise the devices and systems that implement these networks—the higher the accuracy of individual networks, the higher the accuracy of the committees.

It is also important to consider whether using larger networks, instead of committees of smaller networks, would yield the same results if the same number of synapses (or memristors) was used in the large network as in the committee of smaller networks. In our previous work we found that the accuracy of networks before disturbance (which we call "starting accuracy") has a huge effect on the robustness to non-idealities—the larger the starting accuracy, the more robust the networks become [20]. One way to achieve higher starting accuracy is to have larger networks, e.g. if we have a network with one hidden layer, we might increase the number of neurons in that hidden layer, which would likely result in higher accuracy after training and thus higher robustness.

Figure 9 shows a comparison of CMs of memristor-based networks disturbed using faulty devices and D2D variability data from Ta/HfO_2 crossbar, when controlled for the total number of memristors that is required to implement them (line resistance was not taken

into account due to long time required to simulate it in large networks). We can observe that committees of two networks, each with 25 hidden neurons, (leftmost data point of the orange curve) achieve ∼0.9% higher median accuracy than individual networks with 50 hidden neurons (second data point from the left in the vermilion curve), despite both requiring almost identical total number of memristors. Committees of two networks, each with 100 hidden neurons, (third data point from the left in the orange curve) achieve ∼1.1% higher median accuracy than individual networks with 200 hidden neurons (rightmost data point in the vermilion curve), even though both require almost the same total number of memristors. Even larger improvement is gained when committees of four networks, each with 50 hidden neurons, (second data point from the left in the blue curve) are used instead—then the accuracy is improved by ∼1.5%, with almost the exact total number of memristors used.

For different non-idealities and even different training schemes of the ANNs, the equivalents of Figure 9 might be different, but there are a few common characteristics in all of them. In all cases, for a given total number of memristors used, there is an optimal number of networks that should be used in a committee. Additionally, we observe that the more severe a non-ideality is, the more apparent the effectiveness of committees becomes. Finally, sometimes the committees (for a fixed total number of memristors) might achieve lower accuracy than individual networks but only if the networks that they replace are very small and the non-ideality is not very detrimental. If the networks that are being replaced with committees of smaller networks, are sufficiently large, the committees will achieve higher accuracy. An example of that is shown in Supplementary Figure ~~S5~~ S7 where aVMCO device is minimally affected by the non-idealities and so the advantage of committees becomes apparent only when replacing larger networks.

The reason why committees work in the context of non-ideal implementations and why they work best when they are used to replace large networks might, to some extent, lie in their training. When it comes to training fully connected networks, their accuracy tends to saturate as more ~~weights~~ parameters are added. Supplementary Figure ~~S2~~ S4 shows that networks with 50 hidden neurons can be trained to achieve significantly higher accuracy than networks with 25 hidden neurons. However, networks with 200 hidden neurons achieve only slightly higher accuracy than networks with 100 hidden neurons. This also means that networks with 200 hidden neurons will be only slightly more robust to non-idealities than

networks with 100 hidden neurons. When such networks are affected by non-idealities, their accuracy drops to similar values but the smaller network can work in a committee with ~~one more network~~other networks, totalling almost the same number of memristors as the large network, but achieving higher accuracy overall. This is the most likely reason why the committees of smaller networks are effective at dealing with non-idealities, especially when replacing large networks.

In addition to the accuracy improvements, committees can provide flexibility in memristive implementations of neural networks. Digital implementations of ANNs have very predictable behaviour due to the precision of digital logic. Analogue implementations, on the other hand, can vary greatly even if they use the same weights before the mapping onto conductances—that is a result of the stochastic nature of memristors that implement these ANNs. The parallel and modular nature of committee machines makes memristive systems much more flexible. For example, if the verification accuracy of one of the ANNs in a memristor-based CM deteriorates below acceptable levels, its outputs could be disabled to ensure higher accuracy of the rest of the committee.

Importantly, this introduced parallelism comes at almost no extra cost. For a fixed total number of memristors, a committee of smaller networks, compared to a large individual network, would only require a few additional output and bias neurons, and an averaging functionality, which could potentially be implemented in hardware. For example, an ANN with 50 hidden neurons would require 846 neurons in total, while a committee of two ANNs, each with 25 hidden neurons (and thus requiring almost the same total number of memristors), would require 857 neurons in total.

In summary, our simulations employing experimental data from three different types of memristive devices show that committee machines employing ensemble averaging can be used to mitigate the effects of device- and system-level non-idealities in memristor-based neural networks. EA allows to achieve higher inference accuracy in physically implemented neural networks that suffer from faulty devices, device-to-device variability, random telegraph noise, and even line resistance. This method is a universal way to deal with the most common non-idealities and is straightforward to implement during the fabrication stage. Increased modularity of these memristive neural network systems will increase not only their inference accuracy, but also their robustness and flexibility, even without the need to sacrifice area. Although some level of non-idealities in memristors is unavoidable, CM method allows us

450 to deal with these on the system level and is agnostic to a particular technology or, to some
451 degree, type of the non-ideality.

452 METHODS

453 Experiments

454 Ta/HfO_2 RRAM 1T1R array consists of NMOS transistors fabricated in a commercial
455 fab (feature size of 2 µm) and Pt/HfO_2/Ta devices. The bottom electrode was deposited by
456 evaporation of 20 nm Pt layer on top of a 2 nm tantalum (Ta) adhesive layer; the electrode
457 was patterned by photolitography and a lift-off process. A 5 nm HfO_2 switching layer was
458 deposited by atomic layer deposition using water and tetrakis(dimethylamido)hafnium as
459 precursors at 250 °C. Sputter-deposited Ta of 50 nm thickness followed by 10 nm Pd was
460 used in a lift-off process to serve as the top electrode. The filamentary based Ta_2O_5 device
461 consists of a TiN/4nm stoichiometric Ta_2O_5/20 nm nonstoichiometric TaO_x/10 nm TaN/TiN
462 stack with a cross-sectional area of 75 nm × 75 nm, while the non-filamentary-based aVMCO
463 has a cross-sectional area of 135 nm × 135 nm and is composed of a TiN/8 nm amorphous-
464 Si/8 nm anatase TiO_2/TiN stack. Ta_2O_5 and aVMCO devices were fabricated by imec. The
465 detailed fabrication process parameters can be found in references [11, 28, 29] for Ta/HfO_2,
466 Ta_2O_5 and aVMCO RRAMs respectively.

467 The conductance of Ta/HfO_2 devices was modulated by applying SET pulses (500 µs @
468 2.5 V and gate voltage increasing from 0.6 V to 1.6 V). After each of the 11 cycles, RESET
469 pulses were applied (5 µs @ 0.9 V increasing to 2.2 V and gate voltage of 5 V). The voltage
470 was being increased linearly throughout the 100 pulses. All electrical tests for Ta_2O_5 and
471 aVMCO devices were done with a Keysight B1500A. The RTN data is extracted by switching
472 the device into 8 uniformly distributed resistance levels between 25 kΩ and 200 kΩ, and 8
473 nearly uniformly distributed resistance levels between 1 MΩ and 7.5 MΩ with incremental
474 RESET DC sweeps [30] for Ta_2O_5 and aVMCO respectively. RTN measurement is then
475 carried out at each resistance level at a 0.1 V and 3 V read-out for Ta_2O_5 and aVMCO
476 respectively, with a sampling time of 2 ms/point and 10,000 sampling point per resistance
477 level for an RTN measurement period of 20 s.

Simulations

In this work, feed-forward ANNs with fully connected layers and continuous weights were trained to recognise handwritten digits using the MNIST data base. All 60,000 MNIST training images were used during the training stage; training set consisted of 50,000 images and verification set consisted of 10,000 images. All 10,000 test images were used to evaluate the inference accuracy of ANNs. Networks used 784 input neurons representing pixel intensities of MNIST images of 28×28 pixel size, as well as one bias neuron. 10 output neurons were used; they represented the ANNs' predictions of 10 handwritten digits. Hidden ~~layer~~ layers used sigmoid activation function, while the output layer used softmax activation function. Weights were optimised by minimising cross-entropy error function using stochastic gradient descent. Learning rate of 0.01 and patience of 25 epochs were used. 25 networks were trained for each architecture explored by initialising them differently. When numerically optimising ANNs' weightings, optimisation was performed by employing verification set, while the performance was evaluated using the test set. The code was implemented in Python.

Weights were mapped onto pairs of memristors' conductances using proportional mapping scheme—synaptic weights were made proportional to one of the conductances in the pair, while the other was left unelectroformed. The zero weight was interpreted as given—in practise, it would be implemented by not electroforming the device, thus resulting in its negligible conductance. Although aVMCO devices do not have electroforming stage, for consistency we assumed that additional insulating circuit elements could be used to implement the zero weight. Negative weights would be implemented by placing certain memristors in dedicated bit lines of the crossbars whose outputs would be subtracted from the outputs at the corresponding bit lines implementing positive weights. Maximum weights after mapping were optimised separately for each set of network architecture and conductance levels; in each case this was done by excluding a certain proportion, p_{L}, of weights with largest absolute values. What p_{L} values were used for each simulation is summarised in Supplementary Table SI. More details on the mapping procedure can be found in our past work [20].

All non-idealities, except for line resistance, were simulated by disturbing the individual conductances of memristor-based ANNs. To investigate line resistance, ~~loop~~ nodal analysis was employed. By setting up simultaneous linear equations using Ohm's law and Kirch-

hoff's current ~~and voltage laws~~law, those were solved in sparse matrix representation using Python's library scipy.

After simulating memristor non-idealities, committees of different ANNs were composed. Committees used EA, i.e. the outputs of individual networks in a committee were averaged to produce a single output vector. In EA, the output vectors of individual networks can simply be added together (if the weightings of different networks are the same, as we assume in the main text); the label corresponding to the entry with the highest value would be the prediction of the committee. This addition can be performed either in software, or, if the activation function of the last neuronal layer can be implemented physically, it can be performed by adding corresponding currents produced by the circuitry of this activation function.

In the simulations, neural networks that go into a committee were chosen randomly. This was done to reflect the most convenient strategy when manufacturing such systems—because one does not need to selectively choose the networks, manufactured crossbars can be easily programmed without the need to replace them if they perform poorly when working individually (unless their effect is so detrimental that they have to be ignored which can be made possible with this technique). Besides, devices might change over time, thus these simulations, which show what happens when one does not selectively choose the networks, are valuable to investigate conditions where it is not possible to replace the networks.

In the simulations, 25 base networks were used (each having different set of weights) for each of the architectures. Then all of their weights were mapped onto pairs of conductances using HRS/LRS values extracted from experiments. Finally, to reflect the effect of each of the non-idealities, all networks were disturbed multiple times. In each disturbance iteration, multiple combinations of networks were chosen and their performance as a committee of certain size was evaluated. In total, for ~~each simulation (except numerically optimised committees which used 1,000 points)~~most simulations, 10,000 data points were recorded for a committee of every size—these data captured the variations of base networks, their combinations and different disturbance iterations. Only simulations involving line resistance or numerical optimisation of weights had fewer data points for some committee sizes (due to long simulation times).

DATA AVAILABILITY

~~All data generated or analysed during~~ The data that support the findings of this study are ~~included in this published article (and its supplementary information file)~~available from the corresponding author upon reasonable request.

AUTHOR CONTRIBUTIONS

A.M. and D.J. conceived the idea and designed the study. A.M., P.F. and Z.C. performed the experimental measurements. D.J. performed the simulations and analysed the experimental and simulation results. C.L. and Q.X. provided the experimental data of the programming of a Ta/HfO_2 1T1R RRAM array. A.M., W.D.Z. and A.J.K. supervised the research. D.J. wrote the initial manuscript. All authors contributed to the discussions of the results and improved the text.

COMPETING INTERESTS STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

FUNDING

A.M. acknowledges funding from the Royal Academy of Engineering under the Research Fellowship scheme, A.J.K. acknowledges funding from the Engineering and Physical Sciences Research Council (EP/P013503/1) and the Leverhulme Trust (RPG-2016-135), W.D.Z. acknowledges funding from the Engineering and Physical Sciences Research Council (EP/S000259/1).

559 [1] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning
560 in NLP," *arXiv preprint arXiv:1906.02243*, 2019.
561 [2] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with
562 pruning, trained quantization and huffman coding," in *International Conference on Learning
563 Representations*, 2016, San Juan (Puerto Rico), arXiv preprint arXiv:1510.00149.
564 [3] C. Li, Z. Wang, M. Rao, D. Belkin, W. Song, H. Jiang, P. Yan, Y. Li, P. Lin, M. Hu, N. Ge,
565 J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Long short-term
566 memory networks in memristor crossbar arrays," *Nature Machine Intelligence*, vol. 1, no. 1,
567 pp. 49–57, 2019, doi: 10.1038/s42256-018-0001-4.
568 [4] Z. Wang, C. Li, W. Song, M. Rao, D. Belkin, Y. Li, P. Yan, H. Jiang, P. Lin, M. Hu, J. P.
569 Strachan, N. Ge, M. Barnell, Q. Wu, A. G. Barto, Q. Qiu, R. S. Williams, Q. Xia, and J. J.
570 Yang, "Reinforcement learning with analogue memristor arrays," *Nature Electronics*, vol. 2,
571 no. 3, p. 115, 2019, doi: 10.1038/s41928-019-0221-6.
572 [5] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, W. Wang, and D. Ielmini, "Solving matrix
573 equations in one step with cross-point resistive arrays," *Proceedings of the National Academy
574 of Sciences*, vol. 116, no. 10, pp. 4123–4128, 2019, doi: 10.1073/pnas.1815682116.
575 [6] S. R. Nandakumar, M. Le Gallo, I. Boybat, B. Rajendran, A. Sebastian, and E. Eleftheriou,
576 "A phase-change memory model for neuromorphic computing," *Journal of Applied Physics*,
577 vol. 124, no. 15, p. 152135, 2018, doi: 10.1063/1.5042408.
578 [7] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. D. Nolfo, S. Sidler, M. Gior-
579 dano, M. Bodini, N. C. P. Farinha, B. Killeen, C. Cheng, Y. Jaoudi, and G. W. Burr,
580 "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*,
581 vol. 558, no. 7708, pp. 60–67, 2018, doi: 10.1038/s41586-018-0180-5.
582 [8] S. Yu, Z. Li, P. Y. Chen, H. Wu, B. Gao, D. Wang, W. Wu, and H. Qian, "Binary neu-
583 ral network with 16 Mb RRAM macro chip for classification and online training," in *In-
584 ternational Electron Devices Meeting*. IEEE, 2016, San Francisco (United States), doi:
585 10.1109/IEDM.2016.7838429.
586 [9] J. Woo, K. Moon, J. Song, S. Lee, M. Kwak, J. Park, and H. Hwang, "Improved synap-
587 tic behavior under identical pulses using AlO_x/HfO_2 bilayer RRAM array for neuromor-

588 phic systems," *IEEE Electron Device Letters*, vol. 37, no. 8, pp. 994–997, 2016, doi: 10.1109/LED.2016.2582859.

590 [10] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015, doi: 10.1038/nature14441.

593 [11] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature communications*, vol. 9, no. 1, p. 2385, 2018, doi: 10.1038/s41467-018-04484-2.

597 [12] A. Chen and M. R. Lin, "Variability of resistive switching memories and its impact on crossbar array performance," in *2011 International Reliability Physics Symposium.* IEEE, 2011, Monterey (United States), doi: 10.1109/IRPS.2011.5784590.

600 [13] J. Kang, Z. Yu, L. Wu, Y. Fang, Z. Wang, Y. Cai, Z. Ji, J. Zhang, R. Wang, and Y. Yang, "Time-dependent variability in RRAM-based analog neuromorphic system for pattern recognition," in *International Electron Devices Meeting.* IEEE, 2017, San Francisco (United States), doi: 10.1109/IEDM.2017.8268340.

604 [14] L. Xia, W. Huangfu, T. Tang, X. Yin, K. Chakrabarty, Y. Xie, Y. Wang, and H. Yang, "Stuck-at fault tolerance in RRAM computing systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 102–115, 2017, doi: 10.1109/JETCAS.2017.2776980.

608 [15] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, S. Williams, J. Yang, and Q. Xia, "Analogue signal and image processing with large memristor crossbars," *Nature Electronics*, vol. 1, no. 1, pp. 52–59, 2018, doi: 10.1038/s41928-017-0002-z.

612 [16] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, and E. Eleftheriou, "Mixed-precision in-memory computing," *Nature Electronics*, vol. 1, no. 4, p. 246, 2018, doi: 10.1038/s41928-018-0054-8.

615 [17] M. Hu, J. P. Strachan, Z. Li, and S. R. William, "Dot-product engine as computing memory to accelerate machine learning algorithms," in *17th International Symposium on Quality Electronic Design*, 2016, Santa Clara (United States), doi: 10.1109/ISQED.2016.7479230.

618 [18] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nature*
619 *materials*, vol. 18, no. 4, p. 309, 2019, doi: 10.1038/s41563-019-0291-x.
620 [19] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits,"
621 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist
622 [20] A. Mehonic, D. Joksas, W. H. Ng, M. Buckwell, and A. J. Kenyon, "Simulation of inference
623 accuracy using realistic RRAM devices," *Frontiers in Neuroscience*, vol. 13, p. 593, 2019, doi:
624 10.3389/fnins.2019.00593.
625 [21] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid
626 neural networks," in *Artificial Neural Networks for Speech and Vision.* Chapman and Hall,
627 1993, pp. 126–142.
628 [22] S. Hashem and B. Schmeiser, "Improving model accuracy using optimal linear combinations of
629 trained neural networks," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 792–794,
630 1995, doi: 10.1109/72.377990.
631 [23] B. Li, L. Xia, P. Gu, Y. Wang, and H. Yang, "Merging the interface: Power, area and accuracy
632 co-optimization for RRAM crossbar-based mixed-signal computing system," in *Proceedings of*
633 *the 52nd Annual Design Automation Conference*, 2015, San Francisco (United States), doi:
634 10.1145/2744769.2744870.
635 [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional
636 neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105,
637 Lake Tahoe (United States), doi: 10.1145/3065386.
638 [25] Z. Wang, C. Li, P. Lin, M. Rao, Y. Nie, W. Song, Q. Qiu, Y. Li, P. Yan, J. P. Strachan,
639 N. Ge, N. McDonald, Q. Wu, M. Hu, H. Wu, R. S. Williams, Q. Xia, and J. J. Yang, "In situ
640 training of feed-forward and recurrent convolutional memristor networks," *Nature Machine*
641 *Intelligence*, vol. 1, no. 9, pp. 434–442, 2019, doi: 10.1038/s42256-019-0089-1.
642 [26] H. Jiang, L. Han, P. Lin, Z. Wang, M. H. Jang, Q. Wu, M. Barnell, J. J. Yang, H. L. Xin, and
643 Q. Xia, "Sub-10 nm ta channel responsible for superior performance of a HfO_2 memristor,"
644 *Scientific reports*, vol. 6, p. 28525, 2016, doi: 10.1038/srep28525.
645 [27] G. W. Burr, R. M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat, R. S. Shenoy,
646 P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. Hwang, "Experimen-
647 tal demonstration and tolerancing of a large-scale neural network (165 000 synapses) using
648 phase-change memory as the synaptic weight element," *IEEE Transactions on Electron De-*

649 *vices*, vol. 62, no. 11, pp. 3498–3507, 2015, doi: 10.1109/TED.2015.2439635.

650 [28] Y. Fan, L. Zhang, D. Crotti, T. Witters, M. Jurczak, and B. Govoreanu, “Direct evidence
651 of the overshoot suppression in Ta_2O_5-based resistive switching memory with an integrated
652 access resistor,” *IEEE Electron Device Letters*, vol. 36, no. 10, pp. 1027–1029, 2015, doi:
653 10.1109/LED.2015.2470081.

654 [29] B. Govoreanu, D. Crotti, S. Subhechha, L. Zhang, Y. Chen, S. Clima, V. Paraschiv, H. Hody,
655 C. Adelmann, M. Popovici, O. Richard, and M. Jurczak, “A-VMCO: A novel forming-free, self-
656 rectifying, analog memory cell with low-current operation, nonfilamentary switching and excel-
657 lent variability,” in *Symposium on VLSI Technology*, 2015, Kyoto (Japan), doi: 10.1109/VL-
658 SIT.2015.7223717.

659 [30] Z. Chai, W. Zhang, P. Freitas, F. Hatem, J. F. Zhang, J. Marsland, B. Govoreanu, L. Goux,
660 G. S. Kar, S. Hall, P. Chalker, and J. Robertson, “The over-reset phenomenon in Ta_2O_5
661 RRAM device investigated by the RTN-based defect probing technique,” *IEEE Electron Device*
662 *Letters*, vol. 39, no. 7, pp. 955–958, 2018, doi: 10.1109/LED.2018.2833149.

663 [31] C. Sung, S. Lim, H. Kim, T. Kim, K. Moon, J. Song, J.-J. Kim, and H. Hwang, “Effect
664 of conductance linearity and multi-level cell characteristics of TaO_x-based synapse device on
665 pattern recognition accuracy of neuromorphic system,” *Nanotechnology*, vol. 29, no. 11, p.
666 115203, 2018, doi: 10.1088/1361-6528/aaa733.

667 [32] Y. Fang, Z. Yu, Z. Wang, T. Zhang, Y. Yang, Y. Cai, and R. Huang, “Improvement of HfO_x-
668 based RRAM device variation by inserting ALD TiN buffer layer,” *IEEE Electron Device*
669 *Letters*, vol. 39, no. 6, pp. 819–822, 2018, doi: 10.1109/LED.2018.2831698.

670 [33] B. Govoreanu, A. Redolfi, L. Zhang, C. Adelmann, M. Popovici, S. Clima, H. Hody,
671 V. Paraschiv, I. Radu, A. Franquet, J. C. Liu, J. Swerts, O. Richard, H. Bender, L. Altimime,
672 and M. Jurczak, “Vacancy-modulated conductive oxide resistive RAM (VMCO-RRAM): An
673 area-scalable switching current, self-compliant, highly nonlinear and wide on/off-window re-
674 sistive switching cell,” in *International Electron Devices Meeting*. IEEE, 2013, Washington
675 (United States), doi: 10.1109/IEDM.2013.6724599.

676 [34] A. J. Kenyon, M. S. Munde, W. H. Ng, M. Buckwell, D. Joksas, and A. Mehonic, “The
677 interplay between structure and function in redox-based resistance switching,” *Faraday Dis-*
678 *cussions*, vol. 213, pp. 151–163, 2019, doi: 10.1039/C8FD00118A.

679 [35] W. Wu, H. Wu, B. Gao, P. Yao, X. Zhang, X. Peng, S. Yu, and H. Qian, "A methodology
680 to improve linearity of analog RRAM for neuromorphic computing," in *Symposium on VLSI*
681 *Technology*. IEEE, 2018, Honolulu (United States), doi: 10.1109/VLSIT.2018.8510690.

682 [36] Z. Chai, P. Freitas, W. Zhang, F. Hatem, J. F. Zhang, J. Marsland, B. Govoreanu, L. Goux,
683 and G. S. Kar, "Impact of RTN on pattern recognition accuracy of RRAM-based synaptic
684 neural network," *IEEE Electron Device Letters*, vol. 39, no. 11, pp. 1652–1655, 2018, doi:
685 10.1109/LED.2018.2869072.

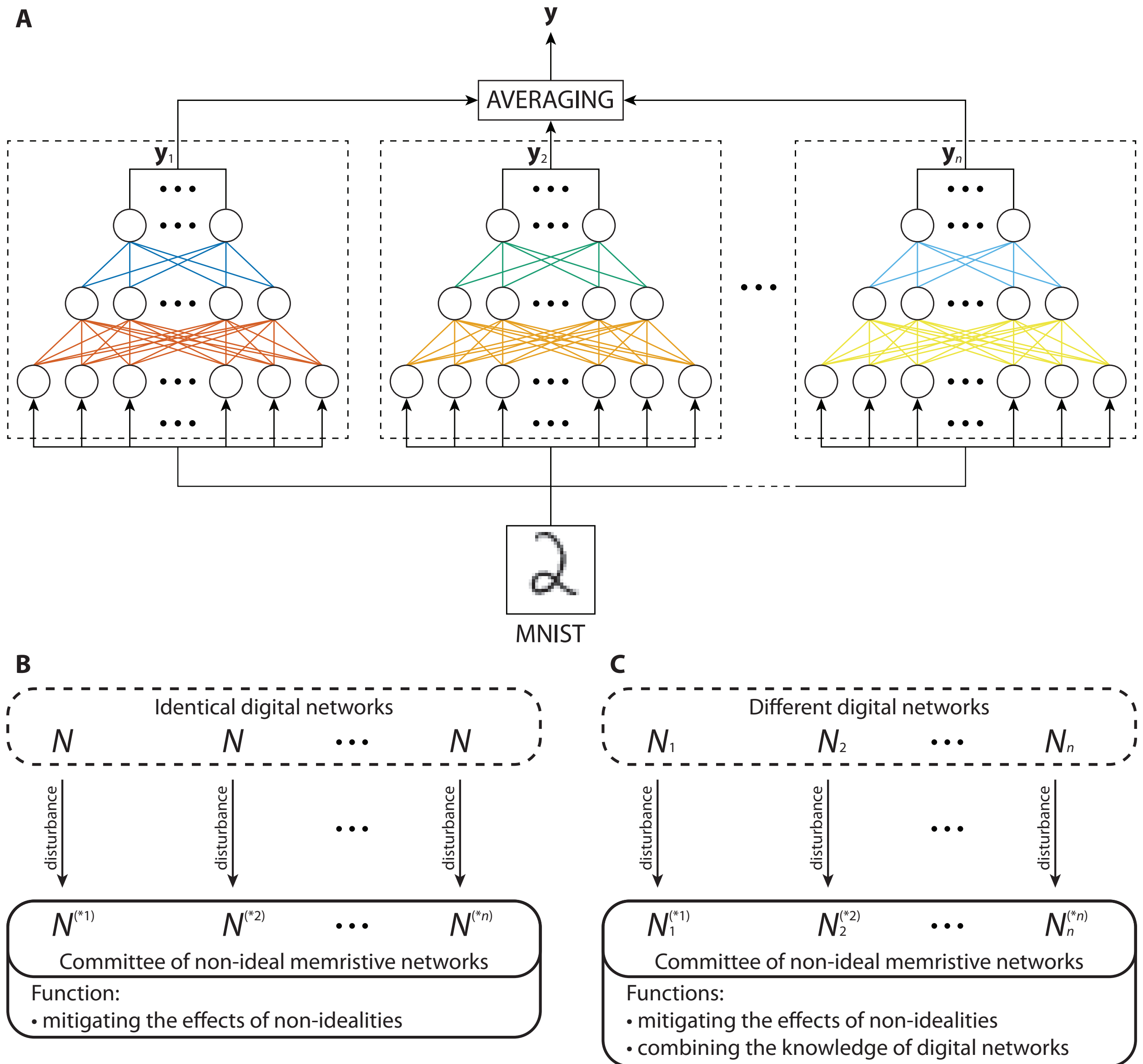

Figure 1. Using multiple neural networks to improve inference accuracy. **A**) The principle of EA. **B**) Using identical digital networks when implementing committees of memristive neural networks only helps to deal with the damage to the networks caused by the non-idealities. **C**) Using different digital networks when implementing committees of memristive neural networks both helps to deal with the damage to the networks caused by the non-idealities and allows to combine the knowledge ~~of individual digital networks~~ about the data set acquired by individual digital networks.

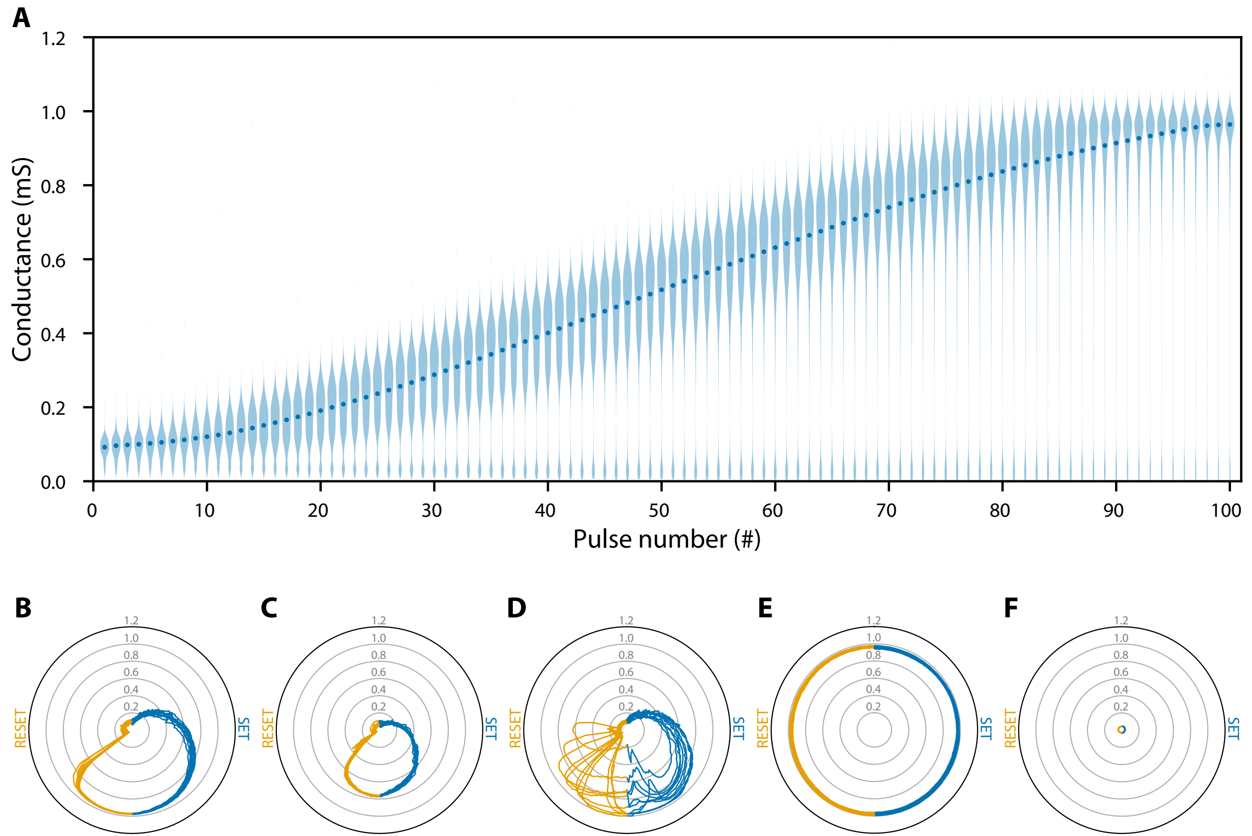

Figure 2. Experimental data of Ta/HfO_2 RRAM crossbar array of shape 128 × 64. **A**) Modulation of devices' conductance over 11 SET cycles, each consisting of a 100 potentiating pulses. Violin plots of gradual conductance changes are shown for all Ta/HfO_2 devices, with dots representing median conductance after a certain number of pulses. 100 points were used for Gaussian kernel density estimation. All violin plots have their maximum widths normalised. **B**-**F**) Examples of devices with their conductance (in mS) **B**) spanning the full range, **C**) spanning part of the full range, **D**) exhibiting cycle-to-cycle variability, **E**) stuck at high values, **F**) stuck at low values. These diagrams show conductance of five devices from Ta/HfO_2 crossbar array over 11 SET and RESET cycles. The radial component represents the conductance, while the angular component represents the number of applied pulses. The first SET cycle starts at the top of each of the diagrams. The conductance (in blue) over 100 SET pulses is displayed in a clockwise fashion across the right half of each of the diagrams. Following that, conductance (in orange) over 100 RESET pulses (starting at the bottom) is displayed across the left half of each of the diagrams, after which the next cycle is displayed. Cartesian version of these plots is shown in Supplementary Figure S9.

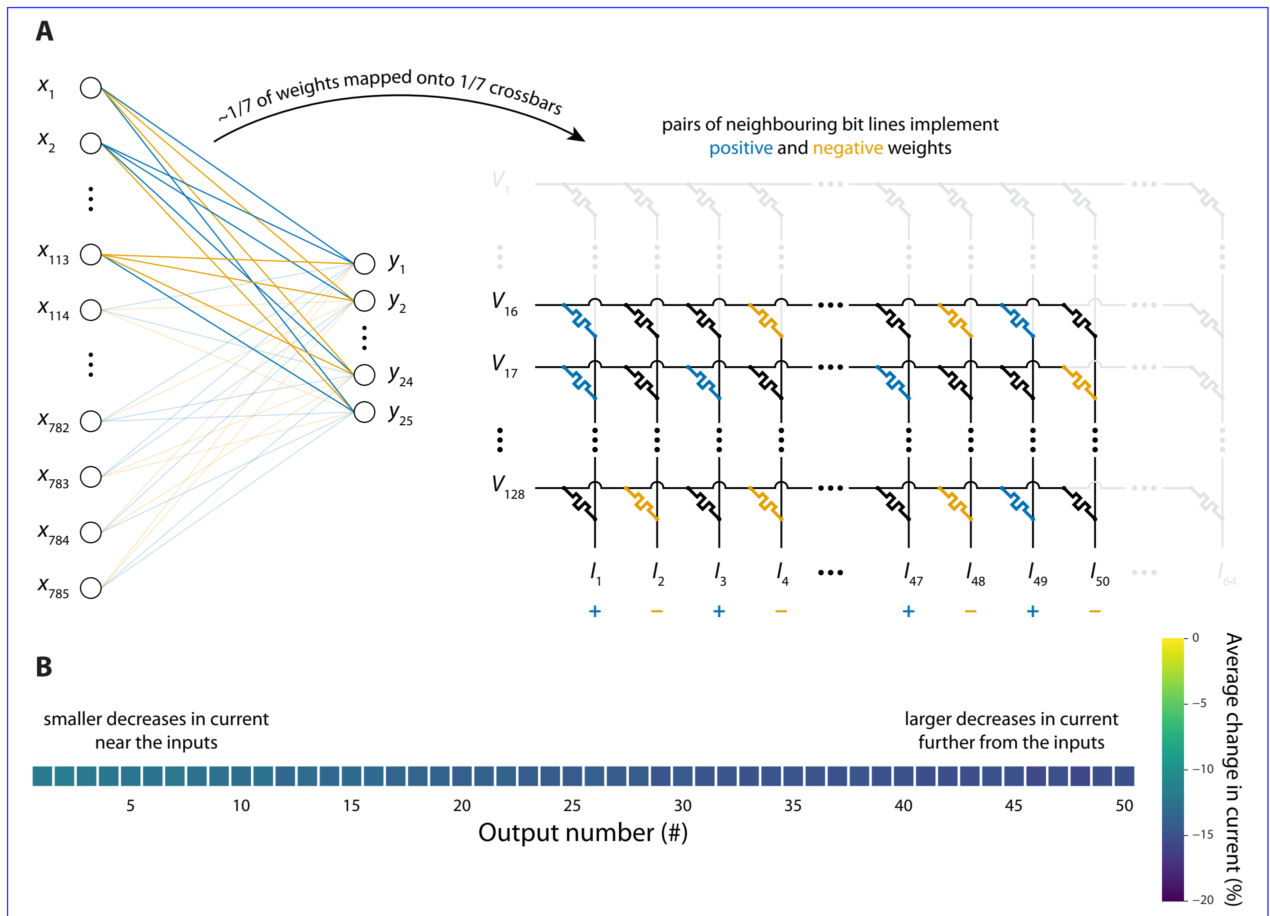

Figure 3. Theoretical implementation of a synaptic layer of shape 785×25 using crossbars of shape 128×64. **A**) Mapping the first subset of weights onto one of the seven crossbars used to implement the whole synaptic layer. Positive weights and negative weights are mapped onto memristors in different bit lines. **B**) Heatmap of average changes in output currents due to line resistance (in all seven Ta/HfO_2 crossbars)~~without and with a scheme that maps certain inputs onto certain word lines depending on expected average intensities of those inputs~~. For this particular simulation, it was assumed that Ta/HfO_2 devices can be programmed perfectly.

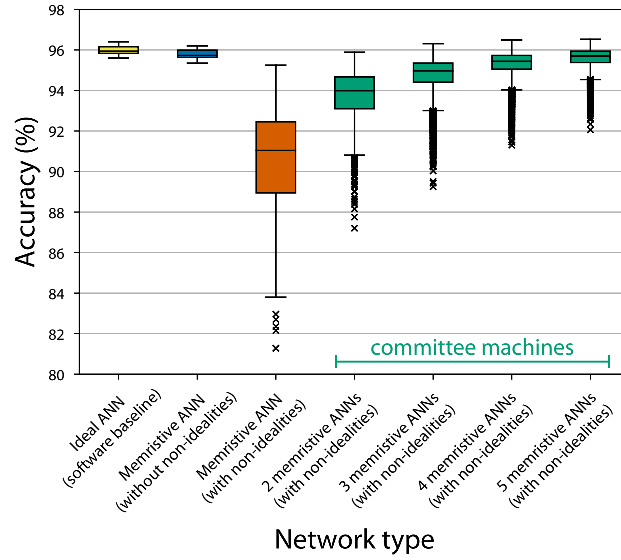

Figure 4. Accuracy achieved by individual networks and their committees when faulty devices, D2D variability data and line resistance of Ta/HfO_2 crossbar are taken into account. The maximum whisker length is set to $1.5 \times$ IQR.

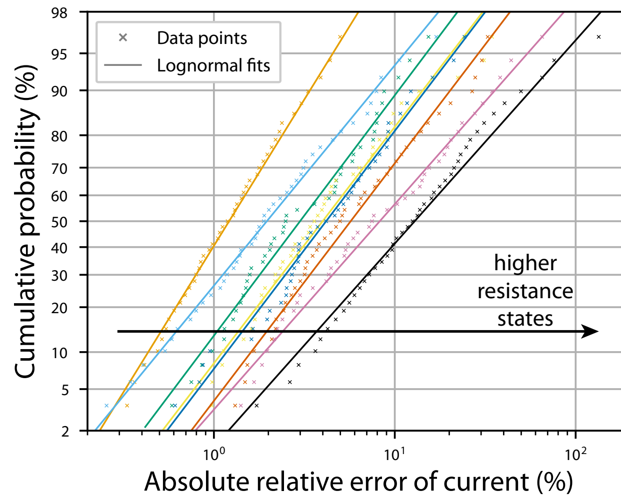

Figure 5. Cumulative probability plots of RTN-induced relative current deviations for all 8 resistance states of a Ta_2O_5 RRAM device. Lognormal fits are shown for each resistance state.

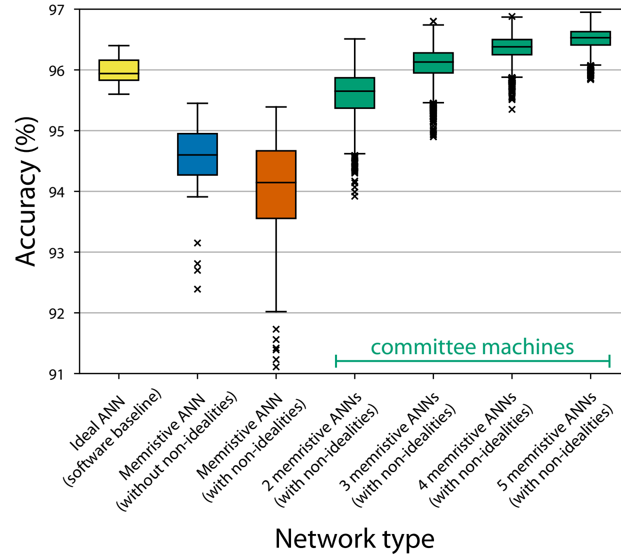

Figure 6. Accuracy achieved by individual networks and their committees when RTN data of a Ta_2O_5 device are taken into account. Additionally, interconnect resistance of ~~$0.3\,\Omega$~~ $0.35\,\Omega$ and $0.32\,\Omega$ in the word and bit lines, respectively, (from Ta/HfO_2 array) was used to include line resistance effects. The maximum whisker length is set to $1.5 \times$ IQR.

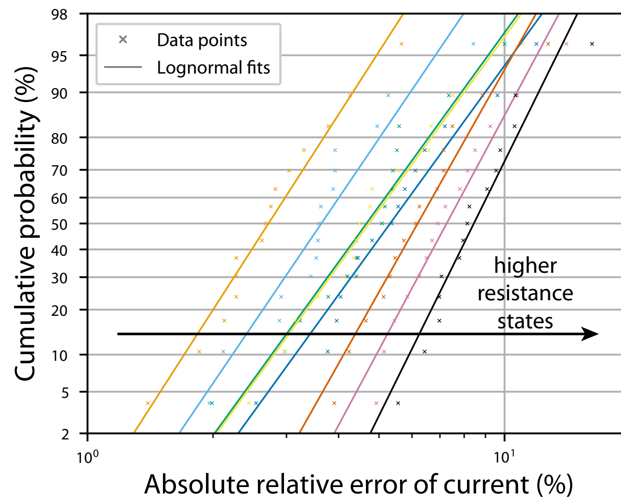

Figure 7. Cumulative probability plots of RTN-induced relative current deviations for all 8 resistance states of aVMCO RRAM device. Lognormal fits are shown for each resistance state.

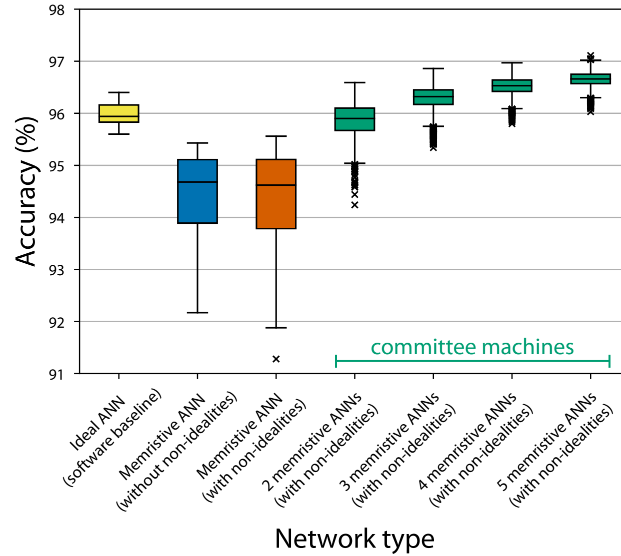

Figure 8. Accuracy achieved by individual networks and their committees when RTN data of an aVMCO device are taken into account. Additionally, interconnect resistance of ~~0.3 Ω~~ 0.35 Ω and 0.32 Ω in the word and bit lines, respectively, (from Ta/HfO_2 array) was used to include line resistance effects. The maximum whisker length is set to $1.5 \times$ IQR.

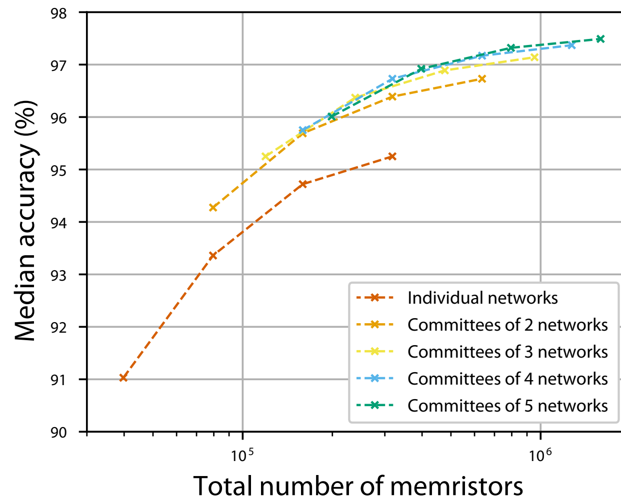

Figure 9. Median accuracy achieved by individual one-hidden-layer memristor-based networks and their committees, when controlled for total number of memristors required. The networks contained 25, 50, 100 or 200 hidden neurons and were disturbed using faulty devices and D2D variability data from Ta/HfO_2 crossbar.

687 **TABLES**

First author (year)	Non-ideality	Device type	Proposed solution
C. Sung (2018) [31]	Current/voltage non-linearity	TaO_x RRAM	Hot-forming step is adopted
C. Li (2018) [15]	Current/voltage non-linearity	Ta/HfO_2 RRAM	1T1R architecture is adopted
Y. Fang (2018) [32]	Device-to-device variability	HfO_x RRAM	Ultra-thin ALD-TiN buffer layer is introduced
B. Govoreanu (2013) [33]	Device-to-device variability	Al_2O_3/TiO_2 (VMCO) RRAM	Non-filamentary RRAM is adopted
A. J. Kenyon (2019) [34]	Device-to-device variability	SiO_x RRAM	The roughness of bottom electrodes is increased
L. Xia (2017) [14]	Faulty devices	-	A modified mapping algorithm and redundancy schemes are used
S. Ambrogio (2018) [7]	Limited dynamic range	PCM	Two pairs of conductance of varying significance for every synaptic weight are used
M. Hu (2016) [17]	Line resistance	-	Advanced mapping algorithms are used to compensate for line resistance effects
W. Wu (2018) [35]	Programming non-linearity	HfO_x RRAM	Electro-thermal modulation layer is deposited on the switching layer
J. Woo (2016) [9]	Programming non-linearity	HfO_2 RRAM	Bilayer structure is adopted
S. Ambrogio (2018) [7]	Programming non-linearity	PCM	PCM devices are used together with CMOS transistors
Z. Chai (2018) [36]	Random telegraph noise	TiO_2/a-Si (aVMCO) RRAM	Non-filamentary RRAM is adopted

Table I. Examples of past efforts at dealing with non-idealities of memristive devices and their systems.