

Plasticity-Enhanced Domain-Wall MTJ Neural Networks for Energy-Efficient Online Learning

Christopher H. Bennett¹, T. Patrick Xiao¹, Can Cui², Naimul Hassan³, Otitoaleke G. Akinola², Jean Anne C. Incorvia², Alvaro Velasquez⁴, Joseph S. Friedman³, and Matthew J. Marinella¹

¹ Sandia National Laboratories, Albuquerque, NM 87123

²Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712

³Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX 75080

⁴ Information Directorate, Air Force Research Laboratory, Rome, NY 13441

{cbennet, mmarine}@sandia.gov, joseph.friedman@utdallas.edu, alvaro.velasquez.1@us.af.mil

Abstract—Machine learning implements backpropagation via abundant training samples. We demonstrate a multi-stage learning system realized by a promising non-volatile memory device, the domain-wall magnetic tunnel junction (DW-MTJ). The system consists of unsupervised (clustering) as well as supervised sub-systems, and generalizes quickly (with few samples). We demonstrate interactions between physical properties of this device and optimal implementation of neuroscience-inspired plasticity learning rules, and highlight performance on a suite of tasks. Our energy analysis confirms the value of the approach, as the learning budget stays below $20\mu J$ even for large tasks used typically in machine learning.

I. INTRODUCTION

The drive towards autonomous learning systems requires computing tasks locally or *in-situ*, defraying rising energy costs due to inefficiencies in the modern computer architecture [1]. A variety of emerging non-volatile memory devices, such as phase-change materials, filamentary resistive RAM, and magnetic memories (spin-transfer-torque-RAM (STT-RAM) and spin-orbit-torque-RAM (SOT-RAM)), may implement this vision. Critically, emerging devices can perform not only data storage but complex physics-powered operations such as vector-matrix multiplies (VMMs) when densely wired [2].

The workhorse algorithm in AI workloads is backpropagation of error (BP). BP relies upon a teacher signal supplied to all layers and the storage of high-quality gradients on each layer during the parameter update phase [3]. In contrast, competitive learning or adaptive resonance methods provide labels sparsely, *e.g.* only to some parts of the system; the rest learn according to internally adaptive units and/or dynamics [4]. Competitive learning relies upon the winner-take-all (WTA) motif, a cascadable non-linear operation that can be used to build deep systems, just as perceptrons can be used to build multi-layer perceptrons (MLP) [5], [6]. Original proposals for building WTA circuits relied upon a chain of inhibition transistors [7]. Analog and digital WTA or spike feedback CMOS systems have been realized [8], [9], and conceptual proposals for WTA systems using emerging devices exist [10], [11]. However, these works either do not discuss scalable (local) learning rules that might lead to large-scale WTA systems, or do not adequately benchmark against state-of-the-art tasks in the machine learning field .

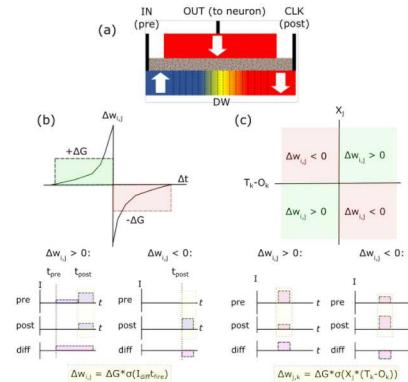


Fig. 1. (a) An illustration of the DW-MTJ analog synapse. (b) abstract learning rule (top) and its temporal implementation using physical currents in the system for the WTA competitive learning system (first layer) and (c) the same for the supervised learning system (second layer).

In order to implement efficient WTA learning, we draw upon the spike-timing-dependent plasticity (STDP) rule, a primitive predictive/correlative engine [12]. As in [13], we implement STDP and WTA learning together with emerging memory, however our chosen synapses are analog and, as in [14], we closely study neuronal behavior/interactions to implement optimal competitive learning with hidden units.

Our chosen analog memory is the three-terminal magnetic-tunnel-junction (3T-MTJ) device. These devices: 1) achieve high switching efficiency due to the SOT interaction at input/output terminals; 2) possess a non-volatile state variable, a domain-wall interface (DWI) moving through a soft ferromagnetic track; 3) can be dually utilized as a synapse, holding an internal conductance state G_{MTJ} when the output terminal is long, or implement the neuron function, when the track is long. In the former case, domain wall synapses notably possess good energy footprint and advantageous operation on neural network tasks in comparison to other nanodevice synaptic options [15]. In the latter case, assuming tight spacing lateral inhibition exists between neighboring DW-MTJ neuron tracks, and the physics-derived leak function can be used to implement rapid inference operations given pre-trained weights [16]. In this work , we describe an efficient combination of unsupervised

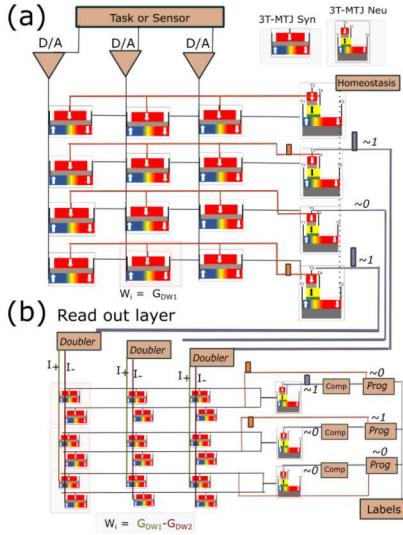


Fig. 2. (a) The clustering layer(s) takes as input and output analog currents. Following spiking of 3T-MTJ neurons, a separate (4th) terminal back-propagates spikes to the row implementing clustering. The clock terminal is typically grounded but may be connected to a homeostatic regulation signal. (b) The read-out layer takes spikes from the hidden layer doubled using current mirrors and encodes real-valued weights with 2 MTJ synapses per cell. Programming follows comparison of spikes to expected values.

(WTA+STDP) and supervised (label-driven) learning in an all-DW-MTJ device array that approaches BP-level performance and remarkable energy efficiency on difficult tasks.

II. OPERATION OF NANOMAGNETIC WTA PRIMITIVE

Our system relies upon three operations 1) Inference: a vector-matrix-multiplies on clustered weights \mathcal{W}_c generate post-synaptic outputs x_* . 2) Domain-Wall Competition : A dynamic step whereby interacting neuron units evolve according to post-synaptic inputs (a vector of currents I_{post}), as well as the behavior or nearby neighbor units, according to a physics-informed model. 3) Learning/Programming: An update step where weights \mathcal{W}_c are updated according to a simplified version of the spike-timing-dependent plasticity (STDP) rule; neurons implement different hidden statistical models of the input [17]. These stages are progressively implemented in the unsupervised phase (label-free). Once N_{us} unlabeled examples from the training set have been seen, weights are frozen and a least-mean-squares (LMS) filter is progressively built in a second weights matrix \mathcal{W}_s using N_s labeled data points.

A. Details of Lateral Inhibition Model

As in [16], the dependence of a magnetic stray field's transverse (vertical) component impinges upon that of neighboring wires. This can be described by:

$$H_z = \frac{4M_s}{\pi} \left(\arctan \frac{2s + 3w}{t} - \arctan \frac{2s + w}{t} \right) \quad (1)$$

based on [18]. Here, M_s is the magnetic saturation field set at 1.6T, w , t , and s are width, thickness of the track and inter-wire spacing respectively. When H_z is in the proper range, it can effectively reduce DW velocity v . Instead of

rigorously calculating H_z in the neural simulator, we focus on an ensemble parameter γ that modifies naive, current-dominated DW motion v_0 :

$$\gamma = \frac{v_0(w, t, I_{in}) - v_{\text{inhib}}(w, t, I_{in}, s)}{v_0(w, t, I_{in})} \quad (2)$$

This ratio captures the predominance of current-driven vs. coupled (field-driven) DW behavior. At very low γ , field influences are negligible; at $\gamma \approx 0.5$, coupling is intermediate, and current and field DW influences are mixed; as γ approaches 1, neighbor field effects outweigh the influence of input current. Physically, the spacing s can vary between 10nm and 150nm spacing in order to reflect a full spectrum of coupling strength. However, γ may not evolve linearly in this regime, as demonstrated in [19].

B. Details of Analog Plasticity Model

As in [20], the number of weights given a domain wall length L_{dw} , track width w , and length of output MTJ terminal L_{mtj} (where the analog conductances are realized) is

$$n_w \leq \frac{L_{\text{mtj}}}{L_{\text{dw}}} \approx 4 \frac{L_{\text{mtj}}}{w} \quad (3)$$

Given $w = 32\text{nm}$, 6 bits could be implemented given an output port length of 512nm. Analog weights can be implemented with the use of notches for precise control and non-linearity [21], or can be obtained intrinsically via fine current controlled pulses. Due to DWI momentum effects, notch-free systems will typically require greater output/synapse length.

During plasticity events, differences in currents between synaptic input and output 3T-MTJ ports determines the motion of the DWI modulating G_{MTJ} . As in Fig. 1, the circuit potentiates the synapse/increases the conductance when the two currents are coincident and depotentiates the synapse/decreases the conductance when they are not. This implements an approximate version of Hebbian/anti-Hebbian learning, or approximate STDP (hereafter $A-STDP$). The teacher signal implementation relies upon a fourth terminal of the DW-MTJ neurons which is connected backward to the synaptic devices of that layer, as in the orange wires shown in Fig. 2(a). Further details on the scheme are given in [22].

C. Integration with Companion Supervised Learning System

A WTA primitive can be difficult to interface, leading to the desire to efficiently combine unsupervised and supervised sub-systems [23]. In our case, the results from the competitively learning DW-MTJ system are forward-propagated to a supervised learning layer that is constructed additionally from DW-MTJ synapses and neurons, as shown in Fig. 2 and first suggested in [24]. This system contains $2MN$ total DW synapses to encode both positive and negative weights, where M is the number of hidden nodes and N is the label-applied terminal set of neurons. We have considered two possible strategies for the supervised learning policy. The first sign-based learning policy can be implemented with great energy efficiency in neuromorphic hardware [25], and reduces to:

$$\Delta W_{j,k} = \Delta G \sigma(X_j(T_k - O_k)), \quad (4)$$

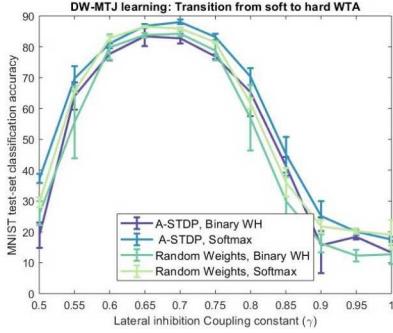


Fig. 3. Calibration of γ ; MNIST, $M = 600$, $N_{us} = 1000$, $N_s = 30000$.

where X_j is the input from hidden neuron j , O_k is the output at the k^{th} terminal neuron, T_k is the target (correct) label, σ is the sign function and ΔG is the unit of conductance change per update. The second policy, softmax learning, requires an analog computation but can achieve superior results in machine learning contexts. Given the original post-synaptic update Y_k , the softmax function is computed subsequently. Weights are ultimately updated according to $\Delta w_{j,k} = -\eta \delta_{j,k}$, given a learning rate η , and $\delta_{j,k} = H_j(O_k - T_k)$ following the cross-entropy formulation, where H_j is the pre-synaptic activation values of that layer j , as in [26].

III. DESCRIPTION OF DATA SCIENCE TASKS

We consider three tasks: 1) the Human Activity Recognition (HAR) set of phone sensor data (*e.g.* body acceleration, angular speed). There are 5 classes of activity (standing, walking, etc), 21,000 training and 2,500 examples of dimension $L = 60$ [27]. 2) the MNIST database of hand-written digits, which includes 60,000 training and a separate 10,000 test examples, and $L = 784$ [28]. 3) The fashion-MNIST (f-MNIST) database, which is of same dimensionality as 2), represents items of clothing (sneakers, t-shirt, etc) and is notably less linearly separable than either of the previous [29].

IV. PERFORMANCE ON TASKS

A. Parameters for successful clustering

For correct clustering system operation, the most critical parameter tends to be the coupling parameter γ . As visible in Fig. 3, while the intermediate/low amount of stray field interaction (over-firing) and dominant stray field interaction (under-firing) both do poorly, the high-intermediate level of interaction in which current matters but is outweighed by locally dominant neighbors results generalizes properly. Computationally, this suggests an intermediate point between 'hard' WTA (in which one or close to one neurons fire) and 'soft' WTA (in which most neurons fire) best implements clustering and forces a useful hidden representations of the input dataset.

Next, we evaluate how critical two common enhancements to standard WTA operation – homeostasis [30] and rank-order coding [31] – are to strong performance in the hidden layer. Fig. 4 shows that these two operations are also important. In the case of homeostasis, we find that a small number

TABLE I
CLASSIFICATION AND REGRESSION TASK PERFORMANCE

Task	Learning Style		
	Random, Ana-BP	STDP, Bin-BP	STDP, Ana-BP
HAR	96.13%	95.83%	96.93%
MNIST	93.52%	93.12%	94.42%
f-MNIST	76.52%	77.52%	79.52%

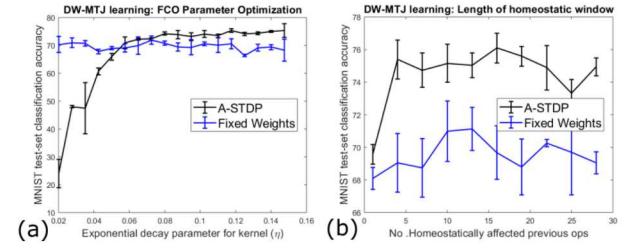


Fig. 4. Rank order filtering (a) and homeostatic delay mechanism (b) contribution to competitive learning with DW-MTJ neuron devices on the MNIST task. Simulated systems had $M = 200$ hidden layer neurons, given $N_{us} = 1000$ clustering samples, and $N_s = 30000$ supervised samples.

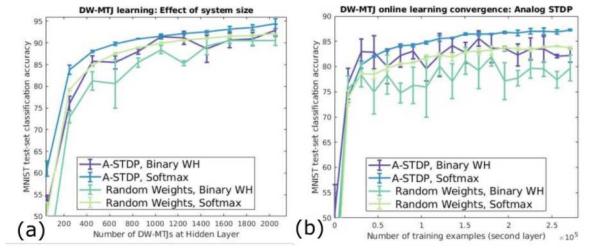


Fig. 5. The effect on MNIST classification performance of (a) the total number of competing hidden layer units and (b) the number of samples provided to the supervised layer to read out the results of the clustering operation. In (b), there are $M = 400$ hidden-layer units.

of homeostatically inhibited time steps provides this benefit already, and a great deal of fine-tuning is not needed. A similar result is obtained for order coded learning, where a sufficiently large exponent is needed to clip the updates to a reasonable number of total neurons firing. Note that when this parameter is very low, the hidden layer tends to again over-fire and redundantly sample. Since correct values of γ also naturally clip the total number that can fire, this suggests that the poor a-STDP results in Fig. 4(a) are unlikely.

B. Dimensional and learning set requirements

Fig. 5 illustrates performance on MNIST task as a function of competing units M and number of supervised training samples given a properly calibrated hidden layer. Ultimately, 94.5% classification on the test-set is achieved when using ana-BP in the second layer with only $N_s = 30000$ examples drawn from the training set (but with a fairly large $M = 1200$). Table 1 summarizes the top results for the other two tasks. For HAR, 97% is reached given $N_s = 15000$ and $M = 600$; f-MNIST requires $M = 1800$ and $N_s = 60000$. This suggests the current design is adequate on more separable tasks, while deeper

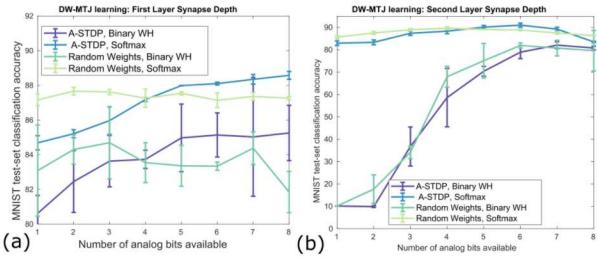


Fig. 6. Effect of writable space of G_{MTJ} for the (a) first/clustered layer and (b) second/supervised layer; synapse depth is 6 bits in the other layer. For both, the task is MNIST and $M = 500$, with 30,000 training samples.

networks may be required to prevent unacceptable system size blow-up on very non-separable (difficult) ones. These are notably low numbers for the total number of labeled data points presented; a modern memristive MLP requires many multiples of the task set, e.g. 200-500k samples for MNIST or f-MNIST [32], [26], and achieves 96% on MNIST and 81 % on f-MNIST. Thus, our present results are very slightly inferior to BP. However, as in Table 1, clustering outperforms the random weights system definitively, given the same learning procedure in the read-out layer.

C. Resilience to Intrinsic Physics Effects in System

Several issues may occur in the physical learning system which are non-ideal: a) synapse-level coarseness, e.g. limited resolution of synapses; b) synapse-level process-induced variation at the output MTJ cell (which creates different $G_{\text{On}}/G_{\text{Off}}$ states and TMR ratio); c) neuron-level stochastic effects due to natural fractal edge roughness in DW-MTJ nanotracks [33] which can cause a neuron, at a given clustering timestep, to fail to compete/fire. For coarseness, Fig. 6(a) shows that \mathcal{W}_c requires 4 bits per synapse to outperform random weights, regardless of second layer policy; performance continues to increase with more resolution, leveling off at 7-8 bits. Meanwhile, the supervised layer is sensitive to synaptic depth when using the binary BP rule but insensitive to it when using the analog rule- regardless of first-layer weight style. Next, Fig. 7(a) shows that the clustering operation is almost unaffected by synapse-level variability. Finally, Fig. 7(b) shows the effects of arbitrary domain wall pinning are significant and linear. If around 5% of neurons do not fire at any given clustering step, 1.0 – 2.5% accuracy is lost. However, the effect of random pinning is negligible when in low current operation.

V. ENERGY FOOTPRINT OF PROPOSED SYSTEMS

Drawing on methodology in [26], [34], and [35], we estimate the energy overhead for the entire online learning procedure. On the device level, we have assumed that on average average $R_{\text{MTJ}} = 1\text{k}\Omega$, DW velocity is $100\frac{\text{m}}{\text{s}}$, $J = 1.0 \times 10^{11}\frac{\text{A}}{\text{m}^2}$ for SOT switching, $w = 32\text{nm}$, $d = 4\text{nm}$, and L_{mtj} is chosen according to Equation (3). We assume the circuit operates in current mode during VMM operations and during the training/plasticity events, and no additional analog-to-digital conversion (ADC) is needed at the hidden layer

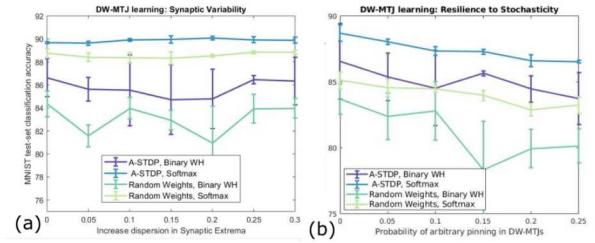


Fig. 7. (a) The effect of increasing variability of maximum and minimum states of DW synapses ($G_{\text{on}}, G_{\text{off}}$) in first layer, e.g. TMR variation. (b) The effect of random DW pinning. For both cases, $M = 600, 1000$ clustering, 30000 training examples given on the MNIST task.

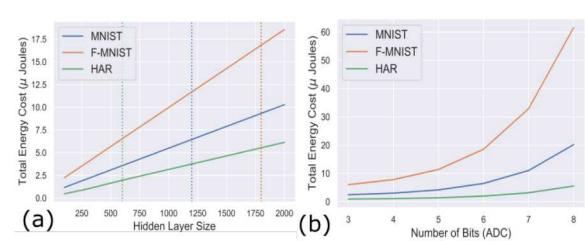


Fig. 8. Dependence of energy footprint given (a) hidden layer dimension M assuming 6 bits in the ADC, and (b) the ADC bit resolution, given the M values noted in (a) as the dotted (vertical) lines.

due to the all-DW design. However, at the output layer, a Ramp ADC, comparators, and softmax subthreshold circuit are implemented to fully interface with digital labels. Based on our estimates, this peripheral circuitry dominates the overall energy footprint and leads to the following results at 6 bits of ADC accuracy for the three tasks using clustered weights and ana-BP in \mathcal{W}_s : $1.96\ \mu\text{J}$ for HAR, $7.41\ \mu\text{J}$ for MNIST, and $18.55\ \mu\text{J}$ for f-MNIST. Lastly, we parameterize hidden layer dimension and bits (Fig. 8). While energy scales linearly with the system size, it scales quadratically as a function of bits. Since 6 bits of weight precision is workable for Bin-BP and far less suffices for Ana-BP, no blow-up in energy is expected. Future energy efficiencies may be unlocked by further increasing domain wall velocities via material optimization [36], or increasing the efficiency of spin-orbit torque switching for more efficient current-mode inference operations.

VI. CONCLUSION

In this work, we have designed and evaluated a learning system which closely draws upon the dynamics of DW-MTJ memory devices to learn efficiently. The major positive result of the work is that current-mode (all DW-MTJ) internal operation, low bit requirements, and a low number of required updates allow us to achieve learning with $< 20\mu\text{J}$ energy budget at very high speed. The major incomplete aspect of the work is that our accuracy results are still inferior to state-of-the-art deep networks using BP. Our immediate next steps are thus to examine deeper (cascaded) implementations of semi-supervised DW-MTJ systems that may be ML-competitive.

ACKNOWLEDGMENT

Sandia National Laboratories is a multimission laboratory managed and operated by NTESS, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

REFERENCES

- [1] M. Horowitz, “1.1 computing’s energy problem (and what we can do about it),” in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 10–14.
- [2] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola *et al.*, “Neuromorphic computing using non-volatile memory,” *Advances in Physics: X*, vol. 2, no. 1, pp. 89–124, 2017.
- [3] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, “Backpropagation: The basic theory,” *Backpropagation: Theory, architectures and applications*, pp. 1–34, 1995.
- [4] S. Grossberg, “Competitive learning: From interactive activation to adaptive resonance,” *Cognitive science*, vol. 11, no. 1, pp. 23–63, 1987.
- [5] W. Maass, “Neural computation with winner-take-all as the only nonlinear operation,” in *Advances in neural information processing systems*, 2000, pp. 293–299.
- [6] ———, “On the computational power of winner-take-all,” *Neural computation*, vol. 12, no. 11, pp. 2519–2535, 2000.
- [7] C. A. Mead, J. Lazzaro, M. Mahowald, and S. Ryckebusch, “Winner-take-all circuits for neural computing systems,” Oct. 22 1991, uS Patent 5,059,814.
- [8] S. Ramakrishnan and J. Hasler, “Vector-matrix multiply and winner-take-all as an analog classifier,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 353–361, 2013.
- [9] J. Park, J. Lee, and D. Jeon, “7.6 a 65nm 236.5 nJ/classification neuromorphic processor with 7.5% energy overhead on-chip learning using direct spike-only feedback,” in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2019, pp. 140–142.
- [10] S. N. Truong, K. Van Pham, W. Yang, K.-S. Min, Y. Abbas, C. J. Kang, S. Shin, and K. Pedrotti, “Ta 2 o 5-memristor synaptic array with winner-take-all method for neuromorphic pattern matching,” *Journal of the Korean Physical Society*, vol. 69, no. 4, pp. 640–646, 2016.
- [11] A. Wu, Z. Zeng, and J. Chen, “Analysis and design of winner-take-all behavior based on a novel memristive neural network,” *Neural Computing and Applications*, vol. 24, no. 7-8, pp. 1595–1600, 2014.
- [12] R. P. Rao and T. J. Sejnowski, “Spike-timing-dependent hebbian plasticity as temporal difference learning,” *Neural computation*, vol. 13, no. 10, pp. 2221–2237, 2001.
- [13] A. F. Vincent, J. Larroque, N. Locatelli, N. B. Romdhane, O. Bichler, C. Gamrat, W. S. Zhao, J.-O. Klein, S. Galdin-Retailleau, and D. Querlioz, “Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems,” *IEEE transactions on biomedical circuits and systems*, vol. 9, no. 2, pp. 166–174, 2015.
- [14] D. Krotov and J. J. Hopfield, “Unsupervised learning by competing hidden units,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 16, pp. 7723–7731, 2019.
- [15] D. Kaushik, U. Singh, U. Sahu, I. Sreedevi, and D. Bhowmik, “Comparing domain wall synapse with other non volatile memory devices for on-chip learning in analog hardware neural network,” *arXiv preprint arXiv:1910.12919*, 2019.
- [16] N. Hassan, X. Hu, L. Jiang-Wei, W. H. Brigner, O. G. Akinola, F. Garcia-Sanchez, M. Pasquale, C. H. Bennett, J. A. C. Incorvia, and J. S. Friedman, “Magnetic domain wall neuron with lateral inhibition,” *Journal of Applied Physics*, vol. 124, no. 15, p. 152127, 2018.
- [17] D. Kappel, B. Nessler, and W. Maass, “Stdp installs in winner-take-all circuits an online approximation to hidden markov model learning,” *PLoS computational biology*, vol. 10, no. 3, p. e1003511, 2014.
- [18] R. Engel-Herbert and T. Hesjedal, “Calculation of the magnetic stray field of a uniaxial magnetic domain,” *Journal of Applied Physics*, vol. 97, no. 7, p. 074504, 2005.
- [19] C. Cui, O. G. Akinola, N. Hassan, C. H. Bennett, M. J. Marinella, J. S. Friedman, and J. Incorvia, “Maximized lateral inhibition in paired magnetic domain wall racetracks for neuromorphic computing,” *arXiv preprint arXiv:1912.04505*, 2019.
- [20] J. A. Curran, Y. Jang, M. D. Mascaro, M. A. Baldo, and C. A. Ross, “Low energy magnetic domain wall logic in short, narrow, ferromagnetic wires,” *IEEE Magnetics Letters*, vol. 3, pp. 3 000 104–3 000 104, 2012.
- [21] O. Akinola, X. Hu, C. H. Bennett, M. Marinella, J. S. Friedman, and J. A. C. Incorvia, “Three-terminal magnetic tunnel junction synapse circuits showing spike-timing-dependent plasticity,” *Journal of Physics D: Applied Physics*, vol. 52, no. 49, p. 49LT01, 2019.
- [22] A. Velasquez, C. Bennett, N. Hassan, W. Brigner, O. Akinola, J. A. Incorvia, M. Marinella, and J. Friedman, “Unsupervised competitive hardware learning rule for spintronic clustering architecture,” *GOMAC 2020, Proceedings*, 2020.
- [23] D. Querlioz, W. Zhao, P. Dollfus, J.-O. Klein, O. Bichler, and C. Gamrat, “Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches,” in *2012 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. IEEE, 2012, pp. 203–210.
- [24] C. H. Bennett, N. Hassan, X. Hu, J. A. C. Incorvia, J. S. Friedman, and M. J. Marinella, “Semi-supervised learning and inference in domain-wall magnetic tunnel junction (dw-mtj) neural networks,” in *Spintronics XII*, vol. 11090. International Society for Optics and Photonics, 2019, p. 1109031.
- [25] C. S. Thakur, R. Wang, S. Afshar, G. Cohen, T. J. Hamilton, J. Tapson, and A. van Schaik, “An online learning algorithm for neuromorphic hardware implementation,” *arXiv preprint arXiv:1505.02495*, 2015.
- [26] C. H. Bennett, V. Parmar, L. E. Calvet, J.-O. Klein, M. Suri, M. J. Marinella, and D. Querlioz, “Contrasting advantages of learning with random weights and backpropagation in non-volatile memory neural networks,” *IEEE Access*, 2019.
- [27] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in *Esann*, 2013.
- [28] Y. LeCun, C. Cortes, and C. Burges, “Mnist handwritten digit database,” *AT&T Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>*, vol. 2, p. 18, 2010.
- [29] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.
- [30] D. Querlioz, O. Bichler, A. F. Vincent, and C. Gamrat, “Bioinspired programming of memory devices for implementing an inference engine,” *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1398–1416, 2015.
- [31] B. S. Bhattacharya and S. B. Furber, “Biologically inspired means for rank-order encoding images: A quantitative analysis,” *IEEE transactions on neural networks*, vol. 21, no. 7, pp. 1087–1099, 2010.
- [32] I. Kataeva, F. Merrikh-Bayat, E. Zamanidoost, and D. Strukov, “Efficient training algorithms for neural networks based on memristive crossbar circuits,” in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.
- [33] S. Dutta, S. A. Siddiqui, J. A. Curran-Incorvia, C. A. Ross, and M. A. Baldo, “The spatial resolution limit for an individual domain wall in magnetic nanowires,” *Nano letters*, vol. 17, no. 9, pp. 5869–5874, 2017.
- [34] V. Parmar and M. Suri, “Design exploration of iot centric neural inference accelerators,” in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*. ACM, 2018, pp. 391–396.
- [35] M. J. Marinella, S. Agarwal, A. Hsia, I. Richter, R. Jacobs-Gedrim, J. Niroula, S. J. Plimpton, E. Ipek, and C. D. James, “Multiscale co-design analysis of energy, latency, area, and accuracy of a reram analog neural training accelerator,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 86–101, 2018.
- [36] F. Ajejas, V. Křížáková, D. de Souza Chaves, J. Vogel, P. Perna, R. Guerrero, A. Gudin, J. Camarero, and S. Pizzini, “Tuning domain wall velocity with dzyaloshinskii-moriya interaction,” *Applied Physics Letters*, vol. 111, no. 20, p. 202402, 2017.