Supplementary Information

Distance-aware attention-inspired memristive networks for energy-efficient analog retrieval

Minseong Park[1,2], Jinzhan Li[1], Arjuman Ara Mimi[1], Suhas Kumar[3,4*] & Su-in Yi[1*]

[1]Texas A&M University, College Station, TX, USA

[2]Sandia National Laboratories, Livermore, CA, USA

[3]Nanyang Technological University, Singapore

[4]Rain AI, San Francisco, CA, USA

*Corresponding authors: yisuin@tamu.edu, su1@alumni.stanford.edu

Content: Supplementary Note 1-3, Supplementary Figure S1-S8, and Supplementary References.

Supplementary Note 1 – Comparison between RBM and DIM-SDAM

The hardware implementation of restricted Boltzmann machine (RBM) faces significant bottlenecks: (1) iterative weight tuning over multiple epochs, which strains the endurance of memristors, and (2) the requirement for on-chip random number generation to execute stochastic logistic activations. In contrast, our DIM-SDAM approach sidesteps these complexities by utilizing a deterministic mapping process. This not only preserves device longevity but also ensures that the entire training-to-inference pipeline is self-contained within the hardware, a feat that stochastic models like RBM may fail to achieve fully on-chip. In addition, while RBM implementations typically necessitate differential memristor pairs to represent bipolar weights, our DIM-SDAM is realized through a single-crossbar architecture. By utilizing a single device per synapse, we effectively eliminate the need for power-hungry differential sensing circuitry, enabling an area-efficient and truly autonomous end-to-end system.

RBMs have been successfully utilized for data reconstruction, and conceptually, similar frameworks could be adapted for associative memory tasks[1,2]. While such possibilities remain largely unexplored and have yet to be established in hardware, RBMs could serve as potential candidates for associative memory. However, several aforementioned critical challenges must be overcome to achieve this, which differ significantly from our implementation. Our work here opens the possibility for energy-based networks– including Hopfield networks, Ising machines, and RBMs–to be further explored as candidates for high-density associative memory, while providing a streamlined, deterministic benchmark for full-hardware integration.

Supplementary Note 2 – Energy and Throughput Benchmarking CPU, FPGA, and GPU

We benchmarked the performance of DIMS-DAM on heterogeneous computing platforms, including CPU, GPU, FPGA, and a memristor-based accelerator[1]. Specifically, we used an optimistic compute-bound lower bound of energy efficiency (E_{op}) and throughput (Θ_{op}) for other digital platforms, such as an Intel Xeon Platinum 8180M CPU (14 nm, 205 W, INT8 Θ_{op} =140 GOP/s, E_{op} =0.683 GOP/s/W), an NVIDIA GeForce RTX 5090 GPU (4 nm, 575 W, INT8 Θ_{op} =2.54×147.08×10³ GOP/s, E_{op} =2.54×425/575×346 GOP/s/W) projected by NVIDIA GeForce RTX 4090 D GPU (5 nm, 425 W, INT8 Θ_{op} =147.08×10³ GOP/s, E_{op} =346 GOP/s/W), and a AMD/Xilinx Versal VC1902 FPGA (7 nm, 165 W, INT8 Θ_{op} = 135.00×10³ GOP/s, E_{op} =818.19 GOP/s/W)[3–5]. For all platforms, matrix–vector multiplication operations (denoted as MVM-1 and MVM-2 for similarity and projection, respectively) in each inference were assumed to be executed entirely on the target hardware. Each DIMS-DAM iteration consists of a fixed number of arithmetic operations (one multiplication and one addition): 3.921×10^4 OPs per MVM resulting in N_{op}=7.842×10^4 OPs and thereby a total of N_{total}=2.509×10^6 OPs per batch.

All reported energy and throughput values were normalized per operation (J/OP and s/OP) and subsequently scaled to the full DIMS-DAM workload. INT8 arithmetic was assumed for all digital platforms (CPU, GPU, and FPGA), consistent with commonly reported effective-efficiency configurations[3]. For the memristor-based implementation, mixed-precision analog computation is used, with different input and output bit widths for MVM-1 and MVM-2 consistent to earlier work[1].

For digital platforms (CPU, GPU, FPGA), the energy per batch was calculated to be 1.46×10^{-9} J/OP for the CPU, 1.54×10^{-12} J/OP for the GPU, and 1.22×10^{-12} J/OP for the FPGA. The total energy (E_{batch}) and throughput (Θ_{batch}) per batch are computed as:

$$E_{\text{batch}} = E_{\text{op}} \times N_{\text{total}}, \Theta_{\text{batch}} = \Theta_{\text{op}} / N_{\text{total}}$$

For RRAM, the energy per operation was derived from experimentally reported energy-efficiency values (TOPS/W) for each matrix–vector multiplication[1]. MVM-1 operated with a 5-bit input and 7-bit output precision and achieved an energy efficiency of 10 TOP/s/W, corresponding to an energy cost of 1.0×10^{-13} J/OP. MVM-2 used a lower precision configuration with 2-bit inputs and 4-bit outputs, enabling a higher energy efficiency of 40 TOP/s/W and an energy cost of 2.5×10^{-14} J/OP. The effective energy per operation for the RRAM-based DIMS-DAM implementation was calculated as an average of the MVM-1 and MVM-2 energy costs, proportional to the number of operations performed in each MVM within a single iteration. The corresponding measured throughput was 500 GOPS and 1750 GOPS for MVM-1 and MVM-2, respectively[1]. Using these metrics, Θ_{op} was computed as 2×388.89 GOPS (a factor of two for compensating N_{total} that includes MVM-1 and MVM-2). Moreover, RRAM results were originally reported for a 130-nm CMOS process. To enable comparison with modern digital platforms, energy and throughput were scaled to a 7-nm node.

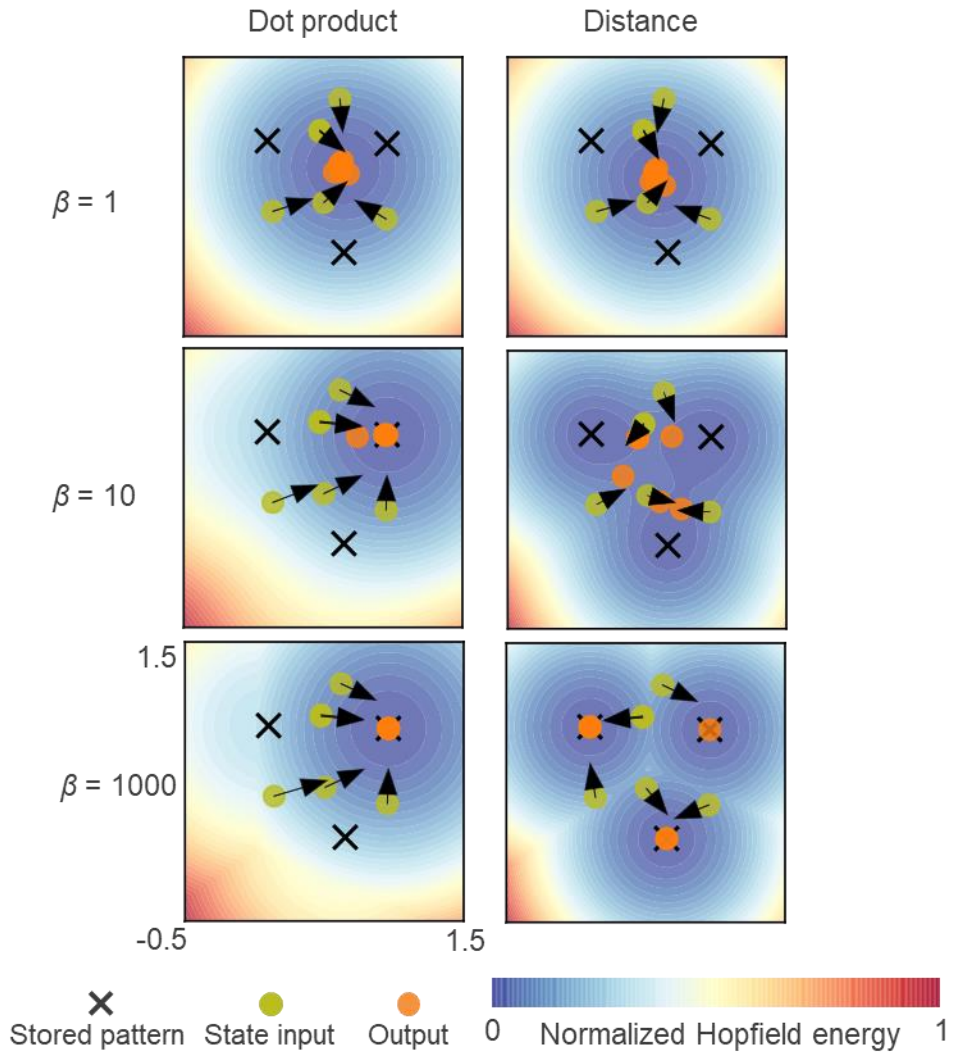
Supplementary Note 3 – Comparison of energy consumptions with TCAM and autoencoder

We benchmarked energy consumptions of representative associative memory and reconstruction model: ternary content-addressable memory (TCAM) and autoencoder, respectively. For TCAM, a representative associative memory, we compare state-of-the-art RRAM with 16T CMOS technologies with respect to the single cell[6,7]. In RRAM technology, the average MVM voltage is 0.05 V, time for MVM is 10 ns, average resistance of RRAM is 10 kΩ, and bit precision is four[6]. The corresponding energy of RRAM-based DIM-SDAM E_{RRAM} is:

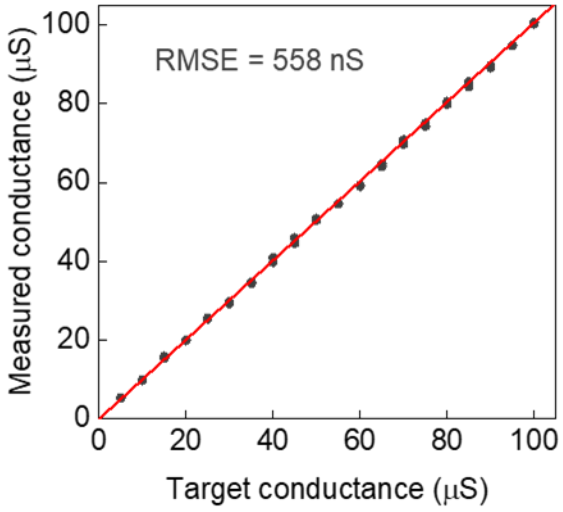
$$E_{\mathrm{RRAM}} = (0.05\ \mathrm{V})^2 \times (10\ \mathrm{k\Omega})^{-1} \times (10\ \mathrm{ns}) \times (4\ \mathrm{bit})^{-1} = 0.625\ \mathrm{fJ\ per\ bit\ per\ search}$$

Note that 'per search' in DIM-SDAM means only MVM-1 for fair comparison with TCAM, which is not capable of reconstruction but computing matching score only. The energy of standard 16T-based TCAM ($E_{16\mathrm{T\_CMOS}}$) was reported as 1 fJ per bit per search[7], resulting in 1.6× energy efficiency in DIM-SDAM compared to TCAM.

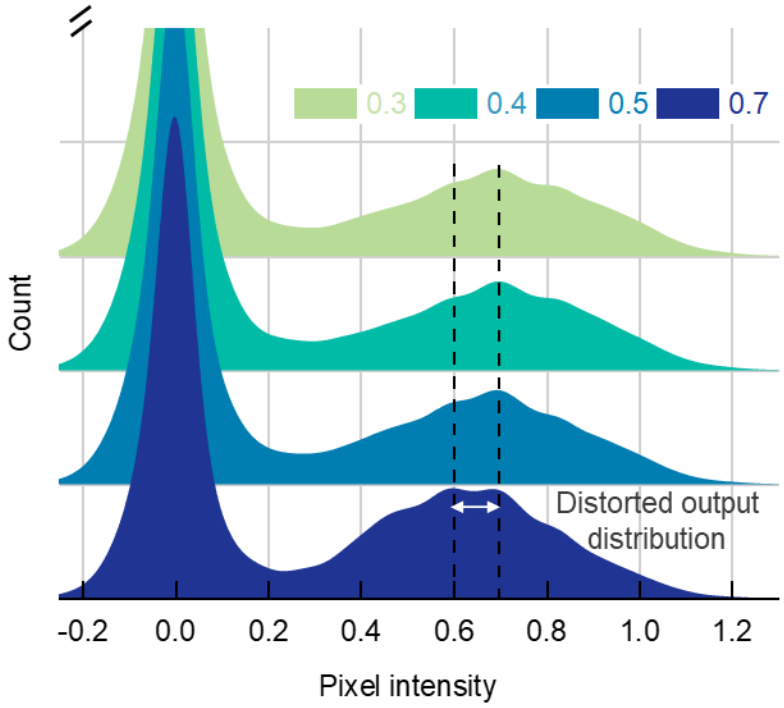
We benchmarked a multilayer perceptron (MLP)-based vanilla autoencoder model, a representative machine learning-based reconstruction model. The model implemented through RRAM crossbars incorporates 784 input neurons, 64 latent space, and 784 output neurons (784–64–784) without bias for 10 Fashion-MNIST images (d = 784) shown in Fig. 3. Note that the RRAM-based autoencoder requires a differential pair to represent negative weight values. The corresponding N_{total} of DIM-SDAM and autoencoder is 3.136×10^4 OPs and 2.007×10^5 OPs, resulting in 6.4× energy efficiency in DIM-SDAM compared to RRAM-based autoencoder.



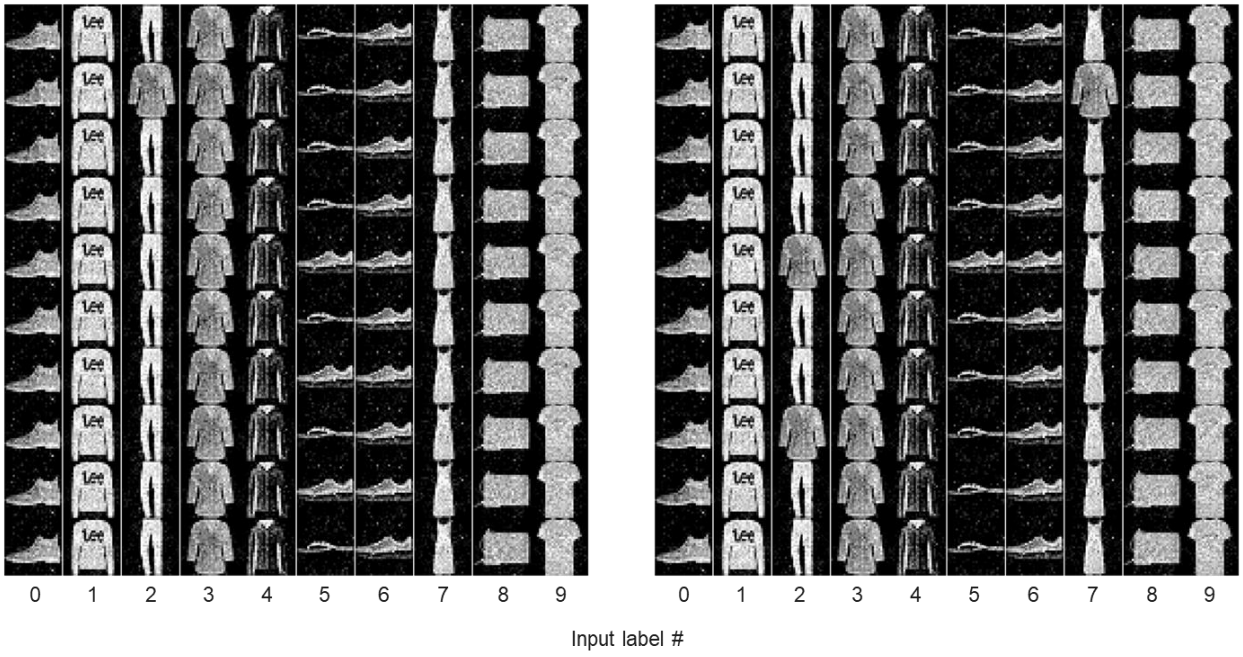
Supplementary Figure. S1 2D energy landscape and convergence with β = 1, 10, and 1000. Three random stored patterns were generated with all positive values. Each landscape of dot product and distance is derived from $\mathcal{E}_{\mathrm{SDAM}}(x)$ and $\mathcal{E}_{\mathrm{DIM-SDAM}}(x)$, respectively. The higher β value guarantees sharpening energy landscape for the convergence to the correct stored patterns.



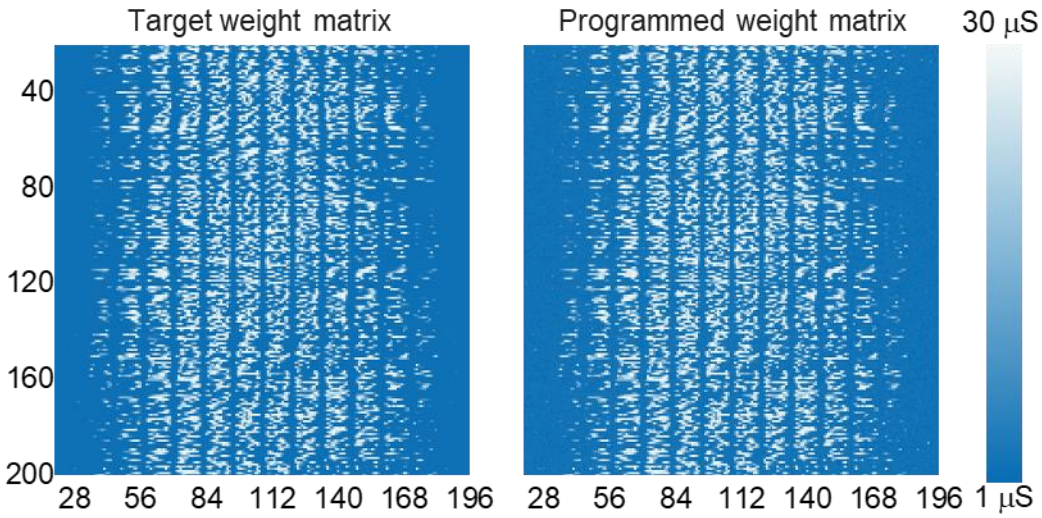
Supplementary Figure. S2 Comparison of target conductance and measure conductance during single cell programming. Each conductance points were measured at readout after 100 pulses.



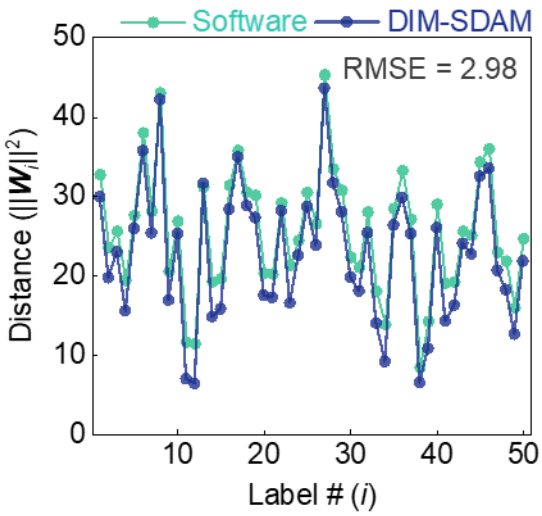
Supplementary Figure. S3 Distribution of output pixel intensities via DIM-SDAM. The large noise levels distort the distribution, resulting in degrading the reconstruction accuracy.



Supplementary Figure. S4 Examples of experimentally reconstructed images with the noise level of 0.5. The left and right image matrix correspond to each batch of 50 images where column refers to each category of Fashion MNIST.

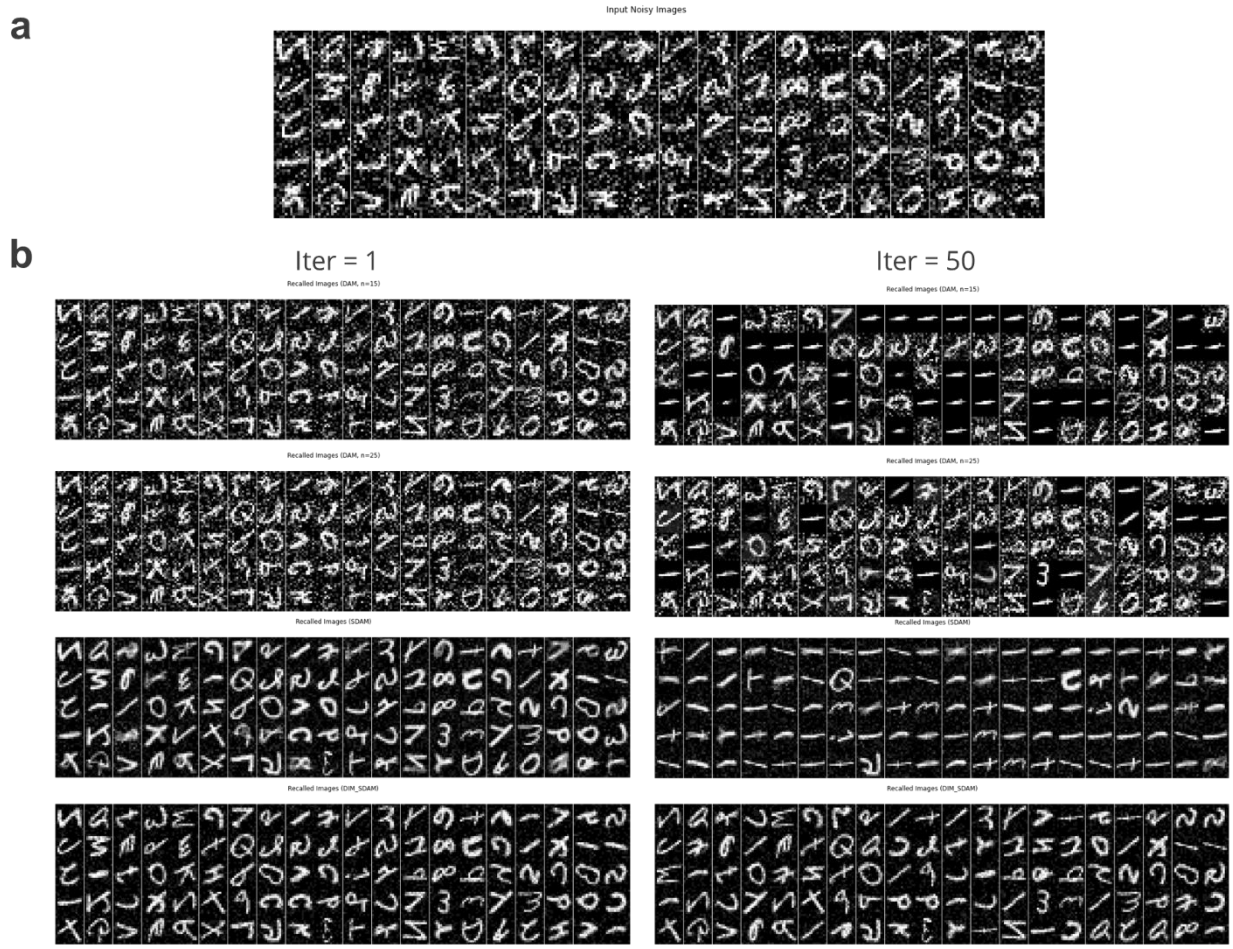


Supplementary Figure. S5 Results of programming weight matrix for EMNIST. 200 × 196 memristors were used to represent 200 EMNIST patterns to be stored.

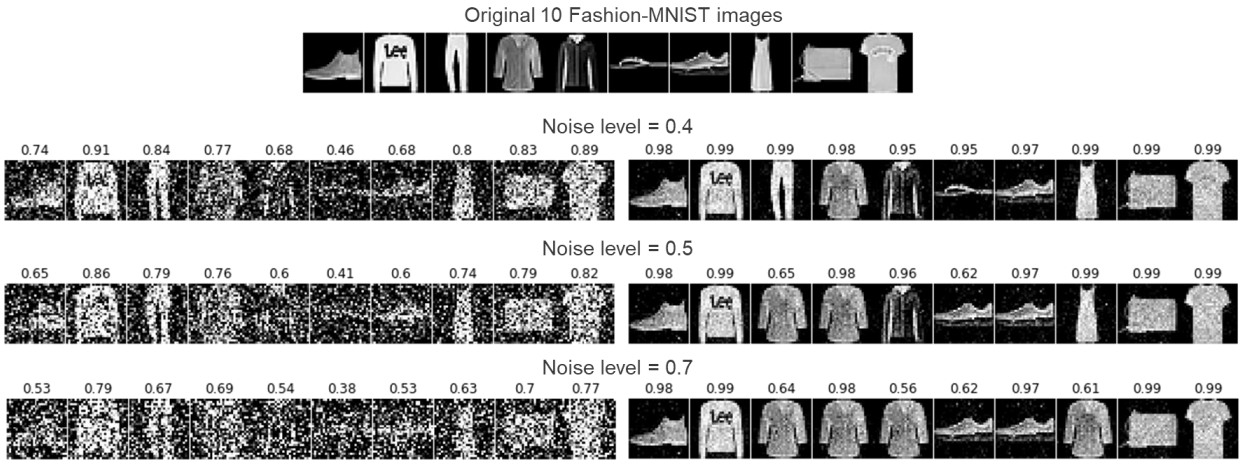


Supplementary Figure. S6 Row-wise norms of stored patterns through software and DIM-SDAM.

The norms are computed only once and reused for computing distance similarity.



Supplementary Figure. S7 Simulated reconstruction results of dense associative memory (DAM) and DIM-SDAM. a Corrupted input EMNIST (100 images) **b** Corresponding reconstruction results at 1st iteration (left) and 50th iteration (right).



Supplementary Figure. S8 Examples of cosine similarity between input and output. The 10 left and right images are corrupted input images and output of DIM-SDAM, respectively. The cosine similarity between one of these images and the corresponding original image is shown as label at each image.

Supplementary References

1. Wan, W. *et al.* A compute-in-memory chip based on resistive random-access memory. *Nature* **608**, 504–512 (2022).
2. Yan, X. *et al.* Reconfigurable Stochastic neurons based on tin oxide/MoS2 hetero-memristors for simulated annealing and the Boltzmann machine. *Nat. Commun.* **12**, 5710 (2021).
3. Guo, K. *et al.* Neural network accelerator comparison. *[Online]. Available:* (2021).
4. NVIDIA GeForce RTX 5090 vs RTX 4090: Specs & Performance. https://boxx.com/blog/hardware/nvidia-geforce-rtx-5090-vs-rtx-4090.
5. Taka, E., Gourounas, D., Gerstlauer, A., Marculescu, D. & Arora, A. Efficient approaches for gemm acceleration on leading ai-optimized fpgas. in *2024 IEEE 32nd annual international symposium on field-programmable custom computing machines (FCCM)* 54–65 (IEEE, 2024).
6. Song, W. *et al.* Programming memristor arrays with arbitrarily high precision for analog computing. *Science (1979).* **383**, 903–910 (2024).
7. Ni, K. *et al.* Ferroelectric ternary content-addressable memory for one-shot learning. *Nat. Electron.* **2**, 521–529 (2019).