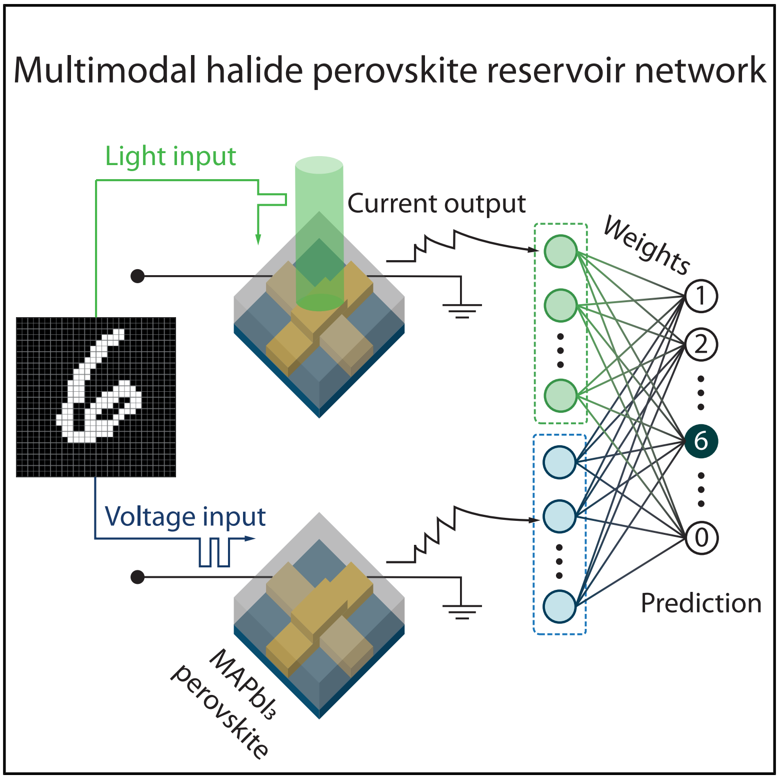
Device

Article

In-sensor computing with halide perovskite-based optoelectronic reservoir networks

Graphical abstract



Highlights

- Ionic currents in halide perovskite devices can be leveraged for short-term memory
- Voltage and light biases cause ion migration on complementary timescales
- An in-sensor multimodal reservoir network reaches high classification accuracies
- The microscale architecture enables dense integration in intelligent sensor arrays

Authors

Jeroen J. de Boer, Agustin O. Alvarez, Moritz C. Schmidt, Bruno Ehrler

Correspondence

b.ehrler@amolf.nl

In brief

J.J. de Boer et al. demonstrate microscale optoelectronic halide perovskite devices capable of combining sensing and computing. The device is easily scalable and can simultaneously process light and electrical signals. The efficient transport of ionic charge carriers in halide perovskites ensures energy efficiency, and the strong light absorption enables multimodal operation. Combining the two input modes, light and electricity, enables higher classification accuracies of images and videos directly at the sensor when compared to using only a single mode.

2
Explore
Early prototypes with exciting performance
and new methodology

de Boer et al., 2026, Device *4*, 101000
February 20, 2026 © 2025 The Author(s).
Published by Elsevier Inc.
https://doi.org/10.1016/j.device.2025.101000

CellPress

Device

CellPress
OPEN ACCESS

Article

In-sensor computing with halide perovskite-based optoelectronic reservoir networks

Jeroen J. de Boer,[1] Agustin O. Alvarez,[1] Moritz C. Schmidt,[1] and Bruno Ehrler[1,2,*]
[1]LMPV-Sustainable Energy Materials Department, AMOLF, 1098 XG Amsterdam, the Netherlands
[2]Lead contact
*Correspondence: b.ehrler@amolf.nl

https://doi.org/10.1016/j.device.2025.101000

THE BIGGER PICTURE Detecting and classifying data, for example, from video cameras, is a key capability for many applications using artificial intelligence, including robotics, self-driving cars, image detection, and biometrics. However, the impressive progress in the capabilities of artificial intelligence comes at the cost of rapidly increasing energy consumption. A large contributor to this energy consumption is the transfer of data from sensors to processors in order to detect or classify the input data. In this work, we demonstrate a microscale halide perovskite semiconductor device that simultaneously senses and processes information. The information can be provided as electrical or optical input, and we show that the classification accuracy is highest if the two inputs are combined. This resembles how the brain merges information from, for example, sight and touch to gain a better understanding of the world.

SUMMARY

Physical reservoir computing can provide efficient neuromorphic in- and near-sensor computing applications. Typically, reservoir networks are designed to process light or voltage inputs. Here, we demonstrate a multimodal optoelectronic reservoir network based on halide perovskite semiconductor devices capable of processing voltage and light inputs, which is also scalable for constructing high-density sensor arrays. The devices consist of micrometer-sized, asymmetric crossbars covered with a methylammonium lead iodide ($MAPbI_3$) perovskite film. Using 4-bit inputs and linear readout layers for classification, we demonstrate multimodal networks capable of processing both voltage and light inputs. The networks reach mean accuracies up to 95.3% ± 0.1% and 87.8% ± 0.1% for image and video classification, respectively. The networks significantly outperformed linear classifier references by 3.1% for images and 14.6% for video. We show that longer retention times benefit classification accuracy for single-mode networks and give guidelines for choosing optimal experimental parameters.

INTRODUCTION

While upscaling of neural networks has resulted in impressive increases in their capabilities, it has also led to an exponential rise in energy consumption.[1] Neuromorphic hardware neural networks inspired by the brain are an appealing, more energy-efficient alternative.[2,3] Brain-inspired networks that process inputs close to the sensor are interesting for reducing inefficient data transfer and power consumption.[4] Physical reservoir computing is a compelling approach for this purpose, as it leverages device-intrinsic dynamics to preprocess inputs. In reservoir computing, a dynamical system nonlinearly transforms the input to increase linear separability,[5] after which a simple linear readout layer is used for classification.[6,7] As only the simple linear readout layer is trained, these networks are easier to implement in hardware than neuromorphic networks with many complex trained layers.[2,8]

Physical reservoirs for in-sensor computing commonly apply the "single dynamical node" approach,[7] shown in Figure 1A. A time-dependent input, $\mathbf{u}(t)$, is fed into the device ("reservoir node"), which changes its state, $\mathbf{x}(t)$. The reservoir node has a volatile memory, so that its state depends on both the presented input and its history. The reservoir node states are collected over time, and a linear weighted sum with the weight matrix $\mathbf{W}$ is taken to yield the final output $\mathbf{y}(t)$. For in-sensor computing, this approach is typically extended to arrays of reservoir nodes,[9] as illustrated in Figure 1B. After the input $\mathbf{u}(t)$ is transformed, the final output $\mathbf{y}$ is obtained by taking a weighted sum of the reservoir states $\mathbf{x}$. This can be implemented by running the readout

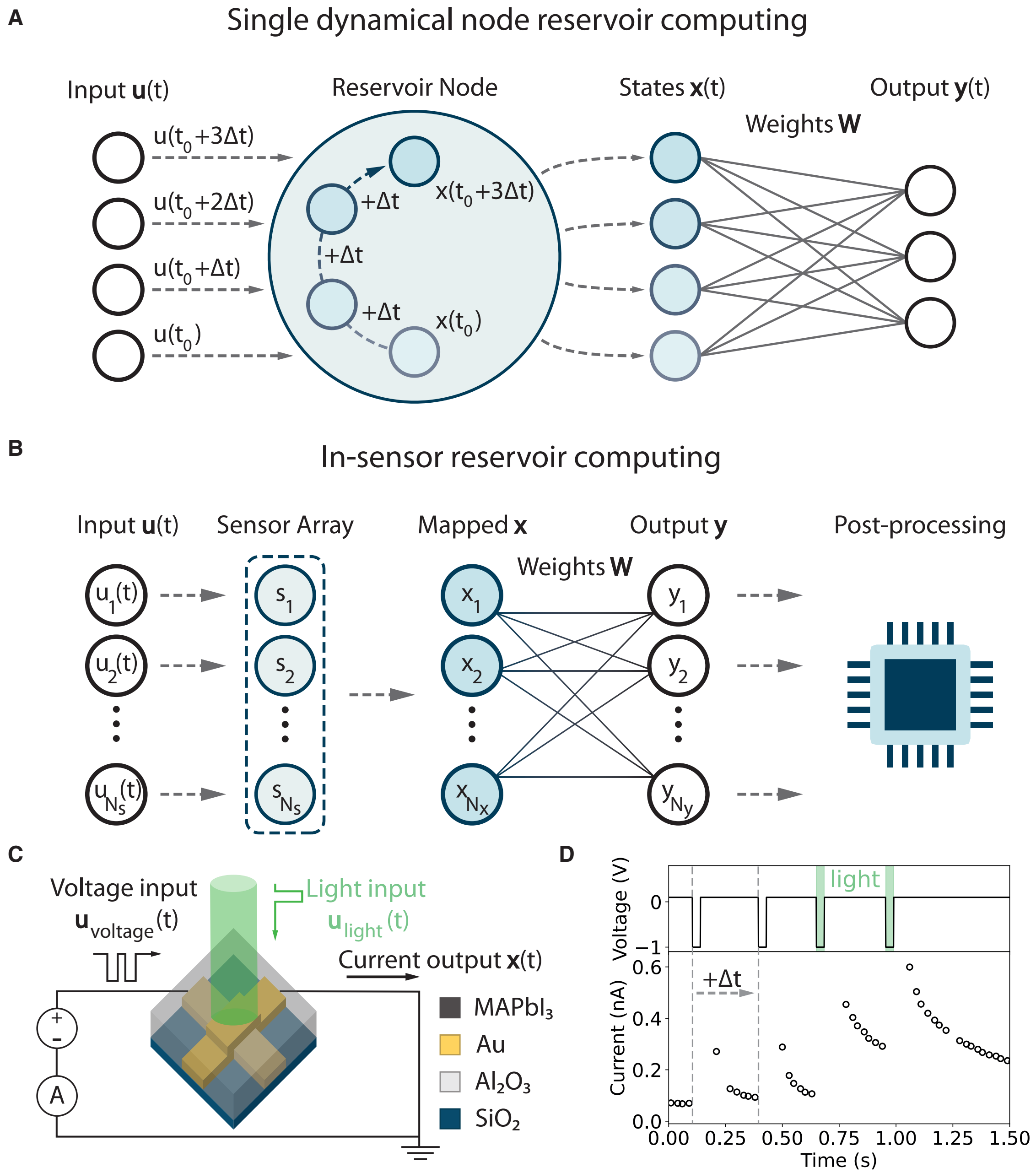


Figure 1. Reservoir computing with volatile artificial synapses

(A) Schematic representation of reservoir computing with a single dynamical node. This node can be implemented with a volatile device ("reservoir node") processing a time-encoded input, represented by input vector **u**(t). Each element of **u**(t) is an input at time t.

(B) Schematic of in-sensor reservoir computing with an array of volatile devices.

(C) Schematic drawing of the volatile optoelectronic halide perovskite artificial synapse. The microscale device can process both voltage and light inputs (**u**(t)) and gives a current as an output (**x**(t)).

(D) An example pulsed voltage measurement showing a slow current decay over hundreds of milliseconds after an applied pulse. Currents are recorded at a +100 mV offset. The currents measured during the pulses are omitted for clarity.

reservoir states through a resistive or memristive device array (weights **W** in Figure 1B).[9] Optionally, the output can be collected for processing on digital platforms. These networks do not require external memory if only the final device states are considered, reducing inefficient data transport.[3]

A broad range of platforms is suitable for physical reservoir computing because the reservoir performs a fixed nonlinear transformation of an input. Implementations include memristive devices,[9,10] photonic circuits,[11,12] and spintronic devices.[13] The ability of optoelectronic devices to process both voltage and light inputs makes them attractive for in-sensor reservoir computing.[14] By combining inputs from the two sensing modes, these devices can encode a richer representation of the input data to increase classification accuracy.[15] Halide perovskites have excellent optoelectronic properties and are well suited for reservoir computing. Next to strong light absorption, they feature complex and tunable transient behavior due to ion migration induced by a light or voltage bias.[16] Several reports have explored the use of halide perovskites for reservoir computing.[17–20] However, in these implementations, the reservoir was limited to detecting either a voltage or a light-based input, and multimodal sensing was not explored. Despite these reports, the difficulty of microfabrication of halide perovskite devices, which is necessary for high-density integration, has not been addressed.[21]

Here, we address these gaps by implementing the microscale halide perovskite devices we recently developed as optoelectronic artificial synapses[22] for reservoir computing. After applying a voltage pulse, the devices outputs an ionic displacement current that decays on the seconds-to-hundreds of milliseconds timescale. Illuminating the device when a voltage is applied increases the current modulation. We investigate the linear separability of 4-bit light and voltage inputs based on this volatile current. Based on the results, we simulate in-sensor reservoir computing networks consisting of arrays of the device that transform and classify handwritten digits from images (MNIST) and video (modified N-MNIST) while accounting for experimentally measured noise. We demonstrate that combining light and voltage inputs in the same multimodal network increases the classification accuracy compared to networks that implement one modality. The multimodal networks reach classification accuracies up to 95.3% ± 0.1% for image and 87.8% ± 0.1% for video datasets, surpassing linear classifier references. When considering only one type of input, light-input networks outperformed networks based on voltage inputs. We show with simulations that this is due to the shorter retention time relative to the input frequency for voltage inputs. The simulations allow facile estimation of network accuracies based on the 4-bit input measurements, valuable when fine-tuning experimental parameters. They show complementary transformations by the light and voltage networks, leading to the high accuracy of the multimodal networks. Our results demonstrate the potential of halide perovskite volatile devices for efficient, multimodal in-sensor computing.

RESULTS AND DISCUSSION

Design of the halide perovskite device

A schematic image of the microscale halide perovskite device is shown in Figure 1C. The device consists of 2.5-μm-wide, back-contacted cross-point electrodes of gold that sandwich an insulating Al_2O_3 layer. A $MAPbI_3$ layer is spin-coated over the cross-point electrodes. We have previously developed this device architecture to prevent degradation of the perovskite layer during microfabrication.[22,23] A $MAPbI_3$ film fabricated in the same fashion was characterized in previous work on macroscale devices, where we extracted a grain size on the order of 100 nm, a defect density of approximately 2×10^{18} cm^{-3}, a diffusion coefficient of 3.96×10^{-11} cm^2/s at 300 K, and an activation energy for ion migration of 0.38 ± 0.03 eV.[24]

The example of a pulsed voltage and light measurement in Figure 1D shows how the device could be used as a reservoir node. Four -1 V pulses, corresponding to the four inputs at different times in Figure 1A, are applied to the device. The current, resembling the readout states $\mathbf{x}(t)$, is measured continually. Each combination of a -1 V pulse and the subsequent dwell time, during which the current is measured, represents one Δt time step from Figure 1A. We use Δt to refer to the time steps instead of the more conventional "τ" to avoid confusion with the characteristic decay time of the current.[25] Applying -1 V pulses results in a current that decays slowly over hundreds of milliseconds after each pulse. The current recorded after the second pulse (0.29 nA) is slightly larger than after the first pulse (0.27 nA). Higher current changes are obtained after the third (0.40 nA) and fourth (0.60 nA) -1 V pulses, during which the device is simultaneously illuminated. This difference demonstrates that the device's response to a new input, i.e., its readout state, depends on both the input itself and the history of previous inputs, a requirement of reservoir computing. The voltage and light pulses affect the current in distinct ways.

Previously, we have shown that the current decay of this device after a voltage pulse is governed by a combined ionic drift and diffusion process.[22] Here, the decay follows the same ionic drift and diffusion processes, as follows from the fit in Figure S1. Drift-diffusion simulations in Figure S2 corroborate that the transient current response is due to an ionic displacement current. These currents are attributed to a combination of ion accumulation or redistribution at the perovskite-electrode interface and ionic redistribution toward the bulk.[26,27] The increase in current after each pulse is due to further accumulation of ions. Illumination during the -1 V pulse likely enhances the accumulation due to the higher ionic conductivity in light,[28] in line with our previous work.[22]

Using this volatile current for reservoir computing requires a linearly separable output for different inputs. We investigate this linearity first for voltage inputs without illumination. Next, we explore it for inputs combining voltage pulses with light inputs.

Voltage input measurements

The voltage input measurements are shown in Figure 2A. Input sequences were provided in four time steps with a 150 ms duration (Δt in Figure 1D). A -1 V pulse can be applied during the first 50 ms of each time step. Next, the current is measured for 100 ms, always without applied voltage. The four time steps allow 16 different voltage-input sequences. These inputs can be represented as binary numbers, where a time step with an applied -1 V pulse is denoted as a binary "1" and a time step without applied voltage as a binary "0".

An example measurement with a 1011 input sequence (input $\mathbf{u}(t)$ in Figures 1A and 1B) is shown in Figure 2B. Three -1 V pulses are applied to the device, with a missing -1 V pulse at the second time step, shown as a dotted line. The currents collected in each time step, corresponding to the readout states $\mathbf{x}$ in Figures 1A and 1B, are referred to as I_1, I_2, I_3, and I_4 in the plot. The current is increased after each pulse and decays in the absence of an applied voltage, consistent with an ionic displacement current.

Please cite this article in press as: de Boer et al., In-sensor computing with halide perovskite-based optoelectronic reservoir networks, Device (2025), https://doi.org/10.1016/j.device.2025.101000

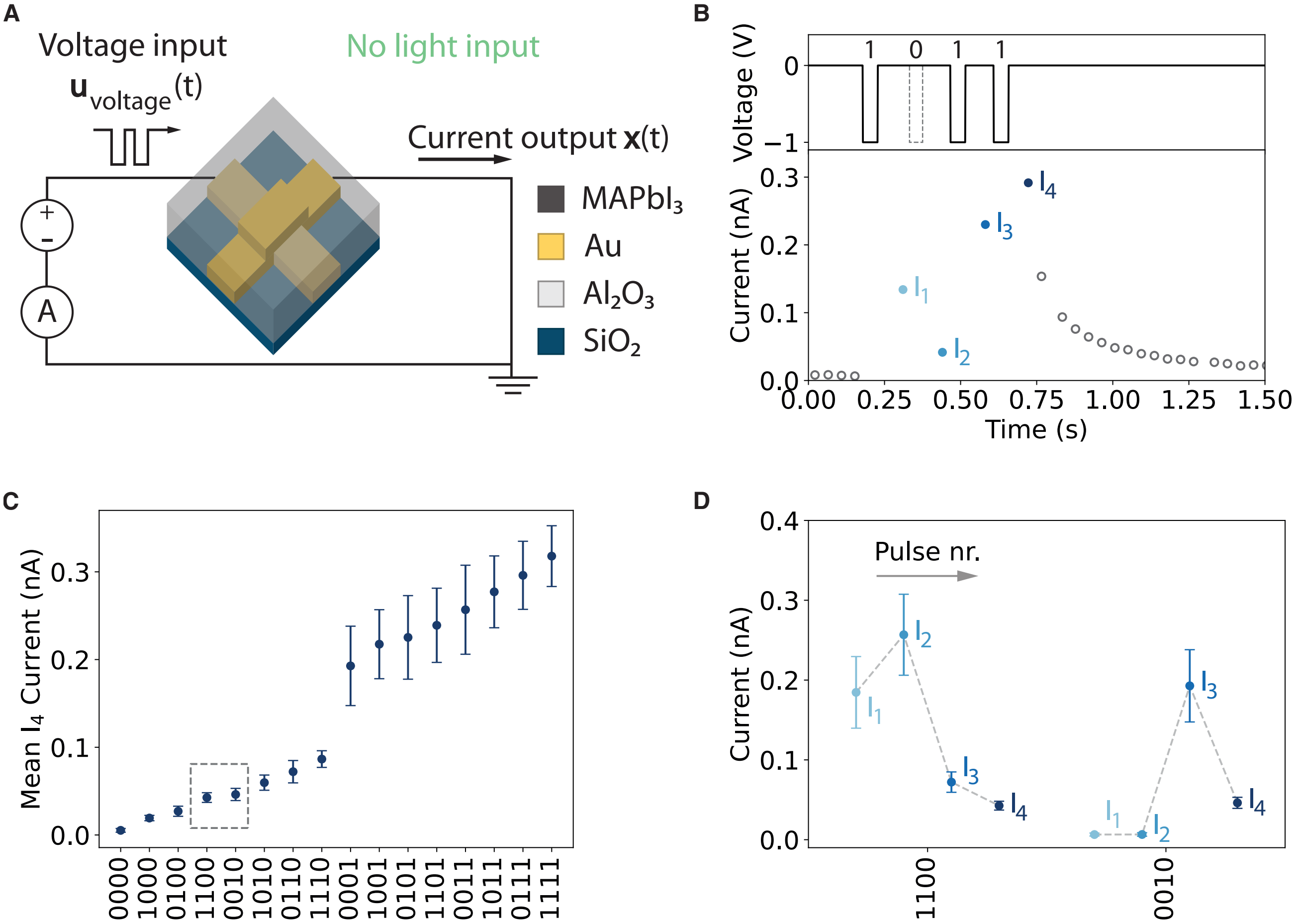


Figure 2. Electronic measurements of 4-bit voltage profiles

(A) Schematic of the measurements. A 4-bit, −1 V pulsed voltage input is applied, and the resulting current is measured.

(B) An example measurement of three −1 V pulses, shown at the top. The dotted line after the first pulse indicates a missing −1 V pulse for that Δt time step. Hence, this voltage profile corresponds to a 1011 input. The bottom shows the measured currents. Currents after each of the four pulses used for further analysis (I_1, I_2, I_3, and I_4) are shown in blue.

(C) Mean with standard deviation of I_4 currents for each 4-bit input. Output currents are higher for inputs with more −1 V pulses and for inputs with pulses applied later in the 4-bit input. Some means are close to overlapping, such as those of the 1100 and 0010 inputs highlighted by the dashed gray box.

(D) Mean with standard deviation of I_1, I_2, I_3, and I_4 currents of the highlighted 1100 and 0010 inputs. While the I_4 currents are similar, the I_1, I_2, and I_3 currents are easily separable.

Each of the 16 possible 4-bit sequences was measured 100 times. The separability of the 16 inputs was investigated by comparing the means and standard deviations of the I_1, I_2, I_3, and I_4 currents.

Figure 2C shows the obtained mean and standard deviation of the I_4 currents. As expected for an ionic displacement current, the I_4 current increases as a larger number of voltage pulses are applied and when pulses are applied for later bits. Some input sequences lead to similar currents, such as the highlighted 1100 and 0010 sequences. Figure 2D shows that the mean I_1, I_2, and I_3 currents are easily separable. The mean currents after each bit of all inputs are given in Figure S3. Even though the standard deviations of several means overlap, the standard errors of the means are small, as demonstrated by Figure S4. This implies that the means are well defined. Stable currents were obtained throughout the measurements, as indicated by the cycle-to-cycle plot in Figure S5, underlining the cycling stability of the device. Device-to-device variability was determined by taking 10 measurements of the 0001 input over 10 devices. The means of the obtained I_4 currents are plotted in Figure S6. The means fall within a standard deviation of one another, suggesting high device-to-device uniformity. The mean energy consumption of an input bit was 61 ± 6 pJ, determined by taking the mean of the product of the voltage, output current, and pulse duration over all input pulses.

For in-sensor reservoir computing applications, it is important that values from each distribution are different. This difference can be determined from the overlap of the probability mass of the distributions, i.e., their overlap coefficient.[29] The overlap coefficients represent the fraction of common random samples of

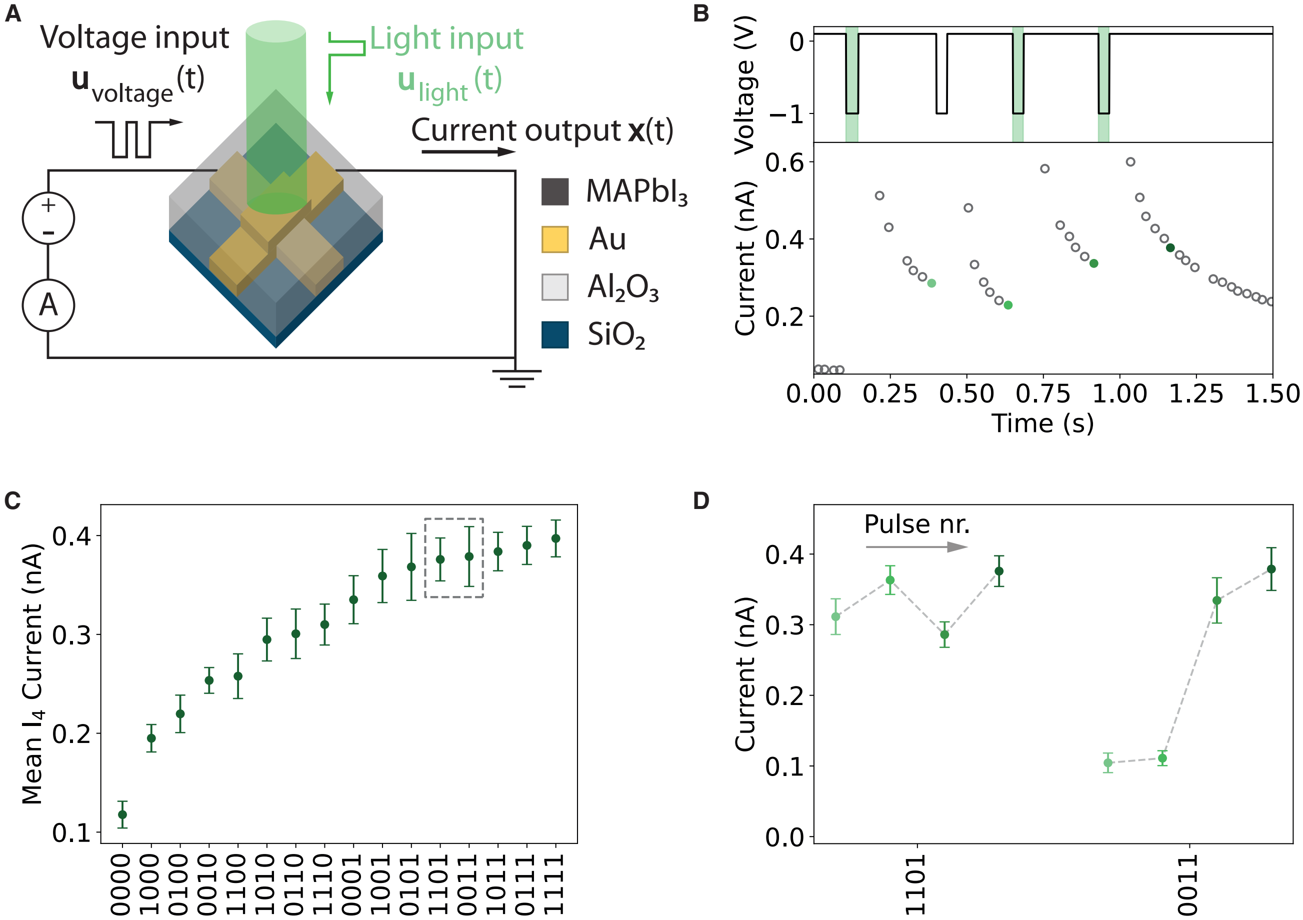


Figure 3. Optoelectronic measurements of 4-bit light pulse inputs

(A) Schematic of the measurements. Periodic −1 V pulses are applied to the device. Four-bit light input sequences are applied simultaneously with the −1 V pulses.

(B) An example measurement of the 1011 sequence is shown as green-shaded regions in the top graph. Measured currents are shown at the bottom and are recorded at a +100 mV offset. The I_1, I_2, I_3, and I_4 currents used for further analysis are shown in green.

(C) Mean with standard deviation of I_4 currents for each 4-bit input. The mean currents are higher for inputs where more light pulses are applied and where these pulses are applied for later bits. The gray dotted box highlights the 1101 and 0011 inputs as an example of inputs with similar means.

(D) The I_1, I_2, I_3, and I_4 currents of the 1101 and 0011 inputs highlighted in (C). While the I_4 currents are similar, the I_1, I_2, and I_3 currents are noticeably different.

two distributions. Large overlap coefficients potentially indicate that similar currents will be obtained for different inputs. The overlap coefficients of the I_4 current distributions are given in Figure S7A. While 68 of the 120 overlap coefficients are negligible, below 1%, 39 inputs have noticeable overlap coefficients of 10% or higher. These inputs could be confused if only one sample is provided to the network, potentially reducing its accuracy. The overlap of the distributions can be reduced by mapping the inputs to both the I_2 and the I_4 currents, as demonstrated by Figure S7B. In this case, 106 overlap coefficients are below 1%, and only 9 overlap coefficients above 10% are found. Figure S7C shows that mapping to all four currents results in negligible overlap coefficients for all inputs. Nonetheless, an important caveat is that mapping to multiple currents requires additional memory elements in the in-sensor computing array, increasing device complexity and reducing energy efficiency.

Light input measurements

We investigate the separability of the input sequences using light in addition to voltage pulses. A schematic of the light input measurement is given in Figure 3A. Each time step (Δt in Figure 1D) for the light inputs is 270 ms. At the start of each time step, a −1 V pulse is applied for approximately 40 ms. Afterward, the voltage is changed to a constant +100 mV, and the current is measured. The voltage pulses set the input frequency (Δt) of the device. Light pulses can be applied simultaneously with the −1 V pulses. Applying a light pulse during the −1 V pulse is considered a binary "1", while no applied light during the −1 V pulse is represented by a binary "0". The device was not illuminated at any other part of the time step. In our current implementation, it was not possible to induce a volatile current by applying light pulses without a bias voltage. This suggests that illumination does not generate a photovoltage that can drive ion migration. Fabricating devices with electronically asymmetric electrodes could remedy this limitation.[19]

An example measurement of a 1011 input sequence (**u**(t) in Figures 1A and 1B) is shown in Figure 3B. Similar to the example voltage input measurement in Figure 2B, the current is increased after each −1 V pulse. When the device is illuminated during the voltage pulse, the current increase is enhanced, in accordance with our previous work.[22] This enhancement can be explained by the higher ion mobility in the perovskite layer under illumination. The current seems to decay more slowly for the light inputs compared to the voltage input in Figure 2B. Fits to the current decay after the 0000 and 0001 input sequences in Figure S8 show that this is due to a shift from a faster drift to a slower diffusion decay. This trend indicates that accumulated ions experience a weaker electric field after light pulses are applied. A possible explanation might be a flattening of the electronic bands as the perovskite layer is illuminated, due to the increase in electronic charge carrier density.[30] Analogous to the voltage sequences, the separability of each of the 16 possible inputs was investigated based on the means and standard deviations of the I_1, I_2, I_3, and I_4 currents (**x** in Figure 1B).

The means and standard deviations of the I_4 currents for each input sequence are given in Figure 3C. Higher currents are obtained when a larger number of light pulses are applied and when the pulses are applied in later time steps. Similar to the voltage inputs, some sequences for the light inputs show comparable I_4 current. The constant 100 mV offset we use reduces the overlap somewhat. Figure S9 shows the mean I_4 currents if no offset was applied, resulting in more similar values. Nonetheless, several means are closely spaced, such as those of the highlighted 1101 and 0011 input sequences. Despite these similar mean I_4 currents, the I_1, I_2, and I_3 currents are more easily separable, as demonstrated by Figure 3D. The mean I_1, I_2, I_3, and I_4 currents and standard deviations of each 4-bit light sequence are given in Figure S10. Similar to the 4-bit voltage sequence measurements, the standard errors of the means are small, as illustrated by Figure S11. A plot of the cycle-to-cycle stability is given in Figure S12. These, combined with the voltage-input cycle-to-cycle stability measurements of the same device (Figure S5), demonstrate its stability over at least 10^3 cycles. The mean energy consumption of an input bit was 584 ± 125 pJ, determined by taking the mean of the product of the voltage, output current, and pulse duration over all input pulses. Overlap coefficients are given in Figure S13. Of the 120 coefficients, 55 are 10% or larger when mapping to only the I_4 current. High overlap coefficients are found particularly for inputs containing three or four light pulses. Similar to the voltage inputs, overlap is reduced by mapping to both I_2 and I_4, with only 18 significant overlap coefficients. Again, mapping to all four currents yields no significant overlap for any combination of inputs.

Image-based handwritten digit classification

We implemented reservoir networks based on either the voltage or the light sequences for MNIST handwritten digit classification to benchmark network performance. Each sample of the binarized MNIST dataset was divided into 2 × 2-pixel patches (‘‘Square’’), 4-pixel rows (‘‘Row’’), or 4-pixel columns (‘‘Column’’). These 4-pixel arrays were then converted to 4-bit binary sequences. White pixels were interpreted as a “1” (an input voltage or light pulse) and black pixels as a ‘‘0’’ (no input pulse). The sequence was constructed by combining the obtained binary values in the order denoted in Figure 4A. Next, we define a mapping $f{:}A \rightarrow B$, where A is any of the 16 possible 4-bit sequences, obtained from the image, and B is the corresponding I_4 current from Figure 2C (voltage input) or Figure 3C (light input). An example mapping of a 2 × 2-pixel square is shown in Figure 4A. The square is converted to the 1010 pixel sequence, which is mapped to the I_4 current of the 1010 voltage (Figure 2C) or light input (Figure 3C). In a physical implementation, each square, row, or column would contain the input to one device in the sensor array. Inputs are separately fed into each pixel in the array. The 4-bit sequences would then be applied as in Figure 2B or Figure 3B, and afterward, the I_4 current of each device in the array would be collected for readout. This method of temporally encoding segments of an image is used in reservoir computing.[31] The purpose of the reservoir in this application is to correlate the pixel values to extract features in the images important for their classification. The Square, Row, and Column approach to transforming the images will therefore lead to the extraction of features in one (Column and Row) or two dimensions (Square). Here, we simulate the reservoir as an array of devices with outputs as characterized in Figure 2C for voltage and in Figure 3C for light inputs.

Figure 4B shows the MNIST sample after mapping all squares to the corresponding I_4 currents of the 4-bit voltage and light inputs. The transformed images correspond to the mapped vector **x** in Figure 1B, where the I_4 currents represent the states collected from each node. We account for experimental noise in the transformation by mapping each square to a random number taken from a normal distribution with the determined I_4 current mean and standard deviation for that sequence. Brighter pixels, corresponding to higher I_4 currents, are obtained for the light-transformed images compared to voltage inputs. Thus, compared to the voltage-input-based transformations, pixel values (currents) from earlier inputs are retained to a greater extent for the light-transformed images, in line with the example 1010 mapping in Figure 4A. The longer retention can be explained by the shift to slower current decay by ionic diffusion after light inputs (Figure S8), resulting in a longer memory window.

Each image in the MNIST dataset was transformed by the Square, Row, or Column mapping approach. Linear readout layers were trained on the transformed datasets. Figure 4C shows the obtained classification accuracies over 10 independent runs for different transformations. The networks are compared to reference networks trained on the binarized MNIST dataset, for which each square, row, or column was mapped to the binary value of the fourth pixel. These reference networks are equivalent to using a regular, memory-less sensor array (**s**), such as a camera, in combination with a resistor array weight matrix (**W**). Comparing the reservoir network accuracies with the reference, therefore, allows an accurate assessment of the contribution of the reservoir. The accuracies are also compared to a linear classifier trained directly on the binarized MNIST dataset. Reservoir transformations increase the linear separability of the dataset if reservoir network accuracies exceed that of this reference.[5]

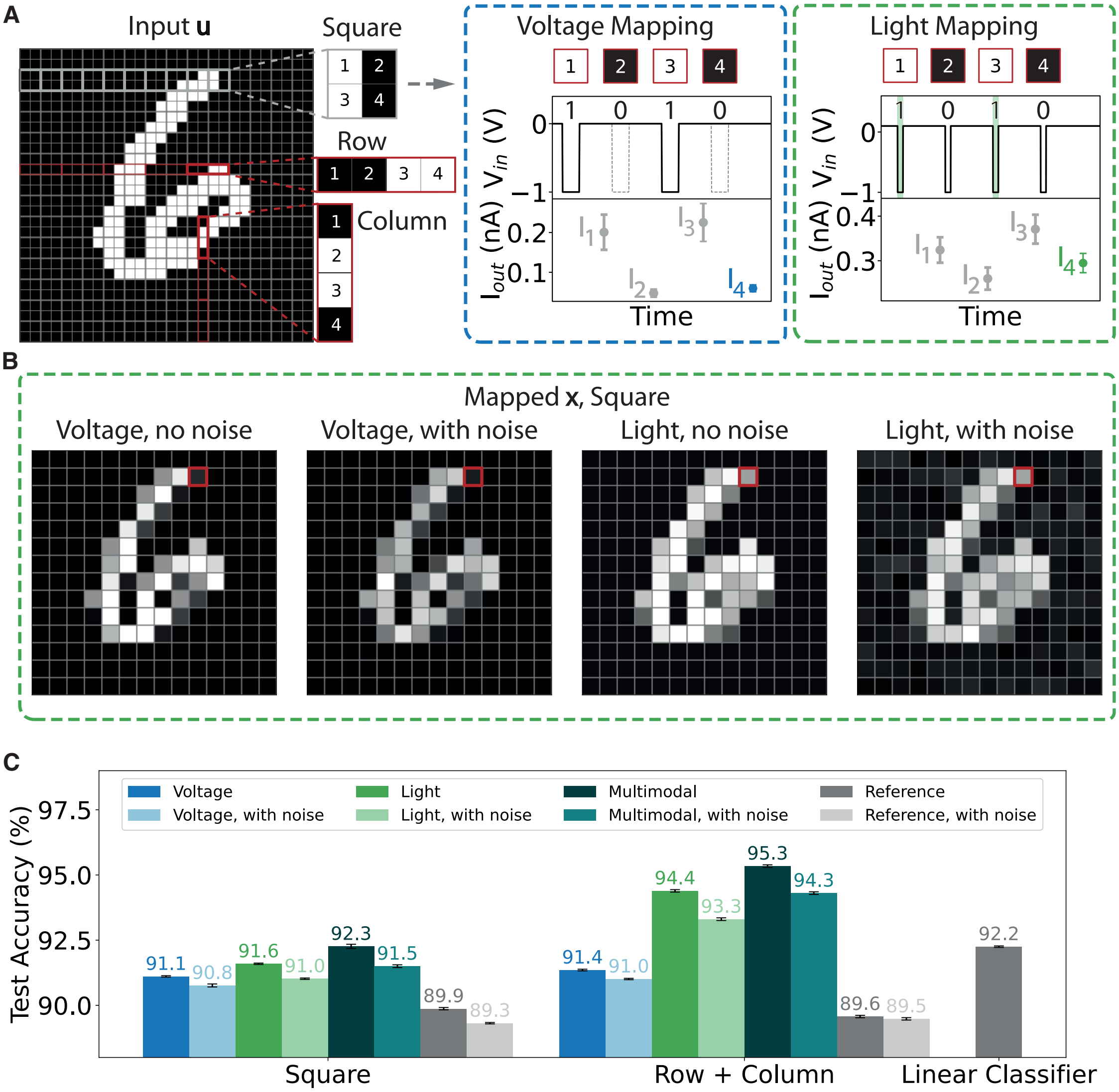


Figure 4. MNIST classification with images transformed based on light and voltage inputs

(A) Example transformation of a binarized MNIST image of a number 6. The image is divided into 2 × 2-pixel squares ("Square"), 4-pixel rows ("Row"), or 4-pixel columns ("Column"). Pixels are laid out in a row from the pixels labeled "1" to "4". White pixels are interpreted as a "1" (voltage or light pulse input) and black pixels as a "0" (no input). The obtained 4-bit binary sequence is matched to experimental voltage or light inputs, shown for the 1010 sequence of the square example. The square, row, or column is converted (mapped) to the I_4 current of that sequence.

(B) Example transformation based on square mapping to the voltage and light input data. In the "no noise" transformations, the 2 × 2-pixel squares were mapped to the mean I_4 current of the sequences. Noise was included in the "with noise" case as described in the methods section. The pixel highlighted in red corresponds to the red patch in (A).

(C) Mean classification accuracies and their standard deviation of datasets transformed by mapping the images with the Square or the combined Row and Column approach to the I_4 currents of the voltage, the light, or a combination of both (multimodal) measurements.

For the Square mapping approach, the light-based networks outperformed those based on voltage inputs (0.5%, $p < 0.001$). We show in Note S1 with simulations that this is due to the longer memory window of light inputs, which results in higher contrast for patches at the edges of the digits after the transformation. The ratio of the retention time to the input frequency of the light inputs is close to the optimum in the simulations. Conversely, the shorter relative memory window of the voltage-based networks provides higher contrast at patches containing many white pixels, which are typically found in the centers of the digits. This is also visualized in Figure 4B. Previously, better performance of reservoir networks was found when combining two sensing modes into a single, multimodal network.[15,32] These multimodal networks combine two nonlinear transformations to increase linear separability. We construct multimodal networks by combining the I_4 currents of the voltage and light networks into a single mapped state vector **x**. The weight matrix is trained on the combined states. Even though the light and voltage networks both encode the same features, we refer to this approach as multimodal because the reservoir is driven by physically distinct inputs.[14,15] A principal-component analysis (PCA) of voltage-input and light-input state vectors is given in Figure S15. The two input modes produce two distinct clusters, indicating that they add complementary information to the reservoir state space. While combining both the voltage and the light modalities of the network increases the energy cost per operation, the light-input network dominates the energy consumption, with 584 ± 125 pJ compared to 61 ± 6 pJ per operation for the voltage-input network.

The multimodal network outperforms all other networks implementing the Square mapping, with a mean accuracy of 92.3% ± 0.1% (all differences $p < 0.001$). The improved performance likely stems from the complementary combination of enhanced contrast at the edge of the digit by the light mapping and at the centers of the digits by the voltage mapping. Taking into account a potential noise comparable to the experimental noise (see the methods section for details) slightly decreases the accuracy of the networks. The accuracy penalty was more severe for the light-input and multimodal networks, likely due to the higher overlap coefficients of the light-input I_4 currents (Figure S13) compared with those of the voltage inputs (Figure S7). In all cases, the reservoir networks outperform the Square reference also when considering the noise. This result shows that the transformations encode information relevant for classification also when considering experimental noise. Without considering experimental noise, the multimodal network with the Square transformation does not significantly outperform the linear classifier ($p = 0.681$).

In previous implementations of in-sensor reservoir networks, the MNIST dataset is typically transformed in a 4-pixel line-by-line fashion (Row or Column in Figure 4A).[15,33–35] We implement the same transformations in Figures S16A, S16B, and S16C. The figure shows that these transformations distort the image more strongly than the Square mapping method we follow in Figure 4. The classification accuracies are given in Figure S16D. The accuracies were slightly higher for the Row mapping, both for the voltage (89.0% ± 0.1%, 88.6% ± 0.1% with noise) and for the light inputs (90.8% ± 0.1%, 90.0% ± 0.1% with noise), but are lower than those obtained for the Square mapping in Figure 4C. This is likely due to a stronger loss of relevant features as the images are compressed in only one dimension. The mean accuracy of the multimodal network was 92.6% ± 0.1% (91.5% ± 0.1% with noise), slightly higher than for the Square mapping ($p < 0.001$). This accuracy exceeds that of the linear classifier ($p < 0.001$), although only if experimental noise is not considered. Similar to the Square mapping, this could be explained by the complementary combination of the light and voltage I_4 current mapping. Furthermore, the 4-pixel rows extend over a larger distance in the image. This might allow the network to better extract relevant features for classification. Notably, the accuracies of the light input and the multimodal networks are higher than those reported in previous work.[15,33–35] For the light-based network, this could be due to a more favorable current output for the input sequences, as explained in Note S1.

Another explanation might be a more thorough hyperparameter search before training the readout layers. For the multimodal network, the accuracy is likely increased by the complementary combination of the light and voltage mapping, in line with previous work.[15,32] Some previously reported accuracies are comparable to or lower than the one-reading reference we report here.[15] This finding highlights the importance of evaluating the reservoir performance in comparison to reference linear classifiers to prevent overestimating the contribution of the reservoir transformations.

A proven way to increase the accuracy of a reservoir computing network is to present the images to the network at different rotations.[31] We implement this here by mapping each digit with both the Row and the Column approaches explained in Figure 4A and concatenating the obtained mapped image vectors to obtain a single combined vector **x** for each image. The readout layers are then trained on the combined Row and Column transformed dataset. Network accuracies are shown in Figure 4C as "Row + Column." As expected, mean accuracies increased compared to those of the Square and separate Row and Column mapping. The highest obtained mean accuracy was 95.3% ± 0.1% for the multimodal network. By combining the Row and Column mappings, the separability of features is increased in both dimensions. Compared to the Square mapping, features are extracted over larger distances in the images (4 instead of 2 pixels in either direction), which can explain the improved performance. Both the light-input and the multimodal networks implementing this mapping approach exceed the linear classifier. Confusion matrices are given in Figure S17. The accuracy of the multimodal network was higher overall compared to the voltage-input network. These networks improve on previously reported in-sensor reservoir implementations[15,33–35] as well as comparable reservoir networks based on various other physical platforms, as summarized in Table S3. The table shows that the energy consumption of a single operation was comparable to or lower than these previous implementations,[9,10,15,35] with the exception of an energy consumption of 100 fJ reported for magnetic tunnel junctions.[36] We note that implementing asymmetric contacts could make the device self-powered when processing optical inputs by inducing a photovoltage,[19] increasing the energy efficiency.

Both the light-input and the multimodal networks outperform logistic regression (92.8% ± 0.01%, Figure S18) also when considering experimental noise. The reservoir networks are outperformed by a single-layer convolutional neural network (98.2% ± 0.01%, Figure S18). Even so, convolutional neural networks are challenging to implement in hardware and are susceptible to device nonidealities.[37] Conversely, more complex reservoir networks, including a larger number of devices, typically benefit from device-to-device variations.[19] Increasing the number of pixels mapped by each device could enhance the network accuracy.[31] The light inputs in particular appear to have a sufficiently long retention time for mapping more than the currently implemented 4 pixels at a time. Previously, applying additional rotations to the MNIST dataset resulted in higher accuracies for software reservoir networks.[31] The same approach might increase the accuracy of our networks as well. Another method to increase the accuracy is to add hidden layers that perform further nonlinear transformations of the data.[19] However, such are difficult to implement in hardware and are therefore less interesting for in-sensor computing.

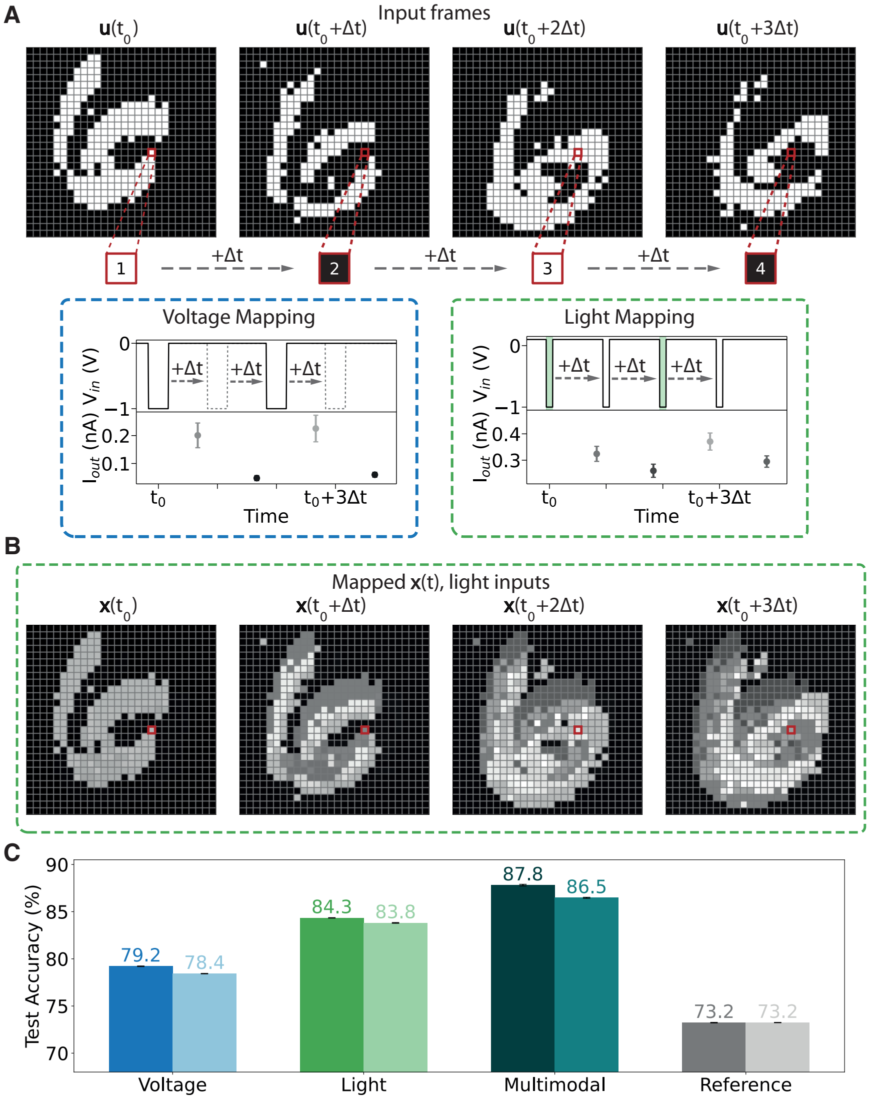


Figure 5. Handwritten digit classification from video with frames transformed based on light and voltage inputs

(A) Four consecutive frames of a number 6. Pixels underneath each frame show the pixel highlighted in red at each time step. For each pixel in the four frames, its four consecutive values are converted to a 4-bit light or voltage pulse input sequence. The sequence is then applied (*in silico*) to the experimentally measured artificial synapse.

(B) Transformations of the frames in (A) after each time step, based on the means of the 4-bit light inputs. Each pixel is mapped following the procedure outlined in (A). The red boxes highlight the same pixel as in the frames in (A).

(C) Classification accuracies and their standard deviation obtained for the reservoir networks based on voltage (blue), light (green), and multimodal (teal) inputs. The accuracies are compared to a reference (gray) trained on the dataset without transformations. Accuracies for networks considering experimental noise are given in lighter colors. Accuracies were obtained by taking the mean of 10 independent training runs.

To disentangle the contributions of the device response to light inputs and the constant offset bias, a light-input network without the +100 mV offset (Figure S9) was trained using the Row + Column approach as well. The classification accuracy was 93.6% ± 0.1% (92.9% ± 0.1% with noise), as shown in Figure S19. The decreased accuracy compared to the light-input network in Figure 4C is likely due to the larger degree of overlap between the I_4 currents. While the lack of the offset bias decreased the accuracy of the network, it still outperforms the voltage-input networks, likely due to the more favorable trend in the I_4 current (Note S1). While the offset bias helps distinguish different inputs more clearly, the main contributor to the higher accuracy of the light-input networks therefore stems from the optical input.

Video-based handwritten digit classification

We investigated the reservoir network performance for temporal inputs relevant to in-sensor computing. Handwritten digit classification from video, based on the N-MNIST dataset,[38] is taken as an example. The light input lends itself well to detecting and transforming visual data by projecting the images on the device array. The voltage implementation could be useful for processing data from a separate sensor detecting, for example, tactile signals.[15] We followed an approach similar to that used above. First, 34 × 34-pixel binned and binarized N-MNIST frames were mapped to the experimental voltage and light data, as shown schematically in Figure 5A. However, instead of squares, rows, or columns, individual pixels of four consecutive frames were mapped to the I_1, I_2, I_3, and I_4 currents.

Figure 5B shows the frames after each pixel is mapped to the corresponding mean currents of the 4-bit light sequence. The number 6 of the previous frames remains visible with lower brightness in each consecutive frame, visualizing that information from previous frames is retained. Figure S20 shows the transformed sample accounting for the experimental noise and the sample transformed based on voltage inputs. In all cases, the number 6 remains recognizable after the transformation. However, the features of previous frames remain brighter for the sample mapped to the light input. This is again a result of the longer retention of information compared to the voltage input, similar to Figure 4.

The dataset was transformed based on voltage- and light-input experimental currents. As in Figure 4, multimodal networks were implemented by combining the voltage- and light-input-transformed frames. Next, linear readout layers were trained on the transformed datasets. A reference network was trained on a dataset that was not transformed. In our implementation, the frames (**u**(t)) are projected on the sensor array (**s**), and the output currents of each device (**x**(t)) are collected continuously. Weighted sums of the currents are taken with the weight matrix (**W**) after each input frame (for each time step Δt) to classify each frame consecutively. Analogous to the MNIST classification in Figure 4, the reference network is equivalent to an in-sensor network with a conventional, memory-less sensor array such as a camera. The benefit of the transformations can be evaluated by comparison with the reference. Network accuracies are given in Figure 5C. The supplemental videos show example transformations and predictions by the networks for voltage (Videos S1 and S2), light (Videos S3 and S4), multimodal inputs (Videos S5 and S6), and the reference (Videos S7 and S8). Compared to the MNIST dataset, the video data are less linearly separable, as follows from the lower reference classification accuracy of 73.2% ± 0.1%. Markedly higher classification accuracies were found for the reservoir networks (79.2% ± 0.1%, 84.3% ± 0.1%, and 87.8% ± 0.1% for voltage, light, and multimodal sequences, respectively, all $p < 0.001$). This implies that the transformations increased the linear separability of the dataset. The improved performance of the light- compared to the voltage-input networks we find here is likely due to the longer retention time of the light inputs (see Note S1). The higher accuracy of the multimodal network likely follows from the complementary transformations by the light and voltage networks. Confusion matrices are given in Figure S21 and show that the higher accuracies of the light-input and multimodal networks are due to an overall increase in classification accuracy across all numbers. Similar to the MNIST classification results, adding experimental noise resulted in a slight decrease in accuracy (0.8%, 0.5%, and 1.3% for the voltage, light, and multimodal networks, respectively, all $p < 0.001$). Figure S22 shows that a light-input network that does not implement the 100 mV offset bias (Figure S9) reached a lower accuracy (81.0% ± 0.1%, 80.1% ± 0.1% with noise) compared to the light-input network with 100 mV offset in Figure 5C. Still, these accuracies are higher than those of the voltage-input networks, highlighting that the optical input itself is the major contributor to the higher accuracies.

The considerable increases in classification accuracy of the MNIST handwritten digits from both images and videos show that the volatile optoelectronic devices are promising for in-sensor computing applications, particularly when implementing them in multimodal networks. The fabrication of the back-contacted microscale device is fully compatible with complementary metal-oxide semiconductor (CMOS), and the architecture lends itself well to high-density integration in in-sensor computing arrays. The multimodal networks could be integrated in different 3D or planar system-on-chip architectures, where the light-input unit is directly used as the sensor and a voltage-input unit is connected to another, e.g., auditory or tactile, sensor and is kept in dark.[4]

Conclusions and outlook

In summary, we have demonstrated physical reservoir networks based on back-contacted, microscale halide perovskite devices. The networks encoded MNIST images and N-MNIST-based videos based on measured currents of the device using 4-bit voltage- and light-input sequences. We have shown with drift-diffusion simulations that the measured current transients are due to ion migration in the device. When employed in a network *in silico*, multimodal networks based on the devices that processed both light and voltage inputs gave notably higher classification accuracies compared to a linear classifier reference. The mean accuracy of the multimodal network was 95.3% ± 0.1%. We explained with simulations that the high accuracy is due to light-based transformations complementing those based on voltage due to their difference in retention time, emphasizing different features in the images. As a result, the accuracies we report here for the light-based and multimodal networks are higher than those in previous work that followed a similar approach.[15,33–35] Notable accuracy increases with respect to the reference were found for MNIST classification from video as well. The mean accuracy of the networks increased by up to 14.6% to 87.8% ± 0.1% for the multimodal mapping. Hence, the ability of the halide perovskite devices to process both voltage and light inputs allows multimodal processing in the same chip, improving accuracy. These accuracy gains, combined with the microscale device architecture that lends itself well to high-density integration and low energy consumption per operation, are promising for efficient in-sensor computing applications.

In this work, we have implemented binary inputs. The capabilities of these arrays could be extended by considering different light intensities and voltage magnitudes.[19] This extension would allow processing real-time analog signals. Analog inputs could simultaneously increase the range of accessible reservoir states, potentially resulting in even more capable nonlinear transformations.

METHODS

Materials

Silicon wafers with a 100-nm dry thermal oxide layer were purchased from Siegert Wafer. PbI_2 (99.99%) was purchased from TCI. Methylammonium iodide (MAI; purity not listed) was purchased from Solaronix. $Al(CH_3)_3$ (97%), anhydrous dimethyl formamide (DMF), DMSO, and chlorobenzene were purchased from Sigma-Aldrich. MA-N1410 resist and MA-D533/s developer were purchased from Micro Resist. All materials were used without further purification.

Halide perovskite device fabrication

Halide perovskite devices were fabricated as reported previously.[22] In short, 2.5-μm-wide gold-bottom electrodes with a thickness of 80 nm were patterned on the silicon wafer using a lift-off process with the MA-N1410 resist. A 15-nm Al_2O_3 insulating layer was deposited over the bottom electrode by atomic-layer deposition from $Al(CH_3)_3$ and H_2O precursor gases in a home-built atomic-layer deposition setup at 250°C. The lift-off process was repeated to pattern 2.5-μm-wide, 80-nm-thick gold-top contacts perpendicular to the bottom electrodes.

Inside an N_2-filled glovebox, the $MAPbI_3$ precursor was prepared by dissolving 1.1 mmol PbI_2 and MAI in 1 mL DMF and

0.1 mL DMSO. The precursor was filtered through a 0.2-μm polytetrafluorethylene filter and spin-coated on a die cut from the patterned wafer at 4,000 rpm for 30 s. After 5 s, 250 μL of chlorobenzene was added to the spinning sample. The sample was annealed at 100°C for 10 min. The sample was then encapsulated with Blufixx epoxy and a glass coverslip and cured for 1 min with a UV torch.

4-bit input measurements

The 4-bit input sequence measurements were performed with a Keysight B2902A Precision Source/Measure Unit (SMU). One channel of the SMU was used to apply voltage pulses to the device and measure the output current. For the light inputs, a second channel was used to drive a 520 nm high-power Cree XLamp XP-E LED. The irradiance was 2.8 mW/cm^2, measured with a Thorlabs PM100D optical power meter with an S120VC sensor. Measurements of each input sequence were repeated 100 times. The I_1, I_2, I_3, and I_4 currents in the main text were determined by taking the mean and standard deviation over all measurements. The currents were measured with a dynamic range of ±100 nA, an accuracy of 0.06% ± 100 pA, and a peak-to-peak noise of $\leq$2 pA.

Drift-diffusion simulations

The drift-diffusion simulations were carried out using Setfos by Fluxim with the device parameters listed in Table S4. We simulated the current due to mobile ions after applying a voltage pulse train of 1–4 voltage pulses of -1 V.

PCA of light- and voltage-input state vectors

For the PCA, voltage and light state vectors were constructed from the 4-bit input measurements. Each vector was 16-dimensional and contained an I_4 current of each of the possible 4-bit inputs. The 100 voltage- and 100 light-input state vectors were normalized separately to account for differences in I_4 current magnitudes. The vectors were then concatenated and standardized, after which PCA was performed.

Transformations of the MNIST and N-MNIST datasets

Lookup tables were constructed from the experimentally measured mean currents of each 4-bit input. Means were taken over 100 measurements for each input. The 28 × 28-pixel MNIST images were binarized with a threshold of 0.5. The binarized images were then divided into square 2 × 2-pixel patches, 4-pixel rows, or 4-pixel columns. The pixels of each patch were converted to a 4-bit sequence, as shown in Figure 4A. The sequences were then mapped by matching them to a 4-bit input sequence in the lookup table. The corresponding currents were added to a new array representing the transformed image. To account for the experimental noise, a random number was taken from a normal distribution with the mean and standard deviation determined from the 100 measurements of the corresponding I_4 current instead of mapping to the I_4 current mean. The mean and standard deviation of each sequence are displayed in Figure 2C (for voltage inputs) and Figure 3C (for light inputs). For the references, a Gaussian blur was applied with a variance determined from the original MNIST dataset (see Figure S23A). This is the noise that would be expected if the binarized MNIST images were projected on a conventional camera. Figure S23B shows that the blurring (kernel size 5, σ = 0.522) approximates the observed noise well.

The N-MNIST dataset was imported with Tonic (version 1.6.0). The original spiking, event-based dataset was filtered, binned, and thresholded to reconstruct videos of the original moving MNIST images. This modified dataset is more relevant for real-time video recognition using the in-sensor networks for simultaneous detection and processing. A de-noise filter with a 10-ms filter time was applied to the dataset, and the samples were binned into 50-ms frames. Next, the pixel values in the first four 34 × 34-pixel frames of each sample were normalized, and the frames were binarized with a threshold of 0.2. This threshold yielded the most recognizable digits in the frames. A similar approach was followed to map the frames to the voltage- and light-input data as for the MNIST mapping. However, instead of squares, rows, or columns, each individual pixel was mapped based on its four consecutive values in the four frames. After each pixel was mapped, the four frames were added to the dataset separately. This extended the training and test datasets from 60,000 to 240,000 and from 10,000 to 40,000 samples, respectively. Noise was introduced as Gaussian blurring with the same parameters as for the MNIST dataset.

Network training

Readout layers were trained on the input vectors with PyTorch (version 2.6.0), using the Adam optimizer. Hyperparameters (learning rate, weight decay, and β_1 and β_2 of the Adam optimizer) were tuned over 100 runs with Optuna (version 4.2.1), using the Tree-structured Parzen Estimator algorithm. A large, fixed batch size of 256 was chosen for faster training.[39] The MNIST dataset was randomly split into a *training* set containing 50,000 samples and a *validation* set of 10,000 samples during hyperparameter tuning. Readout layers with 1,960 (14 × 14 × 10, Square, or 7 × 28 × 10, Row and Column mapping), 3,920 (7 × 28 × 10 × 2, combined Row and Column mapping, or 14 × 14 × 10 × 2, multimodal Square mapping), or 7,840 (7 × 28 × 10 × 2 × 2, multimodal combined Row and Column mapping) weights were trained on the mapped images. The N-MNIST dataset was randomly split into a 200,000-sample *train* and a 40,000-sample *validation* set. The readout layers, consisting of 11,560 (34 × 34 × 10) or 23,120 (34 × 34 × 10 × 2, multimodal mapping) weights, were trained on the transformed frames.

After hyperparameter tuning, training was repeated on the full training dataset with the optimal hyperparameters. Mean network accuracies and standard deviations were recorded from a seed sweep over the same 10 seeds for all networks. All mean accuracies and their standard deviations were rounded to the first decimal place. Standard deviations that would be rounded to 0.0% (e.g., 0.03%) were rounded up to 0.1% instead to account for experimental error. All standard deviations were $\leq$0.10%.

Paired *t* tests were performed to check the significance of differences in accuracy. All mean accuracies within each plot were significantly different ($p < 0.001$), with the exception of the accuracies of the multimodal Square mapping network and the reference in Figure 4C (p = 0.681) and the reference networks with and without noise for the Row mapping in Figure S16D (p = 0.275) and the video dataset in Figure 5C (p = 0.546).

Convolutional neural network training

The convolutional neural network was implemented as a simple, single-layer network with a 4-pixel kernel size to allow a fair comparison with the reservoir networks. The convolutional layer had 16 output channels and a stride of 1 pixel. The hyperparameter search and seed sweep were executed the same way as for the reservoir networks.

RESOURCE AVAILABILITY

Lead contact

Requests for further information and resources should be directed to and will be fulfilled by the lead contact, Bruno Ehrler (b.ehrler@amolf.nl).

Materials availability

This study did not generate new unique reagents.

Data and code availability

Data for this article, including I-V sweeps, pulsed measurements, SEM images, and Python scripts for analysis and network training, are available at the Zenodo Repository: https://doi.org/10.5281/zenodo.17878231.

ACKNOWLEDGMENTS

The work of J.J.d.B., A.O.A., M.C.S., and B.E. received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 947221. The work is part of the Dutch Research Council NWO and was performed at the research institute AMOLF. The authors thank Marc Duursma, Bob Drent, Igor Hoogsteder, Arthur Karsten, and Laura Juškėnaitė for technical support.

AUTHOR CONTRIBUTIONS

J.J.d.B. conceived the work; carried out the experimental work, data analysis, and training of the networks; and wrote the manuscript. A.O.A. conceived the work, helped with the interpretation of the results, and commented on the manuscript. M.C.S. performed the drift-diffusion simulations, helped with the interpretation of the results, and commented on the manuscript. B.E. conceived and supervised the work, interpreted the results, and wrote the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.device.2025.101000.

Received: August 27, 2025
Revised: October 9, 2025
Accepted: November 17, 2025

REFERENCES

1. Mehonic, A., and Kenyon, A.J. (2022). Brain-inspired computing needs a master plan. Nature *604*, 255–260. https://doi.org/10.1038/s41586-021-04362-w.
2. Merolla, P.A., Arthur, J.V., Alvarez-icaza, R., Cassidy, A.S., Sawada, J., Akopyan, F., Jackson, B.L., Imam, N., Guo, C., Nakamura, Y., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. Science *345*, 668–673. https://doi.org/10.1126/science.1254642.
3. Zidan, M.A., Strachan, J.P., and Lu, W.D. (2018). The future of electronics based on memristive systems. Nat. Electron. *1*, 22–29. https://doi.org/10.1038/s41928-017-0006-8.
4. Zhou, F., and Chai, Y. (2020). Near-sensor and in-sensor computing. Nat. Electron. *3*, 664–671. https://doi.org/10.1038/s41928-020-00501-9.
5. Cover, T.M. (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. IEEE Trans. Electron. Comput. *14*, 326–334. https://doi.org/10.1109/PGEC.1965.264137.
6. Jaeger, H., and Haas, H. (2004). Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science *304*, 78–80. https://doi.org/10.1126/science.1091277.
7. Appeltant, L., Soriano, M.C., Van Der Sande, G., Danckaert, J., Massar, S., Dambre, J., Schrauwen, B., Mirasso, C.R., and Fischer, I. (2011). Information processing using a single dynamical node as complex system. Nat. Commun. *2*, 468. https://doi.org/10.1038/ncomms1476.
8. Davies, M., Srinivasa, N., Lin, T.H., Chinya, G., Cao, Y., Choday, S.H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al. (2018). Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. IEEE Micro *38*, 82–99. https://doi.org/10.1109/MM.2018.112130359.
9. Midya, R., Wang, Z., Asapu, S., Zhang, X., Rao, M., Song, W., Zhuo, Y., Upadhyay, N., Xia, Q., and Yang, J.J. (2019). Reservoir Computing Using Diffusive Memristors. Advanced Intelligent Systems *1*, 1900084. https://doi.org/10.1002/aisy.201900084.
10. Du, C., Cai, F., Zidan, M.A., Ma, W., Lee, S.H., and Lu, W.D. (2017). Reservoir computing using dynamic memristors for temporal information processing. Nat. Commun. *8*, 2204. https://doi.org/10.1038/s41467-017-02337-y.
11. Vandoorne, K., Dierckx, W., Schrauwen, B., Verstraeten, D., Baets, R., Bienstman, P., and Van Campenhout, J. (2008). Toward optical signal processing using Photonic Reservoir Computing. Opt. Express *16*, 11182–11192. https://doi.org/10.1364/OE.16.011182.
12. Vandoorne, K., Mechet, P., Van Vaerenbergh, T., Fiers, M., Morthier, G., Verstraeten, D., Schrauwen, B., Dambre, J., and Bienstman, P. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. Nat. Commun. *5*, 3541. https://doi.org/10.1038/ncomms4541.
13. Torrejon, J., Riou, M., Araujo, F.A., Tsunegi, S., Khalsa, G., Querlioz, D., Bortolotti, P., Cros, V., Yakushiji, K., Fukushima, A., et al. (2017). Neuromorphic computing with nanoscale spintronic oscillators. Nature *547*, 428–431. https://doi.org/10.1038/nature23011.
14. Lin, N., Chen, J., Zhao, R., He, Y., Wong, K., Qiu, Q., Wang, Z., and Yang, J.J. (2024). In-memory and in-sensor reservoir computing with memristive devices. APL Mach. Learn. *2*, 010901. https://doi.org/10.1063/5.0174863.
15. Liu, K., Zhang, T., Dang, B., Bao, L., Xu, L., Cheng, C., Yang, Z., Huang, R., and Yang, Y. (2022). An optoelectronic synapse based on α-In2Se3 with controllable temporal dynamics for multimode and multiscale reservoir computing. Nat. Electron. *5*, 761–773. https://doi.org/10.1038/s41928-022-00847-2.
16. Tress, W., Marinova, N., Moehl, T., Zakeeruddin, S.M., Nazeeruddin, M.K., and Grätzel, M. (2015). Understanding the rate-dependent J–V hysteresis, slow time component, and aging in CH3 NH3 PbI3 perovskite solar cells: the role of a compensated electric field. Energy Environ. Sci. *8*, 995–1004. https://doi.org/10.1039/C4EE03664F.
17. Chen, L.-W., Wang, W.-C., Ko, S.-H., Chen, C.-Y., Hsu, C.-T., Chiao, F.-C., Chen, T.-W., Wu, K.-C., and Lin, H.-W. (2021). Highly Uniform All-Vacuum-Deposited Inorganic Perovskite Artificial Synapses for Reservoir Computing. Advanced Intelligent Systems *3*, 2000196. https://doi.org/10.1002/aisy.202000196.

18. Luo, Z., Wang, W., Wu, J., Ma, G., Hou, Y., Yang, C., Wang, X., Zheng, F., Zhao, Z., Zhao, Z., et al. (2025). Leveraging Dual Resistive Switching in Quasi-2D Perovskite Memristors for Integrated Non-volatile Memory, Synaptic Emulation, and Reservoir Computing. ACS Appl. Mater. Interfaces *17*, 19879–19891. https://doi.org/10.1021/acsami.4c21159.
19. Sharma, D., Luqman, A., Ng, S.E., Yantara, N., Xing, X., Tay, Y.B., Basu, A., Chattopadhyay, A., and Mathews, N. (2024). Halide perovskite photovoltaics for in-sensor reservoir computing. Nano Energy *129*, 109949. https://doi.org/10.1016/j.nanoen.2024.109949.
20. John, R.A., Demirağ, Y., Shynkarenko, Y., Berezovska, Y., Ohannessian, N., Payvand, M., Zeng, P., Bodnarchuk, M.I., Krumeich, F., Kara, G., et al. (2022). Reconfigurable halide perovskite nanocrystal memristors for neuromorphic computing. Nat. Commun. *13*, 2074. https://doi.org/10.1038/s41467-022-29727-1.
21. Lin, C.H., Cheng, B., Li, T.Y., Retamal, J.R.D., Wei, T.C., Fu, H.C., Fang, X., and He, J.H. (2019). Orthogonal Lithography for Halide Perovskite Optoelectronic Nanodevices. ACS Nano *13*, 1168–1176. https://doi.org/10.1021/acsnano.8b05859.
22. de Boer, J.J., Alvarez, A.O., Schmidt, M.C., and Sitaridis, D. (2025). Microscale optoelectronic synapses with switchable photocurrent from halide perovskite. Preprint at arXiv. https://doi.org/10.48550/arXiv.2508.18869.
23. De Boer, J.J., and Ehrler, B. (2024). Scalable Microscale Artificial Synapses of Lead Halide Perovskite with Femtojoule Energy Consumption. ACS Energy Lett. *9*, 5787–5794. https://doi.org/10.1021/acsenergylett.4c02360.
24. Schmidt, M.C., Alvarez, A.O., De Boer, J.J., Van De Ven, L.J.M., and Ehrler, B. (2024). Consistent Interpretation of Time- and Frequency-Domain Traces of Ion Migration in Perovskite Semiconductors. ACS Energy Lett. *9*, 5850–5858. https://doi.org/10.1021/acsenergylett.4c02446.
25. Bard, A.J., and Faulkner, L.R. (2001). Introduction and Overview of Electrode Processes. In Electrochemical methods: fundamentals and applications, D. Harris, ed. (Wiley), pp. 1–43.
26. Diethelm, M., Lukas, T., Smith, J., Dasgupta, A., Caprioglio, P., Futscher, M., Hany, R., and Snaith, H.J. (2025). Probing ionic conductivity and electric field screening in perovskite solar cells: a novel exploration through ion drift currents. Energy Environ. Sci. *18*, 1385–1397. https://doi.org/10.1039/D4EE02494J.
27. Alvarez, A.O., Lédée, F., García-Batlle, M., López-Varo, P., Gros-Daillon, E., Guillén, J.M., Verilhac, J.-M., Lemercier, T., Zaccaro, J., Marsal, L.F., et al. (2023). Ionic Field Screening in MAPbBr3 Crystals Revealed from Remnant Sensitivity in X-ray Detection. ACS Phys. Chem. Au *3*, 386–393. https://doi.org/10.1021/acsphyschemau.3c00002.
28. Zhao, Y.-C., Zhou, W.-K., Zhou, X., Liu, K.-H., Yu, D.-P., and Zhao, Q. (2017). Quantification of light-enhanced ionic transport in lead iodide perovskite thin films and its solar cell applications. Light Sci. Appl. *6*, e16243. https://doi.org/10.1038/lsa.2016.243.
29. Inman, H.F., and Bradley, E.L. (1989). The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. Commun. Stat. Theor. Methods *18*, 3851–3874. https://doi.org/10.1080/03610928908830127.
30. Bertoluzzi, L., Boyd, C.C., Rolston, N., Xu, J., Prasanna, R., O’Regan, B.C., and McGehee, M.D. (2020). Mobile Ion Concentration Measurement and Open-Access Band Diagram Simulation Platform for Halide Perovskite Solar Cells. Joule *4*, 109–127. https://doi.org/10.1016/j.joule.2019.10.003.
31. Schaetti, N., Salomon, M., and Couturier, R. (2016). Echo State Networks-Based Reservoir Computing for MNIST Handwritten Digits Recognition. In 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), J. Bourgeois and F. Magoules, eds. (IEEE)), pp. 484–491. https://doi.org/10.1109/CSE-EUC-DCABES.2016.229.
32. Lin, N., Wang, S., Li, Y., Wang, B., Shi, S., He, Y., Zhang, W., Yu, Y., Zhang, Y., Zhang, X., et al. (2025). Resistive memory-based zero-shot liquid state machine for multimodal event data learning. Nat. Comput. Sci. *5*, 37–47. https://doi.org/10.1038/s43588-024-00751-z.
33. Jang, Y.H., Kim, W., Kim, J., Woo, K.S., Lee, H.J., Jeon, J.W., Shim, S.K., Han, J., and Hwang, C.S. (2021). Time-varying data processing with nonvolatile memristor-based temporal kernel. Nat. Commun. *12*, 5727. https://doi.org/10.1038/s41467-021-25925-5.
34. Koh, S.-G., Shima, H., Naitoh, Y., Akinaga, H., and Kinoshita, K. (2022). Reservoir computing with dielectric relaxation at an electrode–ionic liquid interface. Sci. Rep. *12*, 6958. https://doi.org/10.1038/s41598-022-10152-9.
35. Yamazaki, Y., and Kinoshita, K. (2024). Photonic Physical Reservoir Computing with Tunable Relaxation Time Constant. Adv. Sci. *11*, 2304804. https://doi.org/10.1002/advs.202304804.
36. Ross, A., Leroux, N., De Riz, A., Marković, D., Sanz-Hernández, D., Trastoy, J., Bortolotti, P., Querlioz, D., Martins, L., Benetti, L., et al. (2023). Multilayer spintronic neural networks with radiofrequency connections. Nat. Nanotechnol. *18*, 1273–1280. https://doi.org/10.1038/s41565-023-01452-w.
37. Yao, P., Wu, H., Gao, B., Tang, J., Zhang, Q., Zhang, W., Yang, J.J., and Qian, H. (2020). Fully hardware-implemented memristor convolutional neural network. Nature *577*, 641–646. https://doi.org/10.1038/s41586-020-1942-4.
38. Orchard, G., Jayawant, A., Cohen, G.K., and Thakor, N. (2015). Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades. Front. Neurosci. *9*, 437. https://doi.org/10.3389/fnins.2015.00437.
39. Shallue, C.J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R., and Dahl, G.E. (2019). Measuring the Effects of Data Parallelism on Neural Network Training. J. Mach. Learn. Res. *20*, 1–49.

DEVICE, Volume *4*

Supplemental information

In-sensor computing with halide perovskite-based optoelectronic reservoir networks

Jeroen J. de Boer, Agustin O. Alvarez, Moritz C. Schmidt, and Bruno Ehrler

Table of Contents

Figure S17. Confusion matrices of MNIST handwritten digit classification.

Table S3. Comparison of MNIST classification accuracies and energy consumptions of physical reservoir networks based on various platforms.

Figure S18. MNIST classification by logistic regression and a single-layer convolutional neural network (CNN).

Figure S19. MNIST classification by a light-input network without the 100 mV offset bias, implementing the Row + Column mapping.

Figure S20. Additional N-MNIST frame mappings.

Figure S21. Confusion matrices of handwritten digit classification from video.

Figure S22. Handwritten digit classification from video by a light-input network without the 100 mV offset bias.

Table S4. Simulation parameters used for the drift-diffusion simulations.

Figure S23. Fitting of a Gaussian blurring kernel to the binarized MNIST data.

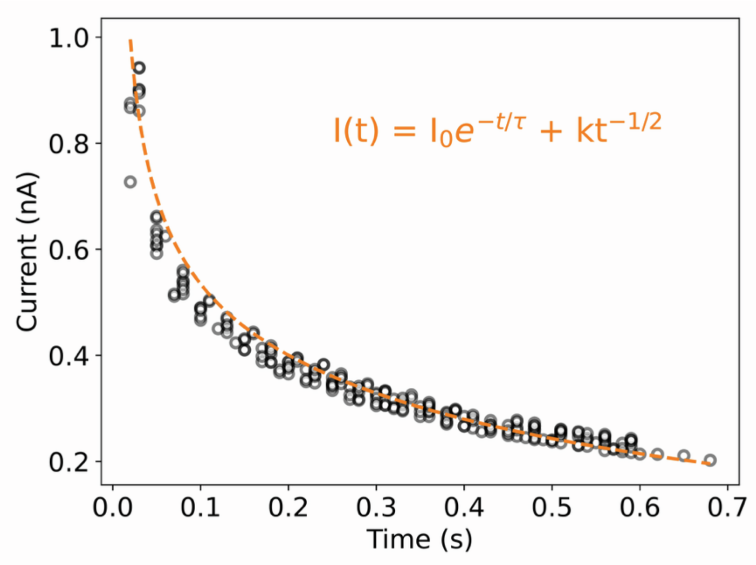


Figure S1. Fit to the combined data of ten transient current measurements as in Figure 1d. The current is measured after the final -1 V pulse, and fit with equation $I(t) = I_0 e^{-t/\tau} + kt^{-1/2}$. Fitting parameters are given in Table S1.

Table S1. Fitting parameters for the fit in Figure S1, along with the fitting error.

Fitting parameter	Value
I_0	0.22 ± 0.02 nA
τ	0.53 ± 0.04 s
k	0.110 ± 0.005 nA $\sqrt{s}$

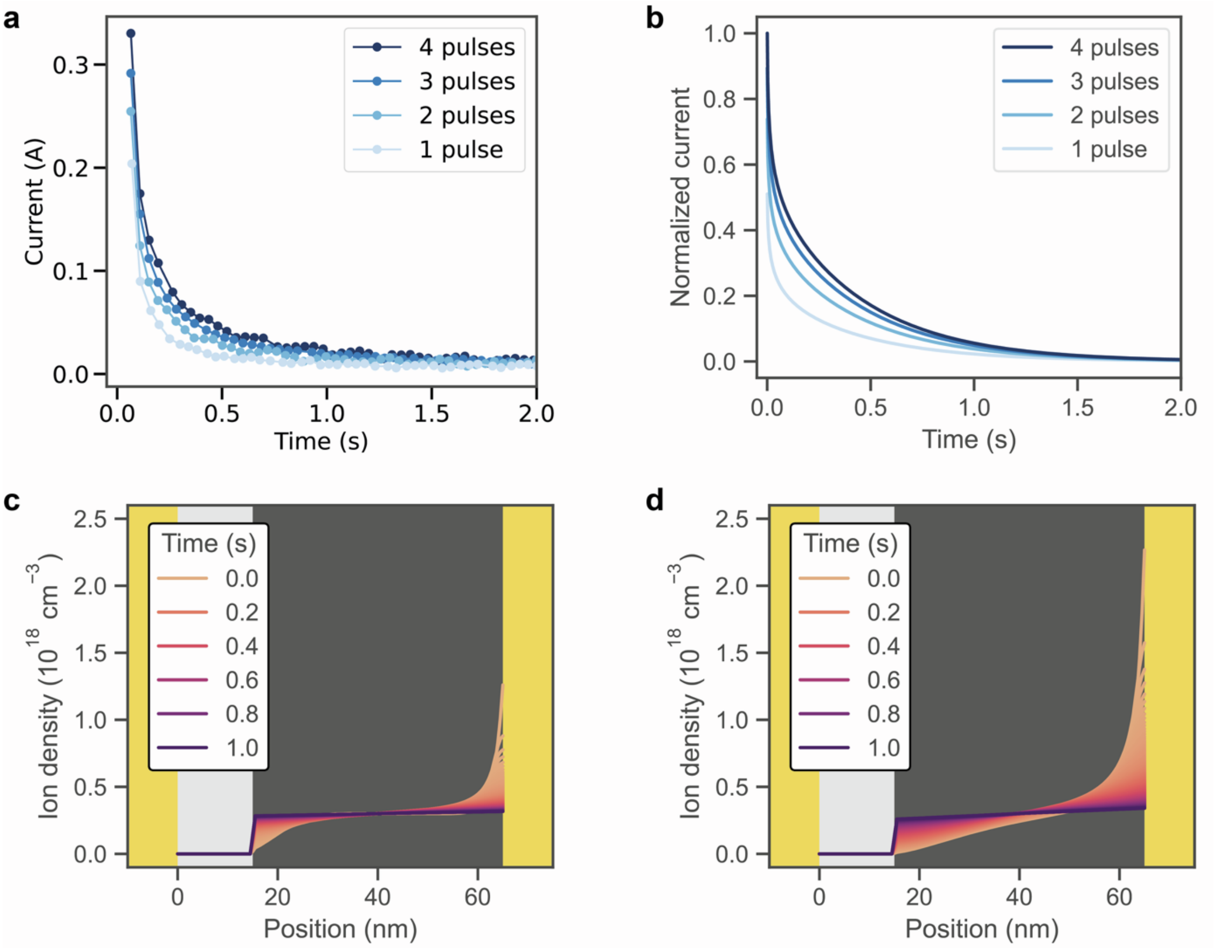


Figure S2. Comparison of measured transient currents with drift-diffusion simulations. ***(a)*** *Measured currents after applying a series of -1 V pulses. The final pulse is removed at t = 0 sec.* ***(b)*** *Drift-diffusion simulations of the measurements.* ***(c)*** *Ion densities after applying a single -1 V pulse. Ions are initially accumulated at the cathode at t = 0 sec and redistribute throughout the perovskite layer within seconds.* ***(d)*** *The same simulation as in* ***(c)****, but after applying four -1 V pulses. The same effect is observed, but with a larger magnitude.*

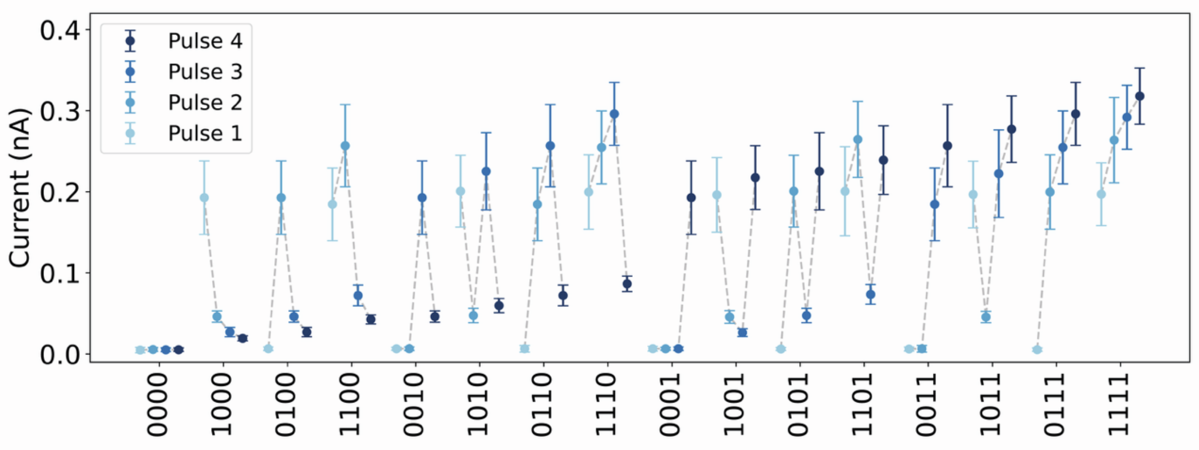


Figure S3. Mean currents after each of the four pulses of the 4-bit voltage inputs. Means taken over 100 measurements, with error bars representing one standard deviation. Gray dotted lines are added for each input to guide the eye. Inputs with similar mean final currents are more easily separable by considering intermediate currents.

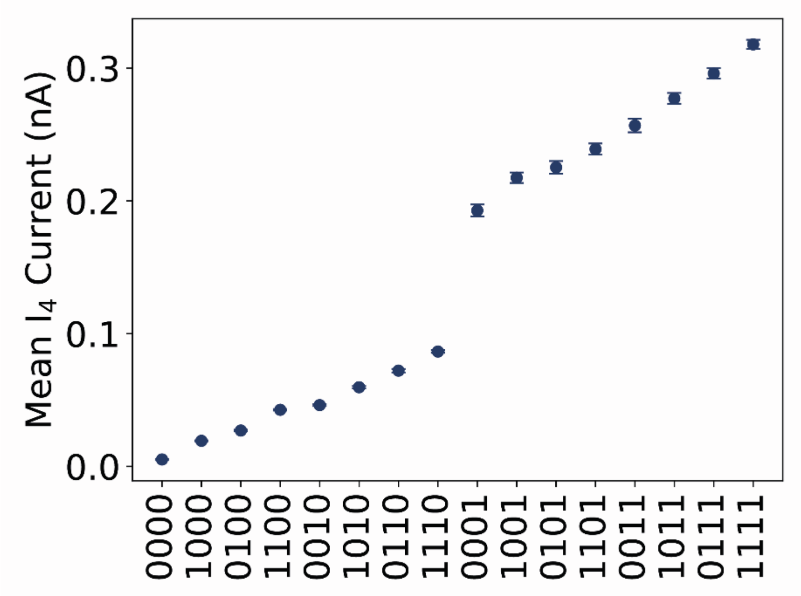


Figure S4. Mean currents with error on the mean of the 4-bit voltage inputs. The low errors on the means indicate that the obtained means are well-defined.

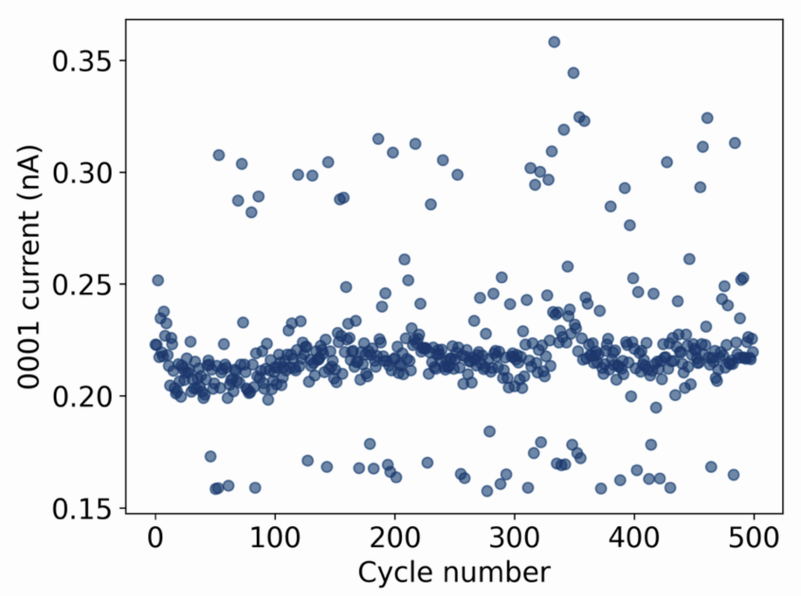


Figure S5. Cycle-to-cycle variation of 500 0001 input I_4 currents. The currents fall in a narrow distribution, with some outliers.

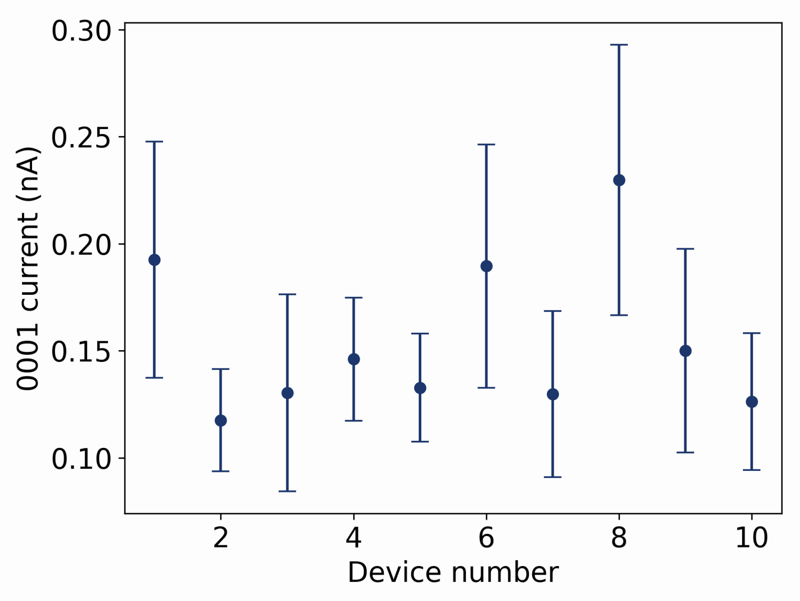


Figure S6. Device-to-device variation of 10 devices, determined by taking the mean over 10 I_4 currents of 0001 inputs for each. The means all fall roughly within one standard deviation of each other.

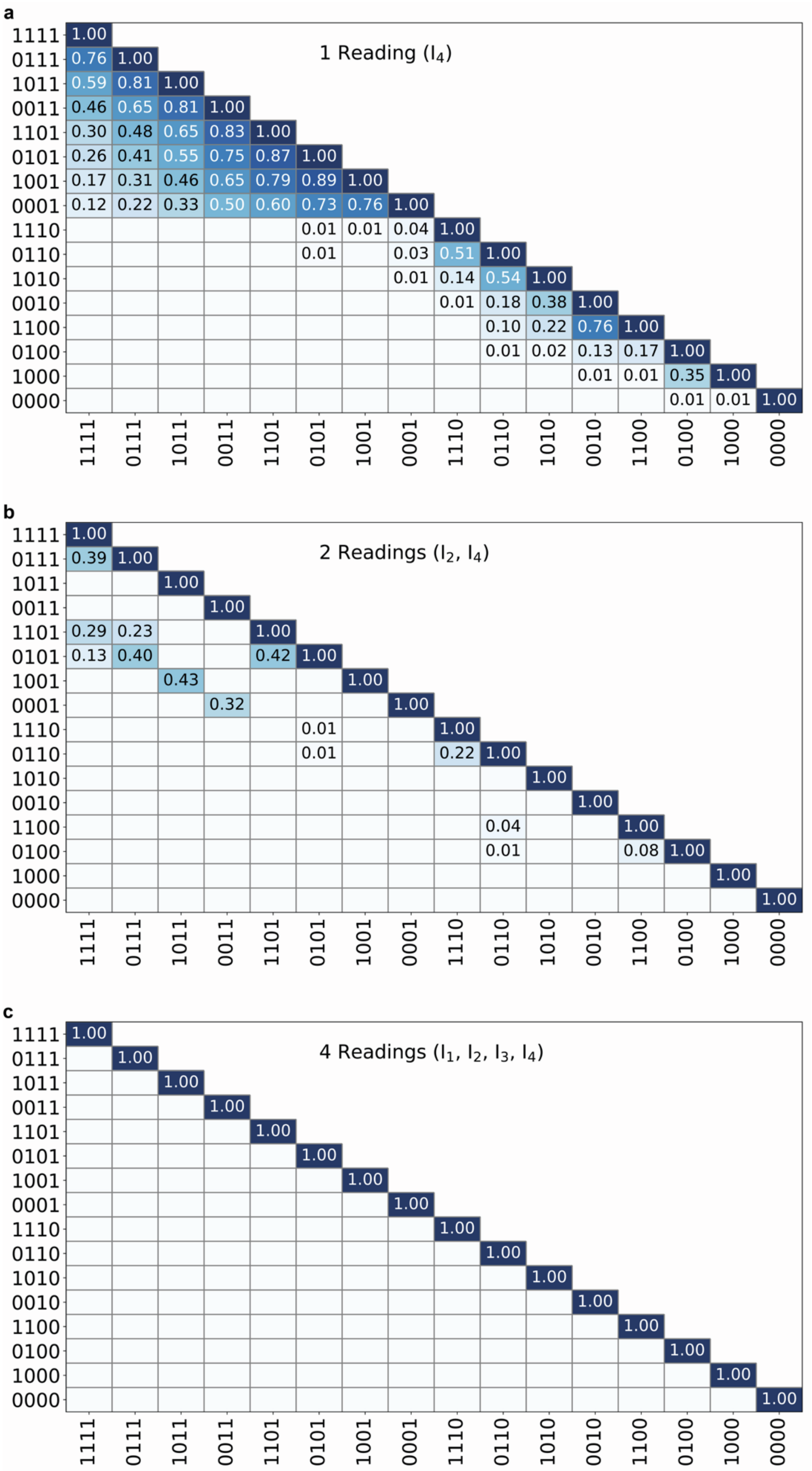


Figure S7. Overlap matrices of the four-bit voltage input mappings to ***(a)*** *the current after the fourth bit,* ***(b)*** *the current after the second and the fourth bit, and* ***(c)*** *the currents after all four bits. Overlap coefficients below 0.01 are not plotted for clarity. Significant overlap coefficients are found for similar inputs when mapping to fewer currents.*

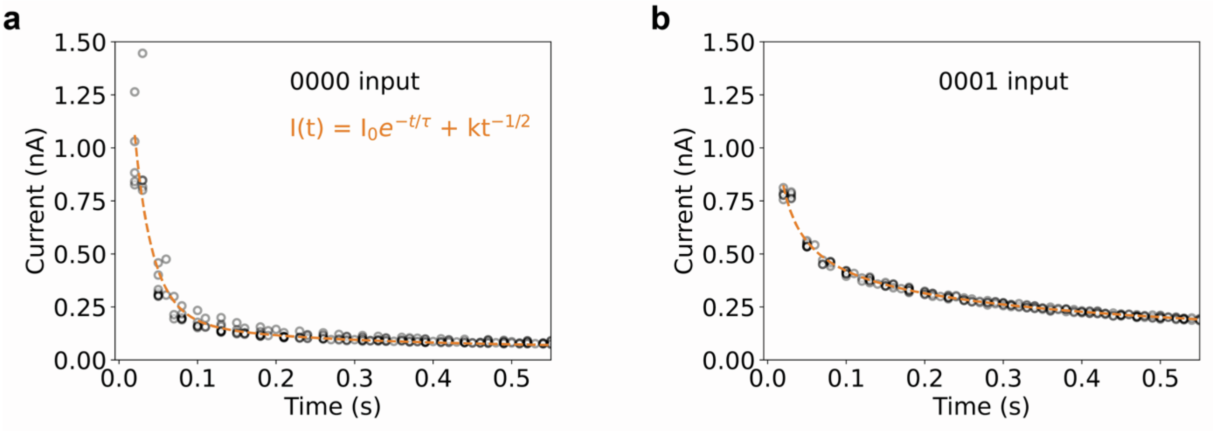


*Figure S8. Fits to the combined data of ten 0000 **(a)** and 0001 **(b)** light input sequence transient current measurements. The current is measured after the final -1 V pulse and fit with equation $I(t) = I_0 e^{-t/\tau} + kt^{-1/2}$. Fitting parameters are given in Table S2.*

Table S2. Fitting parameters of the current decays in Figure S8a and b. Errors represent the fitting error.

Input	**I_0 (nA)**	**τ (s)**	**k (nA $\sqrt{s}$)**
0000	1.64 ± 0.14	0.023 ± 0.002	0.052 ± 0.003
0001	0.12 ± 0.01	0.737 ± 0.066	0.100 ± 0.002

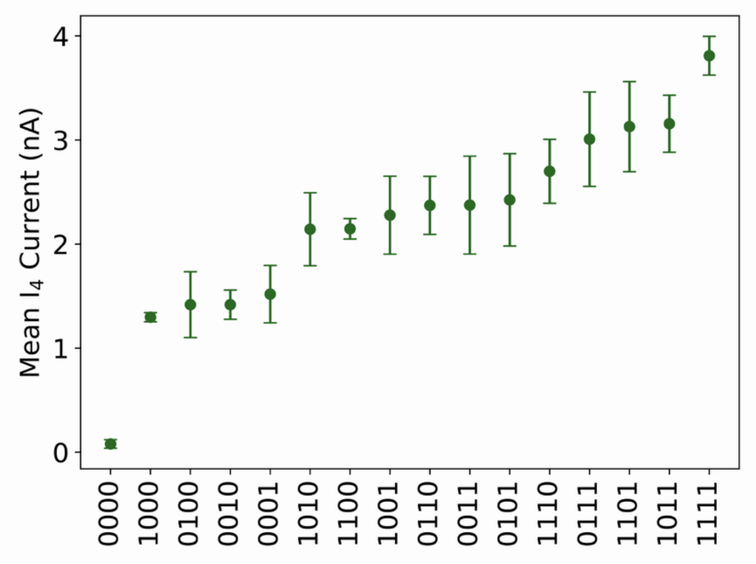


Figure S9. Mean I_4 current for each of the 4-bit inputs if no 100 mV offset bias is applied between the -1 V pulses. More overlap between I_4 currents is found compared to the measurement with the offset in Figure 3c of the main text.

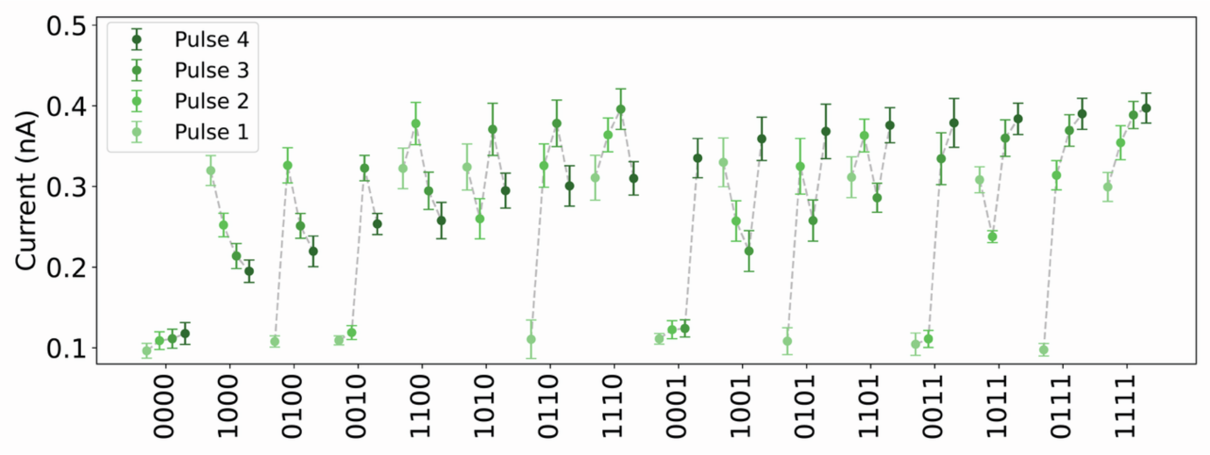


Figure S10. Mean currents after each of the four pulses of the 4-bit light inputs. Means taken over 100 measurements, with error bars representing one standard deviation. Gray dotted lines are added for each input to guide the eye. Inputs with similar mean final currents are more easily separable by considering intermediate currents.

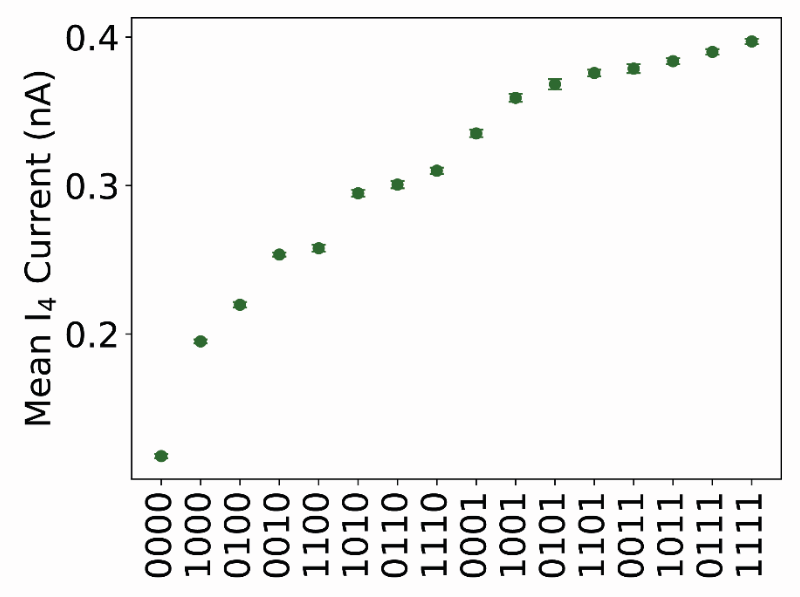


Figure S11. Mean currents with error of the mean of the 4-bit light inputs. The low errors of the means indicate that the obtained means are well-defined.

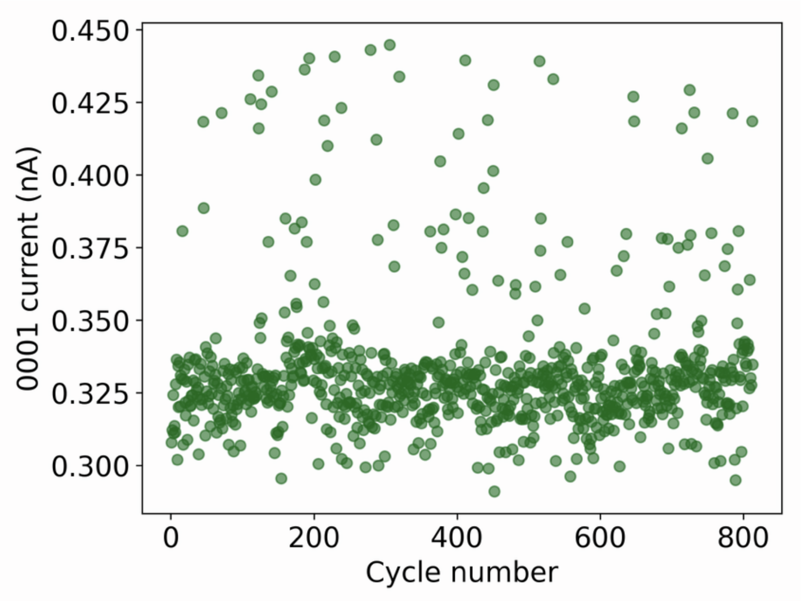
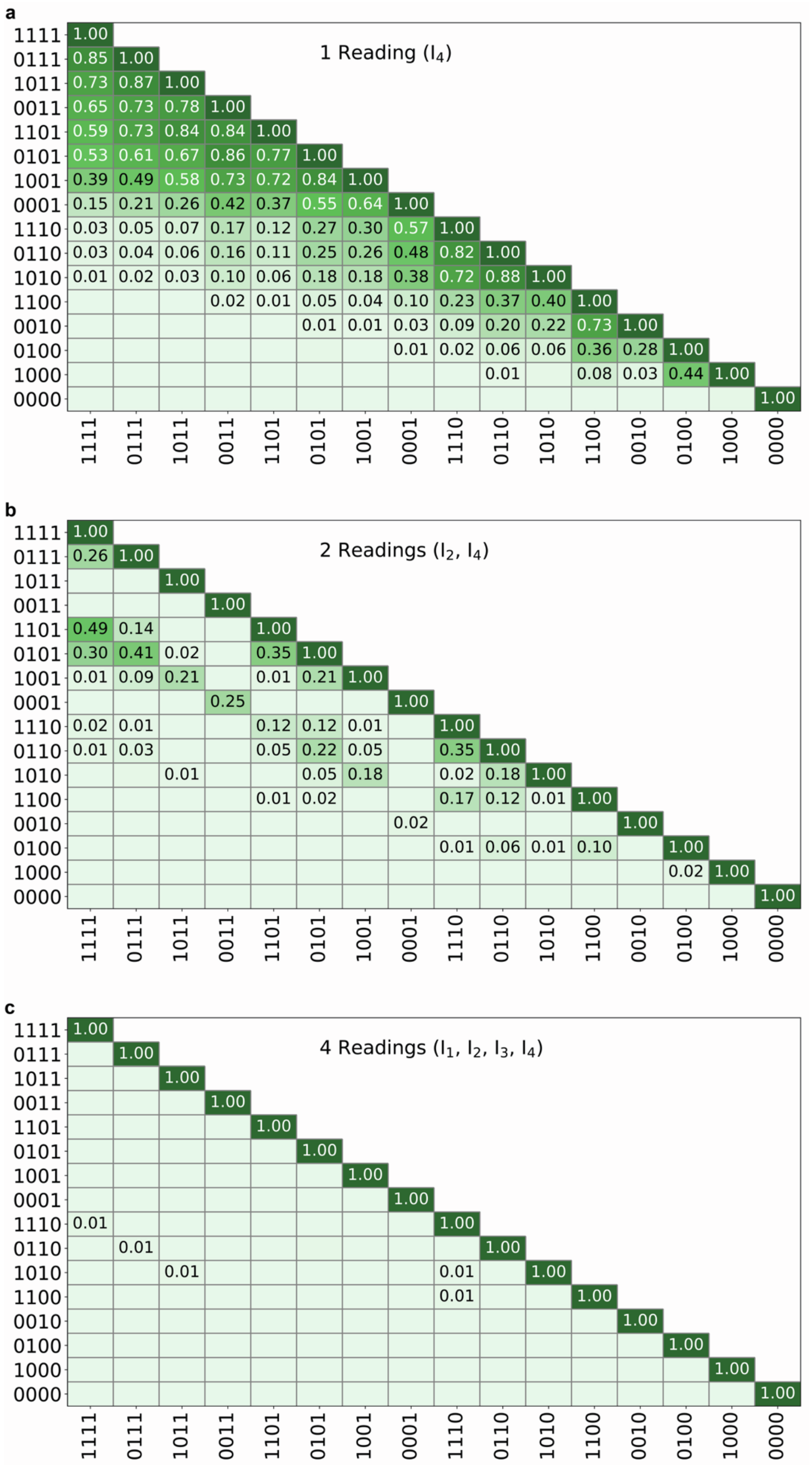


Figure S12. Cycle-to-cycle variation of 800 0001 input I_4 currents. The currents fall in a narrow distribution, with some outliers.



a

	1111	0111	1011	0011	1101	0101	1001	0001	1110	0110	1010	1100	0010	0100	1000	0000
1111	1.00															
0111	0.85	1.00														
1011	0.73	0.87	1.00													
0011	0.65	0.73	0.78	1.00												
1101	0.59	0.73	0.84	0.84	1.00											
0101	0.53	0.61	0.67	0.86	0.77	1.00										
1001	0.39	0.49	0.58	0.73	0.72	0.84	1.00									
0001	0.15	0.21	0.26	0.42	0.37	0.55	0.64	1.00								
1110	0.03	0.05	0.07	0.17	0.12	0.27	0.30	0.57	1.00							
0110	0.03	0.04	0.06	0.16	0.11	0.25	0.26	0.48	0.82	1.00						
1010	0.01	0.02	0.03	0.10	0.06	0.18	0.18	0.38	0.72	0.88	1.00					
1100				0.02	0.01	0.05	0.04	0.10	0.23	0.37	0.40	1.00				
0010						0.01	0.01	0.03	0.09	0.20	0.22	0.73	1.00			
0100								0.01	0.02	0.06	0.06	0.36	0.28	1.00		
1000										0.01		0.08	0.03	0.44	1.00	
0000																1.00

b

	1111	0111	1011	0011	1101	0101	1001	0001	1110	0110	1010	1100	0010	0100	1000	0000
1111	1.00															
0111	0.26	1.00														
1011			1.00													
0011				1.00												
1101	0.49	0.14			1.00											
0101	0.30	0.41	0.02		0.35	1.00										
1001	0.01	0.09	0.21		0.01	0.21	1.00									
0001				0.25				1.00								
1110	0.02	0.01			0.12	0.12	0.01		1.00							
0110	0.01	0.03			0.05	0.22	0.05		0.35	1.00						
1010			0.01			0.05	0.18		0.02	0.18	1.00					
1100					0.01	0.02			0.17	0.12	0.01	1.00				
0010								0.02					1.00			
0100									0.01	0.06	0.01	0.10		1.00		
1000														0.02	1.00	
0000																1.00

c

	1111	0111	1011	0011	1101	0101	1001	0001	1110	0110	1010	1100	0010	0100	1000	0000
1111	1.00															
0111		1.00														
1011			1.00													
0011				1.00												
1101					1.00											
0101						1.00										
1001							1.00									
0001								1.00								
1110	0.01								1.00							
0110		0.01								1.00						
1010			0.01						0.01		1.00					
1100									0.01			1.00				
0010													1.00			
0100														1.00		
1000															1.00	
0000																1.00

Figure S13. Overlap matrices of the four-bit light input mappings to ***(a)*** *the current after the fourth bit,* ***(b)*** *the current after the second and the fourth bit, and* ***(c)*** *the currents after all four bits. Overlap coefficients below 0.01 are not plotted for clarity. Significant overlap coefficients are found for similar inputs when mapping to fewer currents.*

Note S1. Impact of the I_4 currents on classification accuracy

This Note investigates the impact of the obtained currents for 4-bit sequences on the MNIST classification accuracy. We simulate I_4 currents and show the effect of different trends on the MNIST image transformations to explain differences in classification accuracy. The I_4 currents obtained for the light (Figure 3c) and voltage inputs (Figure 2c) are compared to the generated I_4 currents to explain the differences in performance of the networks in the main text. Finally, we discuss how plotting the I_4 currents for different 4-bit input sequences allows facile estimation of the in-sensor reservoir network performance. These results can be used to find optimal experimental parameters.

In reservoir computing, the reservoir performs a nonlinear transformation of the input,[S1] increasing the separability of otherwise convoluted features.[S2] In our implementation, the reservoir consists of devices that transform 4-bit voltage and light input sequences into a current. In this context, input sequences are separable if distinct current values are obtained for each sequence. This requires the device to have a volatile memory. Because of its memory, the device outputs a higher I_4 current for a 0011 sequence than it does for a 0001 input. The volatility, on the other hand, causes the output current to depend on the order in which light or voltage pulses are applied in the sequence. It extends the device response beyond an additive, linear response in which inputs with the same number of pulses, such as the 1001 and 0011 sequences, yield the same I_4 current. Due to the current decay in our experiments, lower I_4 currents are output if a pulse is applied earlier in the sequence. Consequently, a lower I_4 current is obtained for the 1001 sequence than for the 0011 input. The volatility is thus crucial for separating inputs.

Ideally, the I_4 currents should be as distinct as possible after the transformation. The overlap between I_4 currents is minimized if they follow a linear trend in plots such as those in Figure 2c and 3c in the main text. Simulated I_4 currents following a linear trend are given as the gray dotted line in Figure S14a. The normalized current is calculated as $I_4(s) = \frac{s}{N-1}$, where I_4 is the current of the s^th^ sequence on the x-axis and N is the number of possible sequences (16 sequences for a 4-bit input). The first sequence (0000) is assigned $s = 0$. The 4-bit voltage and light sequences in the main text produced I_4 currents that deviate from this linear trend. The voltage inputs (Figure 2c) yielded lower I_4 currents for the first half of the sequences. The light inputs (Figure 3c), on the other hand, produced I_4 currents above the linear trend over the whole range of sequences. We simulate these responses with polynomial functions.

I_4 currents falling below the linear trend are calculated as $I_4(s) = \left(\frac{s}{N-1}\right)^n$, shown in blue in Figure S14a. In the plot, n is 1.5, 2.0, 3.0, or 4.0, where higher values result in a stronger convex, parabolic shape. These trends compress I_4 currents of earlier sequences on the x-axis (the 0000 end) and stretch out later ones (on the 1111 end). In experimental measurements of the I_4 current, this is due to high volatility, causing a 0000 and 1000 input, for example, to give a similar output. Currents above the linear trend were calculated as $I_4(s) = 1 - \left(1 - \frac{s}{N-1}\right)^{\mathrm{n}}$ (red curves in Figure S14a), using the same range of n-values. These currents follow a concave parabolic curve, with a stronger curvature for higher values of n. In this case, currents for later sequences on the x-axis are compressed, and those of earlier inputs are stretched. In experimental measurements, this response is caused by long retention times, resulting in a saturation of the I$_4$ current (e.g. a similar I$_4$ current for 0111 and 1111 sequences).

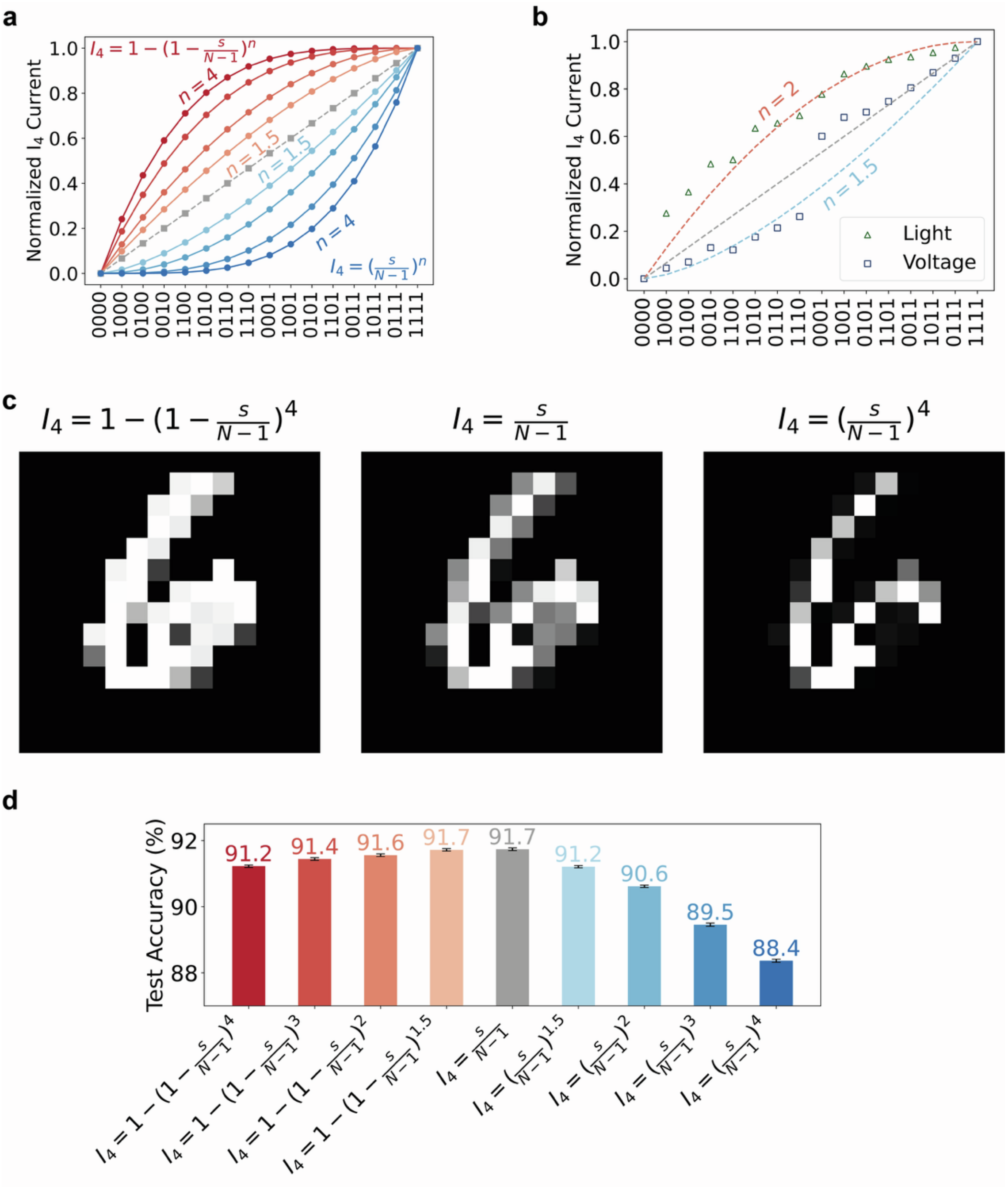


Figure S14. Transformations of the MNIST dataset based on simulated I$_4$ current following linear and polynomial trends. ***(a)*** *Input sequences with normalized I$_4$ currents that increase linearly (squares and dotted gray line), or according to polynomial functions $I_4(s) = 1 - \left(1 - \frac{s}{N-1}\right)^n$ (red circles and continuous lines) and $I_4(s) = \left(\frac{s}{N-1}\right)^n$ (blue circles and continuous lines), with n ∈ {1.5, 2, 3, 4}.* ***(b)*** *Normalized I$_4$ currents of the light (green triangles) and voltage (blue squares) inputs compared to the artificial I$_4$ currents. The light input I$_4$ currents follow a similar trend to the polynomial $I_4(s) = 1 - \left(1 - \frac{s}{N-1}\right)^2$. The I$_4$ currents of the voltage inputs up to the 1110 sequence resemble those of the function $I_4(s) = \left(\frac{s}{N-1}\right)^{1.5}$ and become more linear afterwards.* ***(c)*** *The same MNIST image as in Figure 4a mapped to the I$_4$ currents calculated based on the $I_4(s) = 1 - \left(1 - \frac{s}{N-1}\right)^4$, $I_4(s) = \frac{s}{N-1}$, and $I_4(s) = \left(\frac{s}{N-1}\right)^4$ functions. Mapping to the $I_4(s) = \frac{s}{N-1}$ currents results in a number six with a range of grayscale values. Mapping to currents generated by the polynomial*

functions yields more extreme grayscale values, with more bright pixels for $I_4(s) = 1 - \left(1 - \frac{s}{N-1}\right)^4$ and more dark pixels for $I_4(s) = \left(\frac{s}{N-1}\right)^4$. ***(d)*** *Classification accuracies of the datasets mapped to the linear and polynomial functions. The highest accuracy of 91.7% is obtained for the linear relation. Minor accuracy penalties are obtained for the concave $I_4(s) = 1 - \left(1 - \frac{s}{N-1}\right)^n$ functions, while they are more severe for the convex $I_4(s) = \left(\frac{s}{N-1}\right)^n$ functions.*

These polynomial functions were chosen because they stretch and compress the linear response to the same extent, but at opposite ends of the x-axis. The equal magnitude of stretching and compression allows a fair comparison of the two trends.

Figure S14b illustrates that the measured I_4 currents of the first half of the voltage and all light-inputs are described best by $I_4(s) = \left(\frac{s}{N-1}\right)^{1.5}$ and $I_4(s) = 1 - \left(1 - \frac{s}{N-1}\right)^2$, respectively.

The effects of the deviations from the linearly increasing I_4 currents are visualized in Figure S14c. The linearly increasing I_4 currents result in a transformation of the MNIST image to an image with a range of grayscale values. Transformations based on convex ($I_4(s) = \left(\frac{s}{N-1}\right)^4$) or concave ($I_4(s) = 1 - \left(1 - \frac{s}{N-1}\right)^4$) I_4 currents result in more extreme grayscale values. Convex I_4 currents force intermediate values in the image transformed based on linear I_4 currents closer to 0, resulting in a lossy transformation of the image. The lower right part of the loop of the number six, for example, is no longer visible in the image. Conversely, the image transformed based on concave I_4 currents force otherwise intermediate values closer to 1. This results in saturation in the transformed image, for example in the upper right part of the loop of the number six. At the same time, features that were more difficult to notice, around the curved line of the number for example, are more distinguishable for this transformed image. Both the lossy and saturated transformations are expected to decrease later classification accuracies, as different features become more difficult to distinguish.

Figure S14d shows the classification accuracies obtained by linear readout layers trained on the transformed datasets. The linearly increasing I_4 currents give the highest accuracy (91.7 ± 0.1%). The mean accuracies for concave I_4 current transformations decreased only slightly. For n = 1.5, no statistically significant difference was found compared to the linear I_4 transformation ($p = 0.252$). For higher values of n, the differences were statistically significant ($p < 0.001$), but modest, with an accuracy of 91.2 ± 0.1% for n = 4. Interestingly, the accuracy penalty was markedly more severe for the convex I_4 currents ($p < 0.001$ for all differences). The same mean accuracy of 91.2 ± 0.1% for the n = 4 concave I_4 currents was reached for n = 1.5 already. The accuracy then decreased further for larger values of n, down to 88.4 ± 0.1% for n = 4. The accuracies for the convex n = 1.5 (91.2 ± 0.1%) and concave n = 2 (91.6 ± 0.1%) I_4 transformations are well-matched to those obtained for the voltage (91.1 ± 0.1%) and light (91.6 ± 0.1%) inputs in Figure 4c. This indicates that the polynomial functions are good approximations for the experimentally measured I_4 currents.

These results suggest that, for MNIST classification, the transformation should ideally give equal separability for all inputs (linear trend), or slightly emphasize the earlier sequences in Figure S14 where fewer pulses are applied (mildly concave trends). These inputs correspond to patches with few white pixels, which typically constitute the edges of the digits. Our results, therefore, suggest that these edges should remain distinguishable from the background for optimal classification. Saturation of inputs with more applied pulses, due to transformations with concave I_4 currents, impacts the classification accuracy less severely. Contrast between patches with many white pixels, therefore, seems less important for classification.

Our findings explain the better performance of the light-input networks in Figure 4c and 5c. Moreover, the simulations are generalizable for MNIST classification with any 4-bit input in-sensor reservoir network. This is particularly valuable when optimizing experimental parameters. I_4 currents, or equivalent outputs, can be measured for the different sequences and plotted as in Figure S14a. Figure S14a and d can then function as a simple lookup table to estimate classification accuracy. This removes the need for time-consuming hyperparameter tuning and training of readout layers. Timesteps of the input can be increased (for a more convex I_4 trend) or decreased (for a more concave I_4 trend) to optimize the accuracy of the network. Our results show that, at least for MNIST classification based on 4-bit inputs, experimental parameters should be adjusted to obtain linear or slightly concave I_4 current trends for best results. We note that the timesteps could not be decreased further for our voltage inputs to obtain more concave trends due to limitations of our experimental setup.

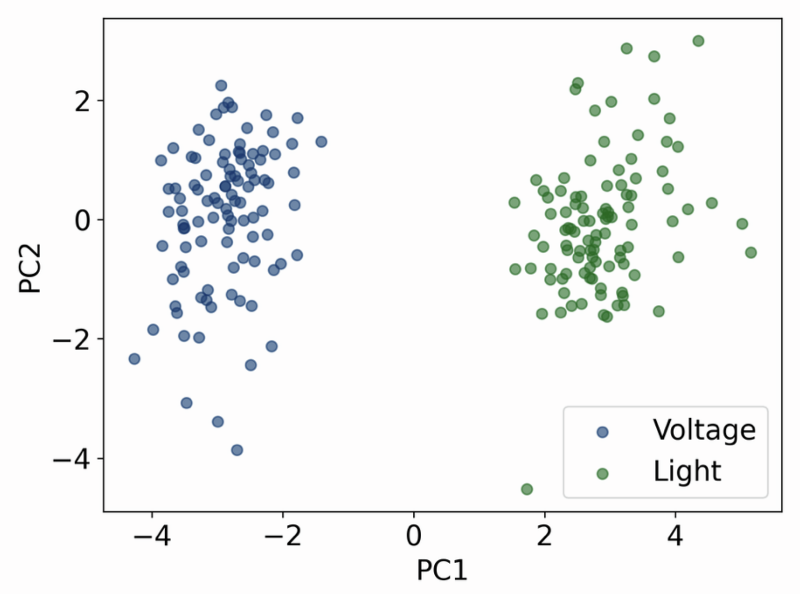


Figure S15. Principal component analysis of the voltage-input and light-input state vectors. The two input modes produce distinct clusters that are separated by the first principal component (PC1).

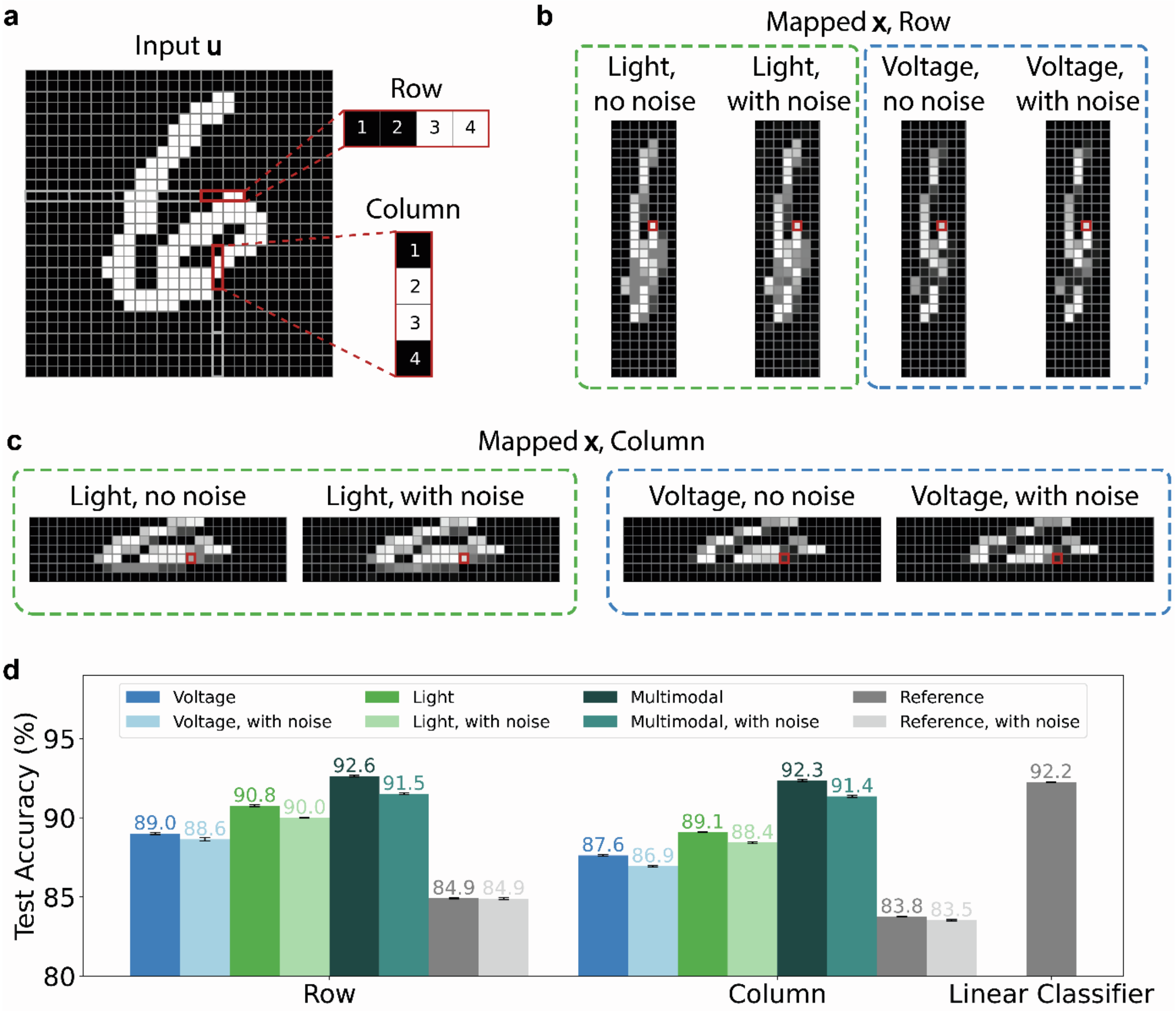
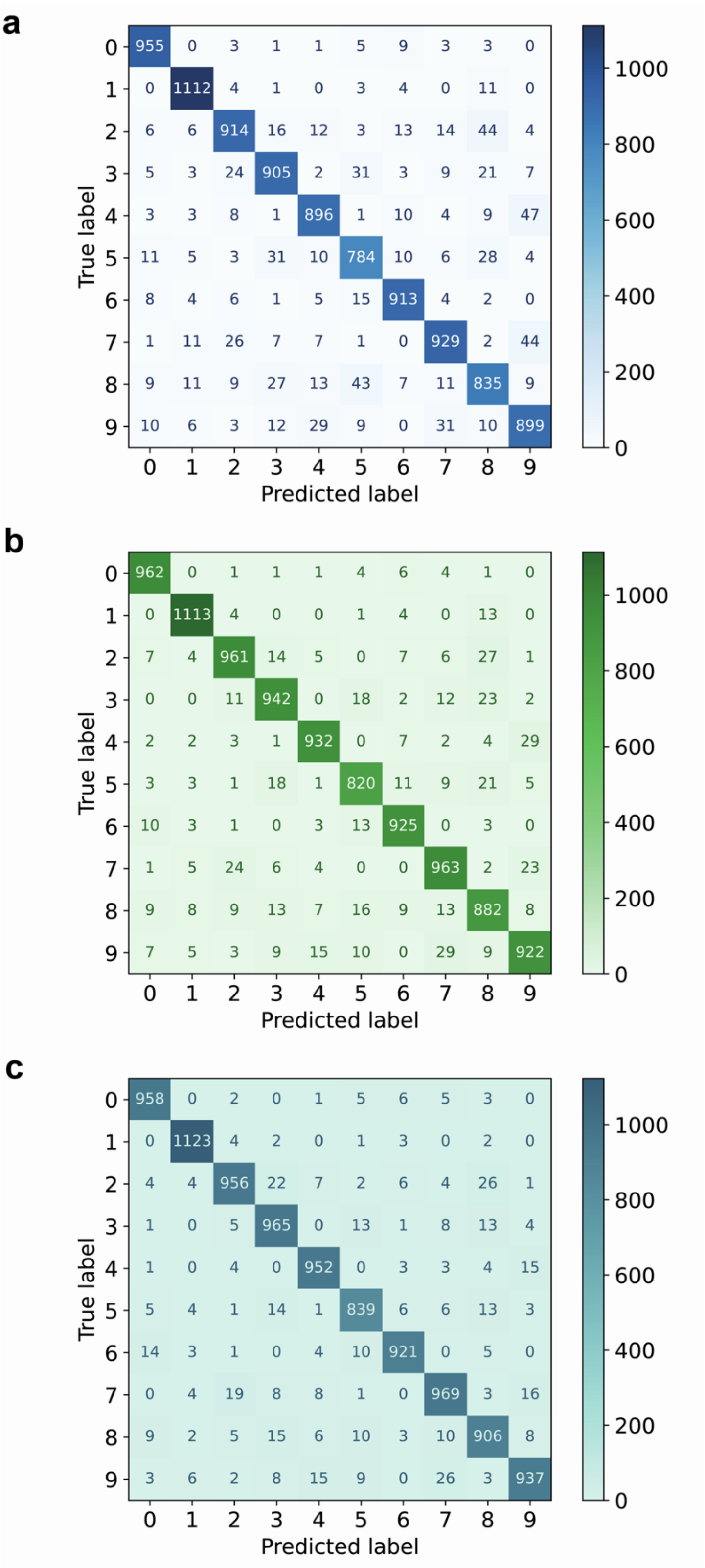


Figure S16. Row and Column *mapping for MNIST transformation and classification.* ***(a)*** *The same sample number six as in Figure 4a divided into four-pixel rows or columns. Rows or columns are mapped to the I_4 currents corresponding to their 4-bit sequences.* ***(b)*** *The number six transformed by* Row*-mapping. The image retains the same number of pixels vertically (28), but becomes fourfold smaller horizontally (to 7 pixels wide) by the transformation. The pixel outlined in red corresponds to the row outlined in* ***(a)****.* ***(c)*** *The number six transformed by* Column*-mapping. The image retains the same number of pixels horizontally (28), but becomes fourfold smaller vertically (to 7 pixels high) by the transformation. The pixel outlined in red corresponds to the column outlined in* ***(a)****.* ***(d)*** *Mean classification accuracies of the voltage, light, and multimodal networks for datasets transformed by* Row *and* Column *mapping. The accuracies are compared to references, for which each row or column was mapped to the binary value of pixel 4. For the voltage, light, and multimodal networks, noise was included by taking random samples from a normal distribution of the I_4 currents for each sequence. For the reference network, a Gaussian blur with a kernel based on the original MNIST dataset (see Methods for details) was applied to obtain a noisy dataset. The accuracies are compared to a linear classifier trained on the binarized MNIST dataset without any transformations. Each test accuracy was determined from 10 independent runs with different random seeds. Error bars indicate one standard deviation.*



a

	0	1	2	3	4	5	6	7	8	9
0	955	0	3	1	1	5	9	3	3	0
1	0	1112	4	1	0	3	4	0	11	0
2	6	6	914	16	12	3	13	14	44	4
3	5	3	24	905	2	31	3	9	21	7
4	3	3	8	1	896	1	10	4	9	47
5	11	5	3	31	10	784	10	6	28	4
6	8	4	6	1	5	15	913	4	2	0
7	1	11	26	7	7	1	0	929	2	44
8	9	11	9	27	13	43	7	11	835	9
9	10	6	3	12	29	9	0	31	10	899

b

	0	1	2	3	4	5	6	7	8	9
0	962	0	1	1	1	4	6	4	1	0
1	0	1113	4	0	0	1	4	0	13	0
2	7	4	961	14	5	0	7	6	27	1
3	0	0	11	942	0	18	2	12	23	2
4	2	2	3	1	932	0	7	2	4	29
5	3	3	1	18	1	820	11	9	21	5
6	10	3	1	0	3	13	925	0	3	0
7	1	5	24	6	4	0	0	963	2	23
8	9	8	9	13	7	16	9	13	882	8
9	7	5	3	9	15	10	0	29	9	922

c

	0	1	2	3	4	5	6	7	8	9
0	958	0	2	0	1	5	6	5	3	0
1	0	1123	4	2	0	1	3	0	2	0
2	4	4	956	22	7	2	6	4	26	1
3	1	0	5	965	0	13	1	8	13	4
4	1	0	4	0	952	0	3	3	4	15
5	5	4	1	14	1	839	6	6	13	3
6	14	3	1	0	4	10	921	0	5	0
7	0	4	19	8	8	1	0	969	3	16
8	9	2	5	15	6	10	3	10	906	8
9	3	6	2	8	15	9	0	26	3	937

*Figure S17. Confusion matrices of MNIST handwritten digit classification by **(a)** the voltage-input, **(b)** the light-input, and **(c)** the multimodal network.*

Table S3. Comparison of MNIST classification accuracies and energy consumptions of physical reservoir networks based on various platforms. Only single-layer networks are considered. Energy consumptions are given as the minimum of processing a single input by a single device. If the energy consumption was not reported, it was estimated based on the experimental results.

Platform	Classification accuracy	Energy Consumption	Reference
Memristive (SiO_x:Ag, fully in hardware)	83%	1 nJ (estimated)	[S3]
Memristive (WO_x)	88.1%	750 pJ (estimated)	[S4]
Photonic memristor ($SrTiO_3$)	90.2%	1.25 μJ (estimated)	[S5]
Magnetic tunnel junction	91.71%	100 fJ	[S6]
Optoelectronic, multimodal (α-In_2Se_3)	86.1%	100 pJ/10 pJ (voltage/light input, estimated)	[S7]
Optoelectronic, multimodal ($MAPbI_3$)	95.3%	61 pJ/584 pJ (voltage/light input)	This work

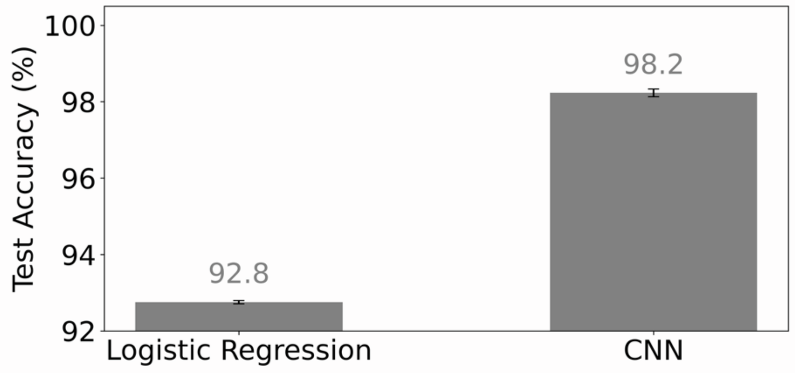


Figure S18. MNIST classification by logistic regression and a single-layer convolutional neural network (CNN).

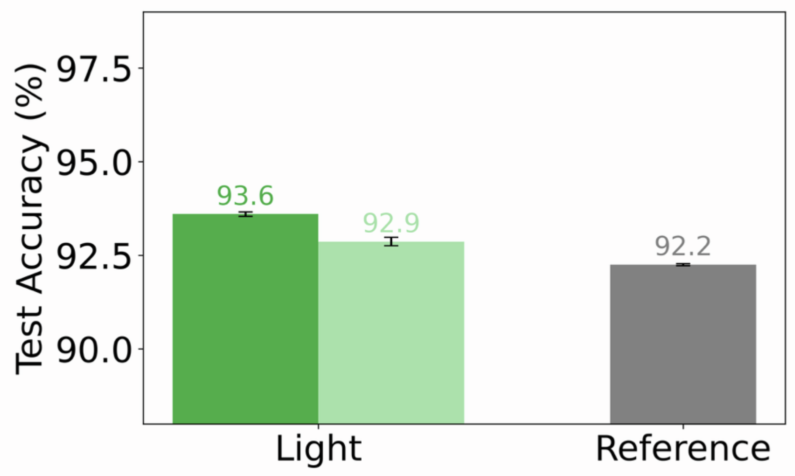


Figure S19. MNIST classification by a light-input network without the 100 mV offset bias, implementing the Row + Column *mapping.*

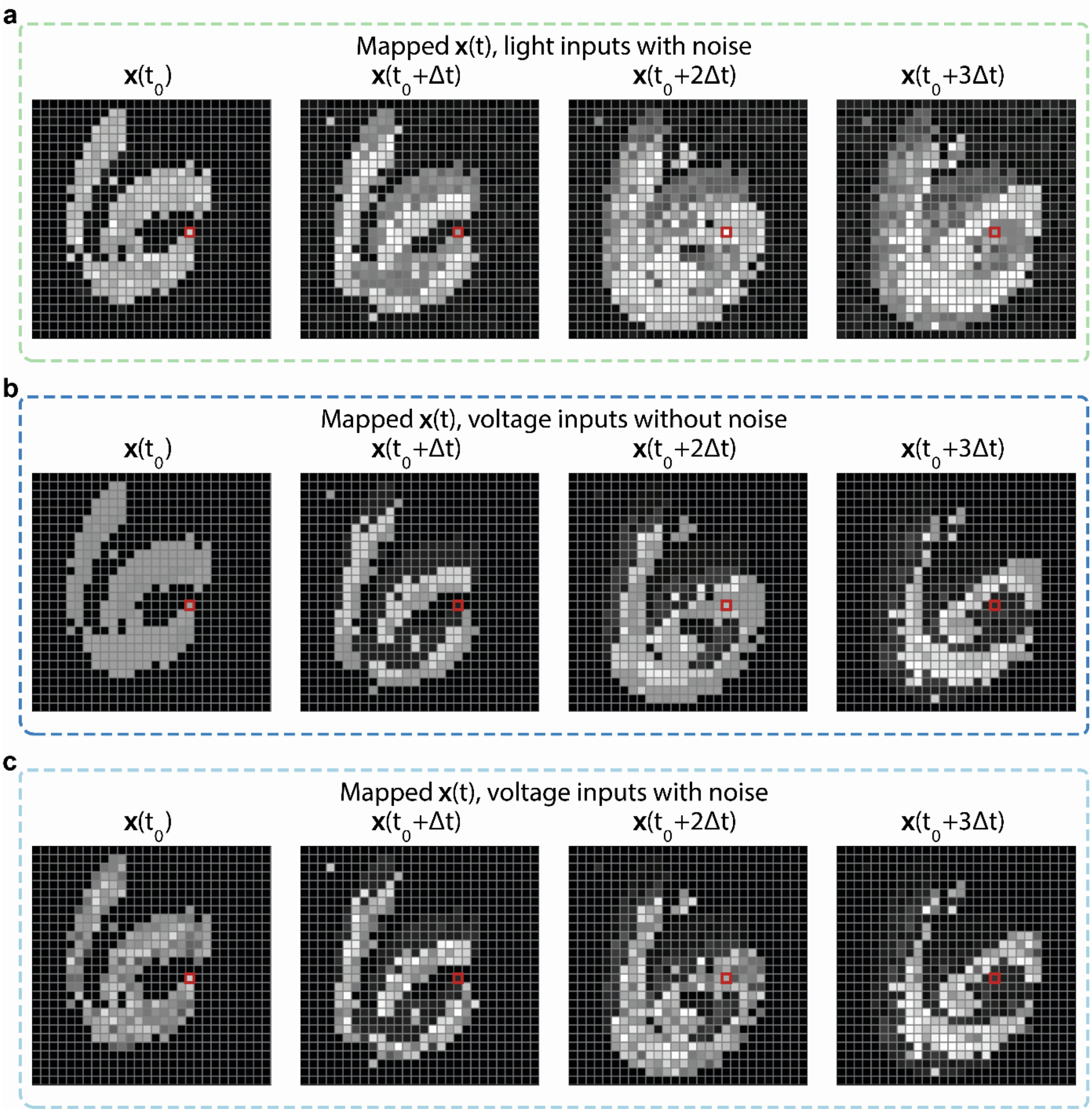


*Figure S20. **(a)** Mapping of each pixel of the N-MNIST frames to values drawn from a normal distribution with a mean and standard deviation of the corresponding 4-bit light input data. **(b)** Mapping to the mean currents obtained for the voltage input measurements. **(c)** Mapping to random values drawn from a normal distribution with means and standard deviations obtained for the corresponding 4-bit voltage inputs.*

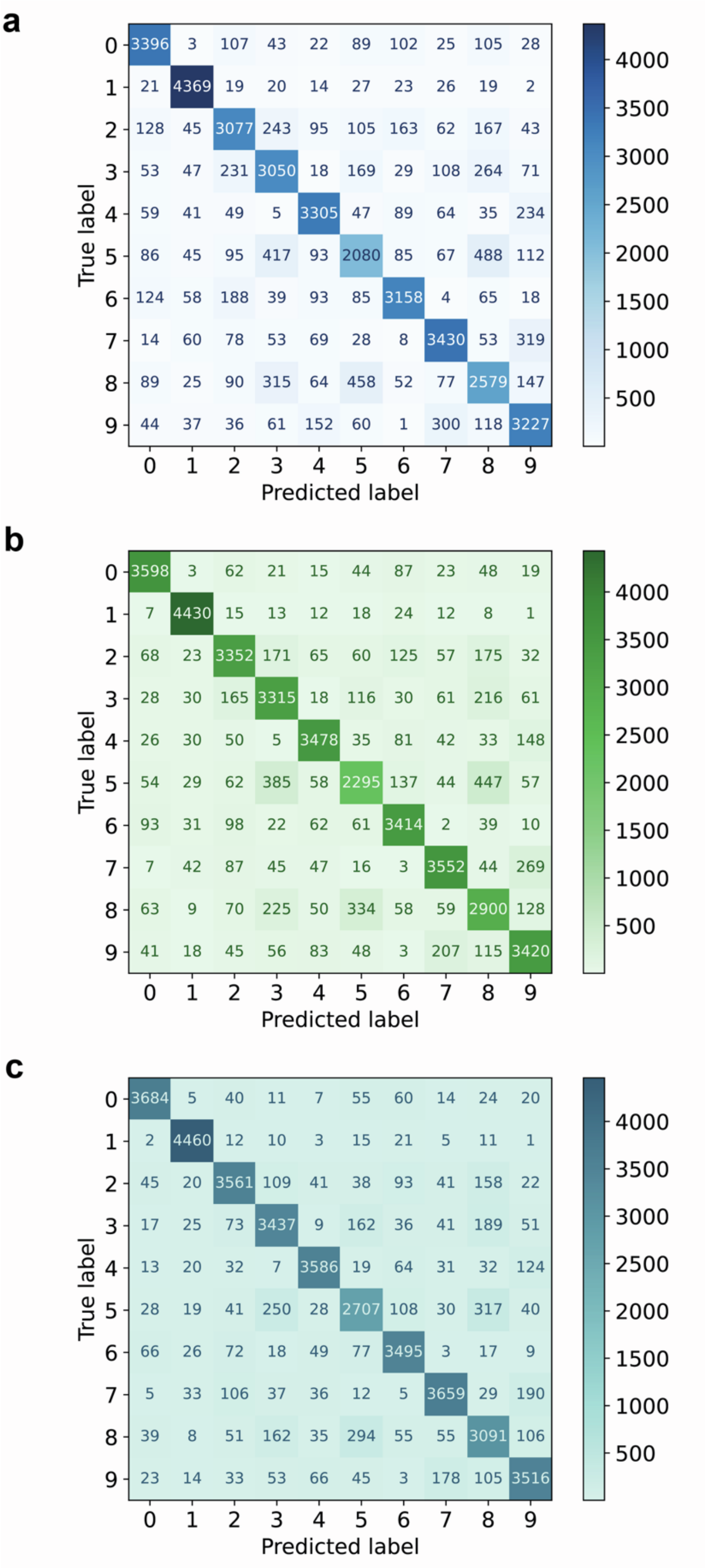


Figure S21. Confusion matrices of handwritten digit classification from video by ***(a)*** *the voltage-input,* ***(b)*** *the light-input, and* ***(c)*** *the multimodal network.*

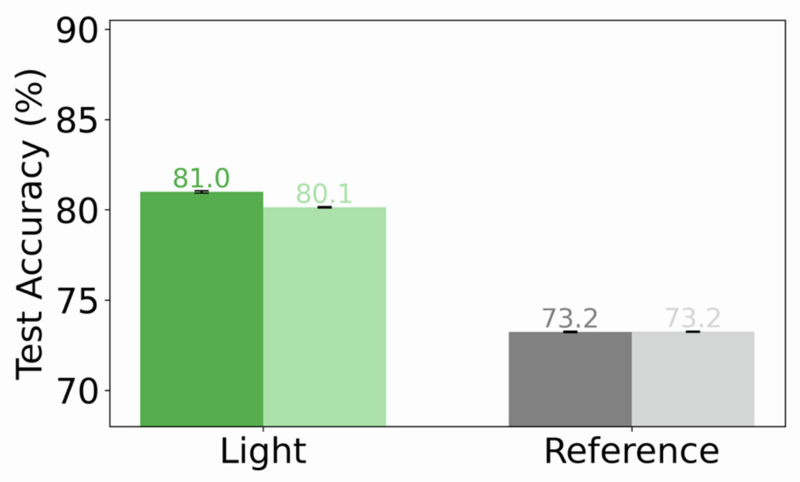


Figure S22. Handwritten digit classification from video by a light-input network without the 100 mV offset bias.

Table S4. Simulation parameters used for the drift-diffusion simulations. A schematic of the simulated device stack is given in Figure S2c and d.

Parameter	Value	Comment
Al_2O_3 Thickness (nm)	15	
Relative permittivity insulator	9	
Electron affinity insulator (eV)	2.5	
Band gap insulator (eV)	5	
Thickness perovskite (nm)	50	
Relative permittivity perovskite	24.1	Taken from ref [S8]
Electron affinity perovskite (eV)	3.9	Taken from ref [S9]
Band gap perovskite (eV)	1.6	Taken from ref [S9]
Effective density of states perovskite conduction band ($1/cm^3$)	$8x10^{18}$	
Effective density of states perovskite valence band ($1/cm^3$)	$8x10^{18}$	
Mobile positive ion density ($1/cm^3$)	$3x10^{17}$	
Immobile negative ion density ($1/cm^3$)	$3x10^{17}$	
Mobility mobile ions (cm^2/Vs)	$1x10^{-10}$	
Work function electrodes (eV)	5.1	
Voltage pulse duration (s)	0.05	
Time between voltage pulses (s)	0.1	

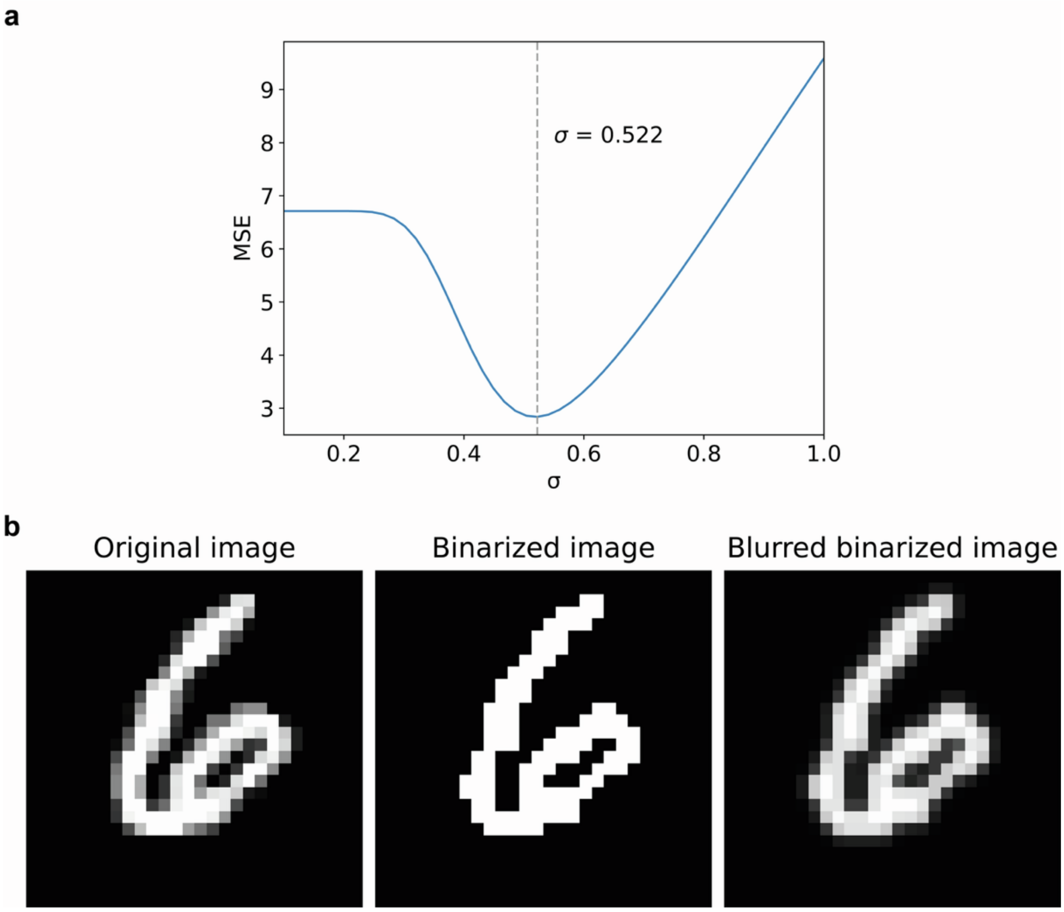


Figure S23. Fitting of a Gaussian blurring kernel to the binarized MNIST data. ***(a)*** *Mean squared error (MSE) obtained by comparing Gaussian blurred binarized MNIST images to the original MNIST dataset as a function of the variance (σ). The variance of the Gaussian blurring kernel with a kernel size of 5 pixels was varied between 0 and 1 to find the lowest MSE value (σ = 0.522).* ***(b)*** *Comparison of an original MNIST image (left) to the binarized image (middle) and the image obtained by applying the Gaussian blurring filter with the optimal variance to the binarized image (right).*

References

S1. Jaeger, H., and Haas, H. (2004). Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science *304*, 78–80. https://doi.org/10.1126/science.1091277.

S2. Cover, T.M. (1965). Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. IEEE Trans. Electron. Comput. *EC-14*, 326–334. https://doi.org/10.1109/PGEC.1965.264137.

S3. Midya, R., Wang, Z., Asapu, S., Zhang, X., Rao, M., Song, W., Zhuo, Y., Upadhyay, N., Xia, Q., and Yang, J.J. (2019). Reservoir Computing Using Diffusive Memristors. Advanced Intelligent Systems *1*, 1900084. https://doi.org/10.1002/aisy.201900084.

S4. Du, C., Cai, F., Zidan, M.A., Ma, W., Lee, S.H., and Lu, W.D. (2017). Reservoir computing using dynamic memristors for temporal information processing. Nat Commun *8*, 2204. https://doi.org/10.1038/s41467-017-02337-y.

S5. Yamazaki, Y., and Kinoshita, K. (2024). Photonic Physical Reservoir Computing with Tunable Relaxation Time Constant. Advanced Science *11*, 2304804. https://doi.org/10.1002/advs.202304804.

S6. Ross, A., Leroux, N., De Riz, A., Marković, D., Sanz-Hernández, D., Trastoy, J., Bortolotti, P., Querlioz, D., Martins, L., Benetti, L., et al. (2023). Multilayer spintronic neural networks with radiofrequency connections. Nat. Nanotechnol. *18*, 1273–1280. https://doi.org/10.1038/s41565-023-01452-w.

S7. Liu, K., Zhang, T., Dang, B., Bao, L., Xu, L., Cheng, C., Yang, Z., Huang, R., and Yang, Y. (2022). An optoelectronic synapse based on α-In2Se3 with controllable temporal dynamics for multimode and multiscale reservoir computing. Nat Electron *5*, 761–773. https://doi.org/10.1038/s41928-022-00847-2.

S8. Brivio, F., Butler, K.T., Walsh, A., and van Schilfgaarde, M. (2014). Relativistic quasiparticle self-consistent electronic structure of hybrid halide perovskite photovoltaic absorbers. Phys. Rev. B *89*, 155204. https://doi.org/10.1103/PhysRevB.89.155204.

S9. Caputo, M., Cefarin, N., Radivo, A., Demitri, N., Gigli, L., Plaisier, J.R., Panighel, M., Di Santo, G., Moretti, S., Giglia, A., et al. (2019). Electronic structure of MAPbI3 and MAPbCl3: importance of band alignment. Sci Rep *9*, 15159. https://doi.org/10.1038/s41598-019-50108-0.