

Efficient visual motion prediction using memristive spatiotemporal compression

Qingjiang Li

qingjiangli@nudt.edu.cn

National University of Defense Technology <https://orcid.org/0000-0001-9779-3198>

ao xu

National University of Defense Technology <https://orcid.org/0009-0003-8469-2146>

lun lu

National University of Defense Technology

Yinan Wang

National University of Defense Technology

youpeng wu

National University of Defense Technology

lang qin

National University of Defense Technology

mingxin deng

National University of Defense Technology

Chang Liu

National University of Defense Technology

Changlin Chen

National University of Defense Technology

Xiao Fu

Shanghai Institute of Technical Physics, Chinese Academy of Sciences

Hangyu Xu

Shanghai Institute of Technical Physics, Chinese Academy of Sciences <https://orcid.org/0000-0002-8204-161X>

binhan du

National University of Defense Technology

Hui Xu

College of Electronic Science and Technology, National University of Defense Technology; Hunan Key Laboratory of Aerospace Intelligent ASIC Technology

Qi Liu

Fudan University <https://orcid.org/0000-0001-7062-831X>

Article

Keywords:

Posted Date: October 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10970001/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: There is **NO** Competing Interest.

Efficient visual motion prediction using memristive spatiotemporal compression

Ao Xu¹, Lun Lu¹, Yinan Wang^{1*}, Youpeng Wu¹, Lang Qin¹, Mingxin Deng¹, Chang Liu¹, Changlin Chen¹, Xiao Fu³, Hangyu Xu³, Binhao Du⁴, Hui Xu¹, Qi Liu^{2*}, Qingjiang Li^{1*}

¹College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China

²State Key Laboratory of Integrated Chips and Systems, College of Integrated Circuits and Micro-Nano Electronics, Frontier Institute of Chip and System, Fudan University, Shanghai, 200083, China

³State Key Laboratory of Infrared Physics, Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China

⁴College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China

Correspondence and requests for materials should be addressed to Y.Wang, Q.Liu, Q.Li

(Email: wangyinan@nudt.edu.cn, qi_liu@fudan.edu.cn, qingjiangli@nudt.edu.cn)

Abstract

Dynamic visual prediction is essential for applications ranging from aerial sensing to autonomous systems and surveillance, but conventional approaches retain multi-frame observations and process full spatial representations, incurring substantial storage, data-transfer and computational overhead. Here we report a memristive spatiotemporal compression architecture for efficient visual motion prediction. Temporally, successive visual changes are integrated into an evolving conductance state, consolidating trajectory information without retaining a multi-frame history. Spatially, analogue row and column current projection compresses the two-dimensional conductance states into one-dimensional currents before prediction. A lightweight memristive multi-layer perceptron directly processes these projection currents to infer future target positions. On a 180-nm memristive platform, the system achieves a mean coordinate error of 1.31 array cells, compared with 1.04 cells for the graphics processing unit (GPU) baseline, while reducing latency by 58.6% and energy consumption by 228 $\times$. Hardware experiments in dual-object scenes further demonstrate trajectory prediction and proximity-risk assessment with accuracy close to that of the GPU baseline, while reducing latency by 69.4% and energy consumption by 684 $\times$. This work advances neuromorphic visual prediction for fast and energy-efficient perception in dynamic environments.

Introduction

Predicting the future motion of objects from visual observations is a fundamental capability underpinning applications such as aerial sensing¹, intelligent transportation², autonomous robotics³ and security surveillance⁴ (Fig. 1c). Unlike instantaneous localization, motion prediction requires information from preceding observations to be retained, the current spatial state to be resolved, and its evolution to be transformed into an estimate of future target motion^{5,6}. The biological visual system performs this task with remarkable efficiency, integrating sparse past observations into internal contextual representations that support anticipation of future motion⁷⁻⁹ (Fig. 1a). Emulating this ability in artificial visual systems by transforming past visual observations into compact representations that support future-motion prediction remains a central challenge (Fig. 1b).

Conventional visual prediction, however, incurs coupled overhead in temporal information retention, spatial data transfer and predictive computation¹⁰⁻¹⁷ (Fig. 1d). Temporal context is commonly maintained through multiple preceding observations that must be stored and repeatedly accessed during inference. Meanwhile, each observation is typically retained as a full two-dimensional spatial representation, resulting in substantial data movement before prediction. A representative class of deep-learning approaches further extracts spatial features from individual frames using convolutional networks (CNN) and models their temporal evolution with recurrent architectures such as long short-term memory (LSTM) networks¹⁰⁻¹⁴; recurrent-free predictive models have also emerged in recent years¹⁵⁻¹⁷. Despite their architectural differences, these approaches share a common bottleneck: redundant spatiotemporal information is retained and processed deep into the prediction pipeline, increasing storage, data-transfer and computational costs^{18,19}.

In this Article, we present a memristive spatiotemporal compression architecture that reduces the amount of temporal and spatial information carried into predictive inference (Fig. 1e). Temporally, successive visual changes are integrated directly into an evolving memristor conductance state, allowing recent motion history to be consolidated into a single physical state without explicitly retaining multiple preceding frames for prediction. Spatially, the resulting two-dimensional conductance state is projected into one-dimensional row and column currents through analogue current summation, reducing the spatial information delivered to the predictor. In the 64×114 memristor-array implementation, 7,296 conductance states are reduced to 178 row and column current features, corresponding to an approximately

41× reduction in spatial dimensionality before prediction. These compressed features are directly processed by a lightweight memristive multilayer perceptron (MLP) to infer future target positions. We experimentally implement the complete architecture on a 180-nm memristive platform, demonstrating motion-history encoding, projection-current localization and hardware future-motion prediction. In the single-object task, the proposed system maintains comparable coordinate accuracy to the GPU-based CNN-LSTM baseline, with a mean coordinate error of 1.31 versus 1.04 array cells, while reducing inference latency by 58.6% and energy consumption by 228× (Fig. 1f). Hardware experiments in dual-object scenes further demonstrate eight-step trajectory prediction with prediction errors close to those of the CNN-LSTM baseline, together with proximity-risk assessment, while reducing latency by 69.4% and energy consumption by 684×. Together, these results establish physical spatiotemporal compression before inference as a promising route towards efficient predictive vision, offering a hardware framework in which sensing, motion-history integration and future-state inference are tightly coupled.

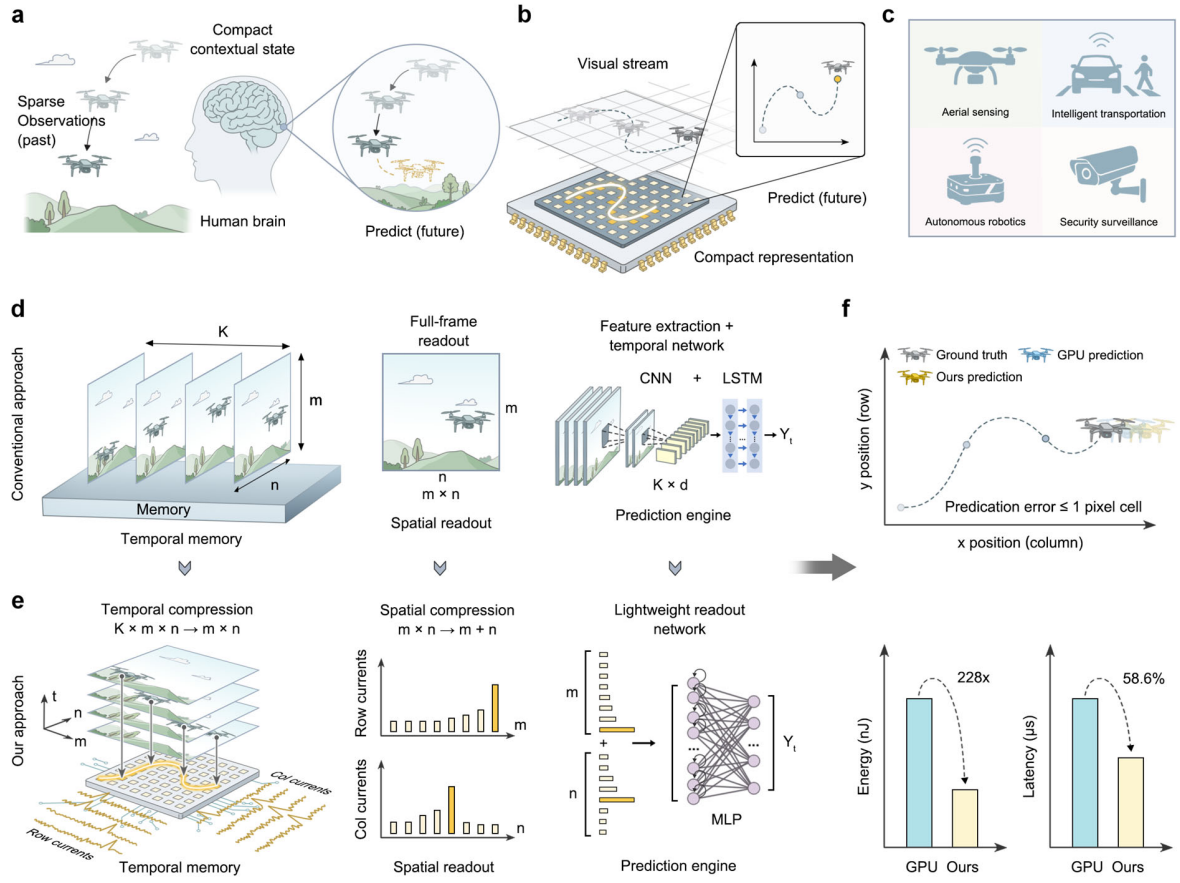


Figure 1. Brain-inspired memristive spatiotemporal compression for predictive vision. **a** Biological vision integrates recent observations to anticipate future motion. **b** Concept of the proposed brain-inspired predictive architecture. **c** Representative application scenarios. **d** Conventional visual prediction based on multiframe buffering, full-frame spatial readout and sequence-based neural processing. **e** Proposed architecture integrating motion history into an evolving conductance state, projecting the two-dimensional state into row and column currents, and performing future-position prediction with a lightweight memristive multilayer perceptron (MLP). **f** Comparison of prediction accuracy, latency and energy consumption with the GPU baseline.

Results

Memristive conductance states compress visual motion history

Memristors provide programmable conductance states that can retain analogue information and evolve under successive electrical stimuli, offering a physical basis for temporal information processing²⁰⁻²⁴. For visual motion encoding, the conductance state must further support stable bidirectional modulation, allowing both positive and negative changes between successive observations to be incorporated into the same evolving representation. The hardware platform integrates a $TiN/TaO_x/HfO_x/TiN$ resistive-switching element with a selector transistor in the one-transistor-one-resistor (1T1R) structure, as illustrated in Fig. 2a. Memristor conductance serves as the state variable for accumulating temporal information. Under alternating potentiation (SET) and depression (RESET) programming, the two resistance states remain clearly separated over 10,000 cycles. The SET operation is performed using $1\ \mu s$ pulses of $2.3\ V$ with a $2\ V$ gate bias, whereas the RESET operation uses $-3.8\ V$ pulses with a $3\ V$ gate bias, as shown in Fig. 2b(see Supplementary Note 1). This stable bidirectional modulation allows successive visual changes to accumulate directly in the conductance state, rather than being stored as separate image frames.

The conductance state was updated according to the intensity change between consecutive frames. For a pixel cell at position (i, j) , the local intensity change is given by

$$\Delta P_{ij(t)} = P_{ij(t)} - P_{ij(t-1)} \quad (1)$$

Following the temporal-contrast principle used in event-based vision, $\Delta P_{ij(t)}$ is compared with symmetric positive and negative thresholds. Positive changes exceeding $+\theta$ trigger SET, negative changes below $-\theta$ trigger RESET, whereas subthreshold changes leave the conductance state unchanged

$$\begin{cases} \Delta P_{ij} > \theta \rightarrow SET \\ \Delta P_{ij} < -\theta \rightarrow RESET \\ |\Delta P_{ij}| \leq \theta \rightarrow No\ update \end{cases} \quad (2)$$

Fig. 2c illustrates a representative three-bit encoding scheme for level-coded conductance updates. The scheme comprises one polarity bit that selects SET or RESET and two magnitude bits that define four levels, L1-L4. After thresholding, the magnitude of each accepted inter-frame change is normalized to a fixed full-scale contrast range as

$$\eta_{ij} = \frac{|\Delta P_{ij}|}{P_{FS}}, \quad |\Delta P_{ij}| > \theta \quad (3)$$

where $P_{FS} = 255$ denotes the full-scale inter-frame intensity change. The normalized magnitude is then quantized into four levels, so that larger visual changes are assigned progressively stronger programming pulses.

Positive intensity changes are mapped to SET voltages of 0.8-1.4 V, whereas negative changes are mapped to RESET voltages of -1.0 to -1.6 V. Single-pulse measurements from 32 independent 1T1R devices show graded increases and decreases in conductance during SET and RESET, respectively, with the magnitude of ΔG increasing across the programmed levels. This bidirectional response directly maps the polarity and relative magnitude of inter-frame intensity changes onto conductance-state updates. The three-bit scheme was used in this work as a representative encoding configuration to capture the polarity and relative magnitude of inter-frame changes. The encoding precision is not an intrinsic constraint of the architecture and can be configured according to task and peripheral-circuit requirements^{24,25}.

Fig. 2d shows the mixed-signal circuit that implements the encoding rule. The current pixel value $P_{ij(t)}$ is subtracted from its registered value $P_{ij(t-1)}$ to obtain ΔP_{ij} . As defined in Eq. (2), ΔP_{ij} is then compared with symmetric thresholds to determine SET, RESET or no programming operation (see Supplementary Note 7). For suprathreshold changes, the control circuit normalizes the absolute frame difference to the fixed contrast range and quantizes it into the predefined magnitude levels. The sign of ΔP_{ij} determines whether a SET or RESET operation is selected, whereas its magnitude is encoded into the corresponding digital-to-analogue converter (DAC) level to generate the programming voltage applied to the selected 1T1R cell. Temporal differencing, threshold comparison and voltage-level encoding are thus performed by the peripheral control circuitry, while the corresponding programming operations modify the device conductance and accumulate motion information in the conductance state of the memristor array (see Supplementary Note 2).

The temporal-compression capability of the conductance state was examined using four distinct motion trajectories converging on the same terminal position from different directions (Fig. 2e). The same target reached the same central position from the left, right, top and bottom, producing an identical instantaneous image at the final time point despite the different preceding trajectories. The conventional frame-based image sensor reporting only the current frame would therefore provide no information about the preceding trajectory^{26,27}. By contrast, sequential conductance updating produced four distinct terminal states, each retaining a trajectory along the corresponding approach direction (Fig. 2f). The array state thus

consolidates information distributed across multiple frames into a single physical representation. The terminal conductance states exhibited dominant responses of 0.95, 0.95, 0.93 and 0.93 for motion from the left, right, top and bottom, respectively, whereas all non-corresponding responses remained at or below 0.03 (Fig. 2g). These results demonstrate that the proposed architecture preserves distinct motion histories within the conductance state even when the terminal images are identical, enabling trajectory information to be retained without explicit multiframe storage. The same in situ motion-state accumulation was further verified using dynamic visual scenes (see Supplementary Note 3). A direct comparison with transient event encoding further distinguishes the accumulated motion-state representation from instantaneous temporal-contrast encoding (see Supplementary Note 4).

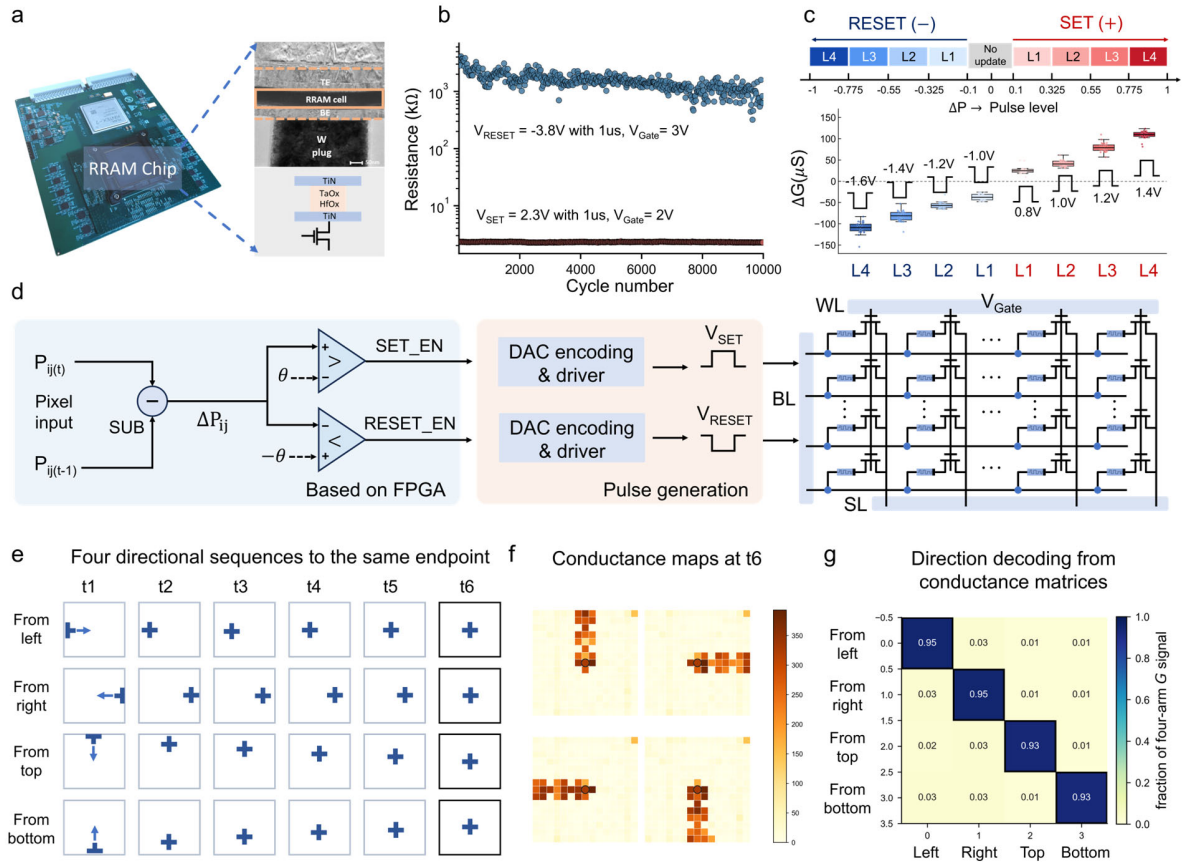


Figure 2. Memristive conductance states compress visual motion history. **a** Photograph of the RRAM computing platform, cross-sectional image of the resistive-switching cell and schematic of the 1T1R structure. **b** Endurance characteristics of a representative 1T1R cell under alternating SET and RESET programming. **c** Three-bit temporal-contrast encoding and graded bidirectional conductance modulation. **d** Mixed-signal circuit for converting frame-to-frame intensity changes into array-programming pulses. **e** Four motion trajectories originating from the left, right, top and bottom and terminating at the same central position. **f** Experimentally measured conductance maps of the 15×15 RRAM array for the four motion trajectories. **g** Directional-response matrix extracted from the terminal conductance states.

Projection-current readout compresses spatial representations

Conventional visual processing generally accesses complete two-dimensional image or state maps before extracting target position, resulting in spatial data movement and processing overhead that increases with image resolution^{28,29}. In our architecture, each $a \times b$ frame from the incoming video stream is first spatially downsampled by pooling to match the $m \times n$ memristor array. Frame-by-frame updates associated with a moving target trigger SET operations in the corresponding array cells, producing a localized increase in conductance. Under a uniform read voltage, this conductance enhancement gives rise to peaks in the summed row and column currents. The column-current peak identifies the horizontal target position, whereas the row-current peak identifies the vertical position. Target coordinates can therefore be obtained directly from the two orthogonal current projections without reading out the complete two-dimensional conductance map (Fig. 3a).

Under a uniform read voltage V_{read} , currents from devices along each row or column are naturally summed according to Kirchhoff's Current Law, producing two orthogonal projections of the conductance distribution

$$I_i^{row}(t) = V_{read} \sum_{j=1}^n G_{ij}(t) \quad (4)$$

$$I_j^{col}(t) = V_{read} \sum_{i=1}^m G_{ij}(t) \quad (5)$$

where $G_{ij}(t)$ denotes the conductance of the cell at row i and column j at time t . $I_i^{row}(t)$ and $I_j^{col}(t)$ denote the summed currents along row i and column j , respectively. Because target-induced SET operations locally increase the conductance, the corresponding row and column projections exhibit pronounced current peaks. The target coordinates are consequently obtained from the peak positions of the two current vectors

$$\tilde{x}(t) = \arg \max_j I_j^{col}(t) \quad (6)$$

$$\tilde{y}(t) = \arg \max_i I_i^{row}(t) \quad (7)$$

where $\tilde{x}(t)$ and $\tilde{y}(t)$ denote the estimated horizontal and vertical coordinates at time t , respectively. $\arg \max$ returns the index of the maximum element in the corresponding current vector.

Thus, the column-current peak determines the horizontal coordinate, whereas the row-current peak determines the vertical coordinate. Target position can therefore be directly extracted from the spatially compressed current representation without recovering the complete two-dimensional conductance state. Repeated readout over the visual stream yields a coordinate sequence $\{t, \tilde{x}(t), \tilde{y}(t)\}$.

We next examined whether the compressed current representation preserves target position during continuous motion using a 360×240 video sequence from the CDnet 2014 benchmark dataset³⁰. The conductance states progressively formed a motion trajectory as successive visual changes accumulated in the array (Fig. 3b). The peak of the column-current projection closely followed the independently tracked horizontal coordinate throughout the sequence (Fig. 3c), while the row-current peak showed corresponding agreement with the vertical coordinate (Fig. 3d). Combining the two peak positions recovered a two-dimensional trajectory that closely matched the tracker-derived reference (Fig. 3e). These results show that, despite spatial compression through projection-current readout, target-coordinate information remains preserved and directly accessible.

We further evaluated the projection-current readout using additional sequences from the Night Videos, Shadow and Bad Weather categories of CDnet2014 (Fig. 3f). Across all three scenarios, the projection-derived trajectories remained closely aligned with the independently tracked references. These results show that target-position information remains accessible from the compressed projection currents under different illumination and background conditions. Additional dynamic-scene encoding results under diverse traffic and environmental conditions are provided in Supplementary Note 5.

The projection-current readout provides both target localization and a compact input for subsequent motion prediction. The current peaks directly determine the instantaneous target coordinates, whereas the complete row and column current vectors are retained as the compressed representation for future-motion prediction. Together with temporal accumulation in the conductance state, analogue projection thus completes the spatiotemporal compression pathway used by the prediction system in the following section. The same sensory-memory hardware pathway was further validated for unified static recognition and dynamic perception (see Supplementary Note 6).

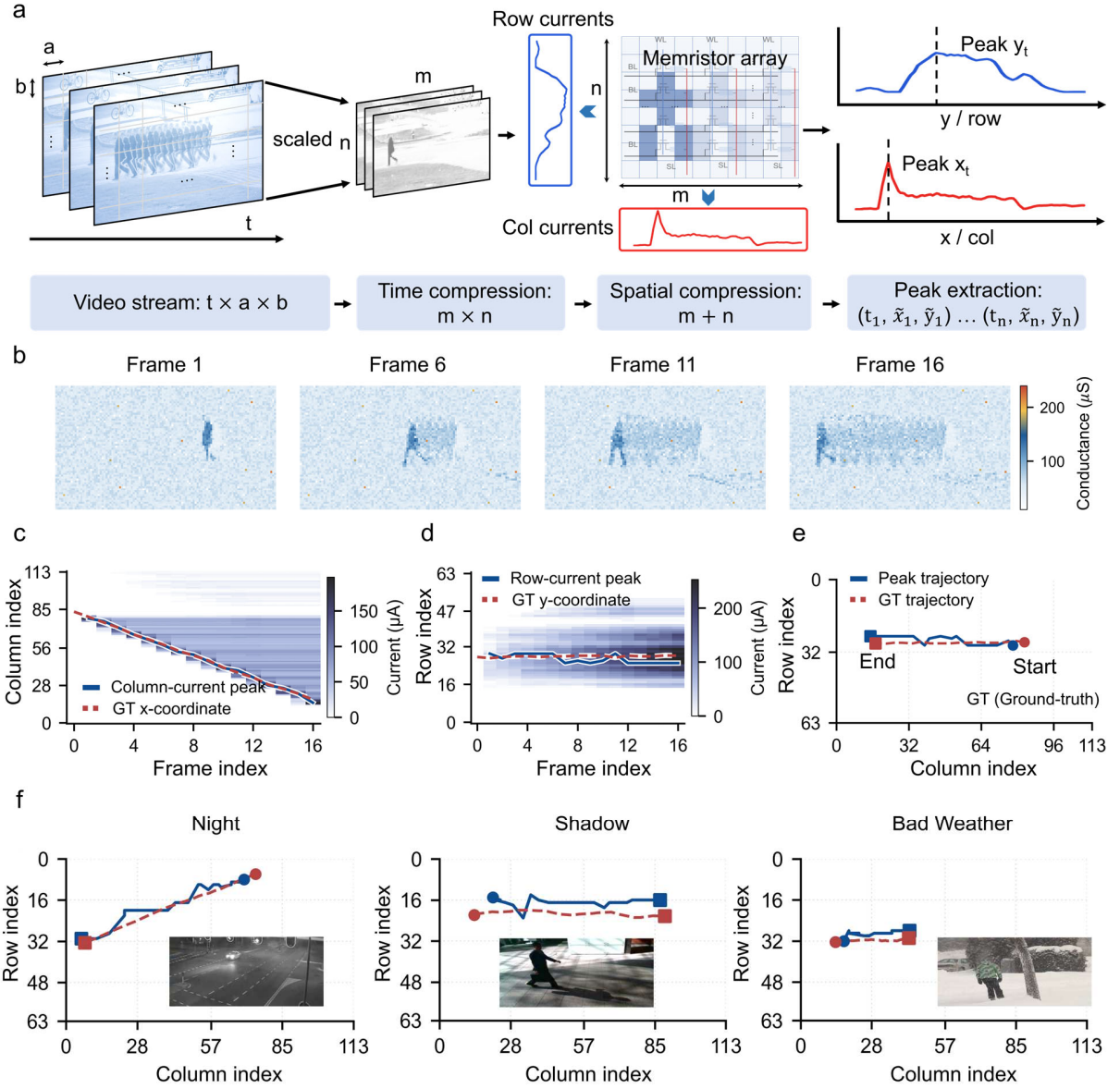


Figure 3. Projection-current readout compresses spatial states while preserving target position. **a** Operating principle of the projection-current readout. **b** Representative conductance states recorded at selected frames during single-target motion, showing the progressive accumulation of the target trajectory. **c** Temporal evolution of the column-current distribution. The column-current maximum provides the estimated horizontal coordinate $\tilde{x}$, which is compared with the independently tracked coordinate x . **d** Temporal evolution of the row-current distribution. The row-current maximum provides the estimated vertical coordinate $\tilde{y}$,

which is compared with the independently tracked coordinate y . **e** Two-dimensional trajectory reconstructed by combining the row- and column-current maxima. The current-derived trajectory is compared with the tracker-derived reference from the initial to the terminal target position. **f** Coordinate recovery under representative Night Videos, Shadow and Bad Weather conditions from the CDnet2014 dataset.

Hardware future-motion prediction

Future target motion cannot be inferred from instantaneous position alone, but requires motion history accumulated over preceding observations^{31,32}. In our architecture, this motion history is already embedded in the evolving conductance state and is retained after row-column projection, enabling future motion to be inferred from a single current representation rather than an explicit sequence of preceding frames (Fig. 4a). For each 64×114 conductance state, the array generates a 178-dimensional vector comprising 64 row currents and 114 column currents. The current peaks provide the instantaneous target coordinates, whereas the complete projection-current vector contains the distributed motion-history information and is supplied to a $178 \times 32 \times 2$ multilayer perceptron. The two output neurons predict the horizontal and vertical displacements, Δx and Δy , in the subsequent frame; adding these displacements to the projection-derived current position yields the predicted future position.

The complete prediction pathway was implemented on the mixed-signal memristive platform shown in Fig. 4b. The system comprises three memristor-based processing elements (PEs), each integrating one 8-Kb memristor array (see Supplementary Note 13). PE1 serves as the sensor-memory front end for motion-state accumulation and row-column current projection, whereas PE2 and PE3 implement the memristive predictor. The predictor was trained using spatially augmented experimental data. Spatial translations and reflections were applied jointly to the measured 178-dimensional row-column current profiles and their coordinate labels, yielding 10,784 training samples and 2,759 validation samples. The original untransformed trajectory was reserved for hardware testing. The trained weights were subsequently differentially programmed into PE2 and PE3 for current-domain inference (see Supplementary Note 8).

We next verified the two hardware components that underpin future-motion prediction: the history-bearing sensory state and the programmed predictor. The measured conductance maps in Fig. 4c show that, as the target moves, the localized conductance response shifts with its current position while residual

conductance changes from preceding positions are retained along the motion path. Each instantaneous array state therefore contains both the current-location cue and a spatially distributed record of recent motion, providing the physical basis for extracting predictive information from a single projection-current vector. The trained predictor was then differentially programmed into the memristor arrays using closed-loop write-read-verify programming (see Supplementary Note 14). The programming-error distribution is centred near zero (Fig. 4d), indicating close agreement between the implemented and target conductances and confirming faithful hardware realization of the prediction network.

We then evaluated the prediction accuracy over the complete hardware test trajectory. The predicted positions closely follow the reference trajectory, including segments with pronounced changes in motion direction (Fig. 4e), indicating that the compressed current representation preserves the motion cues required for trajectory extrapolation. The signed errors along both spatial coordinates are distributed around zero, with no evident directional bias (Fig. 4f). Prediction accuracy was further quantified by the mean two-dimensional coordinate error

$$\bar{e} = \frac{1}{N} \sum_{t=1}^N \sqrt{(x_t^{pred} - x_t^{GT})^2 + (y_t^{pred} - y_t^{GT})^2} \quad (8)$$

where x_t^{pred} and y_t^{pred} denote the predicted horizontal and vertical coordinates at frame t . x_t^{GT} and y_t^{GT} denote the corresponding ground-truth coordinates, and $N = 236$ is the number of test frames. As shown in Fig. 4g, the proposed hardware predictor achieves an average error of 1.31 array cells, compared with 1.04 array cells for the CNN-LSTM baseline, a difference of only 0.27 array cell (see Supplementary Note 9).

We finally compared the system-level latency and energy consumption for one complete prediction (Fig. 4h). The proposed hardware requires 15.75 μs and 12.85 μJ per inference, compared with 38 μs and 2.93 mJ for the GPU-based CNN-LSTM baseline, corresponding to a 58.6% reduction in latency and a 228 $\times$ reduction in energy consumption (see Supplementary Note 15 and Supplementary Table 1). Together, these results demonstrate that the proposed architecture enables hardware future-motion prediction with accuracy comparable to the GPU baseline, while substantially reducing inference latency and energy consumption.

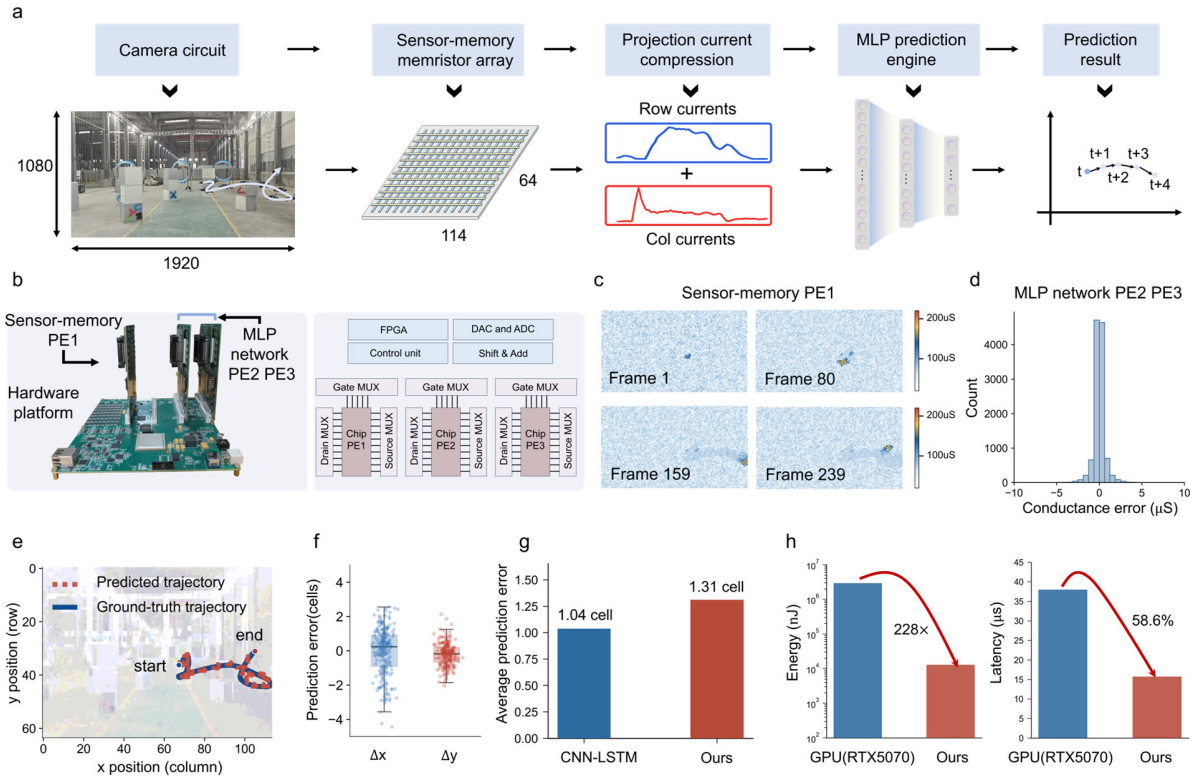


Figure 4. Hardware future-motion prediction. **a** Processing pathway from visual acquisition to target-position prediction. **b** Hardware platform. PE1 performs sensory-state encoding and projection-current readout, whereas PE2 and PE3 jointly implement the memristive MLP using differential conductance mapping; inter-layer control is provided by the FPGA. **c** Experimentally measured conductance states at four representative frames, showing the evolution of the motion-history representation. **d** Distribution of conductance-programming errors for the mapped MLP weights. **e** Ground-truth and hardware-predicted trajectories. **f** Distributions of the signed coordinate errors along the horizontal and vertical directions. **g** Mean two-dimensional coordinate error of the proposed hardware predictor and the GPU-based CNN-LSTM baseline. **h** Energy consumption and latency per single-object prediction. Relative to the GPU-based CNN-LSTM, the memristive system achieves 228× lower energy consumption and a 58.6% reduction in latency.

Hardware multi-object trajectory and proximity-risk prediction

Predictive perception in multi-object scenes is important for collision avoidance and safety assessment, where the future trajectories of interacting targets must be resolved while preserving their identities^{33,34}. To extend projection-current readout to multiple targets, we use a window-conditioned row-current readout to associate the orthogonal row and column responses of individual objects. We experimentally evaluated this approach using two interacting unmanned aerial vehicles (UAVs) mapped onto the 64×114 memristor array (Fig. 5a). Two horizontal peak positions are first identified from the column-current projection. Row currents are then selectively read within 9-column windows centred on the corresponding peaks, allowing the vertical coordinate associated with each target to be recovered (see Supplementary Note 11). This window-conditioned readout reconstructs the two target positions directly from the compressed projection currents without recovering the complete two-dimensional conductance state.

The experimentally measured conductance states show two evolving motion traces as the UAVs approach, interact and subsequently separate (Fig. 5b). The corresponding projection-current responses track the horizontal and vertical positions of both targets throughout the sequence (Fig. 5c). Window-conditioned association of the orthogonal responses then recovers the two-dimensional trajectories of the two UAVs, which remain closely aligned with the independently tracked references (Fig. 5d). These hardware measurements demonstrate that projection-current compression can retain and associate the positional information of multiple moving targets.

Having established multi-target coordinate recovery, we next used the complete projection-current representation for multi-step trajectory prediction. The 64 row currents and 114 column currents form a 178-dimensional input to a $178 \times 32 \times 16$ memristive MLP, whose 16 outputs represent eight future two-dimensional positions of the intruder UAV (see Supplementary Note 10). As shown in Fig. 5e, across prediction horizons from one to eight frames, the median errors of the proposed hardware predictor remain in the same range as those of the CNN-LSTM baseline (see Supplementary Note 12). The relatively small accuracy difference across the prediction horizon indicates that the compressed current representation retains the motion information required for multi-step future-position inference.

We further used the predicted intruder trajectory to assess future proximity between the two UAVs. Because only two-dimensional image-plane coordinates are considered, the resulting risk represents

projected proximity rather than physical collision probability. The proximity risk was defined from the predicted image-plane separation as

$$RISK(t) = \exp\left[-\frac{d(t)}{d_{safe}}\right], \quad d(t) = \|P_I(t + h|t) - P_E(t + h)\|_2 \quad (9)$$

where $P_I(t + h|t)$ denotes the h -step-ahead predicted position of the Intruder UAV, whereas $P_E(t + h)$ denotes the reference position of the Ego UAV at the corresponding future frame. d_{safe} is the prescribed safe-separation distance. As the projected separation decreases, the predicted proximity risk rises and reaches its maximum near the closest image-plane approach, before declining as the two trajectories diverge (Fig. 5f). This agreement in both risk evolution and peak timing shows that the predicted trajectories capture the interaction dynamics required for future proximity assessment.

Risk-identification performance was further evaluated using precision, recall, F1 score, area under the precision-recall curve (AUPRC) and false-alarm rate (Fig. 5g). The proposed method shows overall performance comparable to the CNN-LSTM baseline, with slightly higher precision and AUPRC and a lower false-alarm rate, while the baseline achieves higher recall and F1 score. These results indicate that the compressed projection-current representation retains sufficient trajectory information for reliable image-plane proximity assessment.

The system-level benefit of the compressed representation was finally evaluated for one complete eight-step trajectory prediction. The memristive hardware implementation requires $15.90 \mu s$ and $13.29 \mu J$ per prediction, compared with $52 \mu s$ and $9091 \mu J$ for the GPU-based CNN-LSTM baseline (Fig. 5h). This corresponds to a 69.4% reduction in latency and $684\times$ lower energy consumption (see Supplementary Note 16 and Supplementary Table 2). Together, these hardware experiments show that spatiotemporal compression can support multi-target association, multi-step trajectory prediction and downstream risk assessment from compact current representations, providing an efficient hardware pathway from dynamic visual sensing to predictive decision-relevant information.

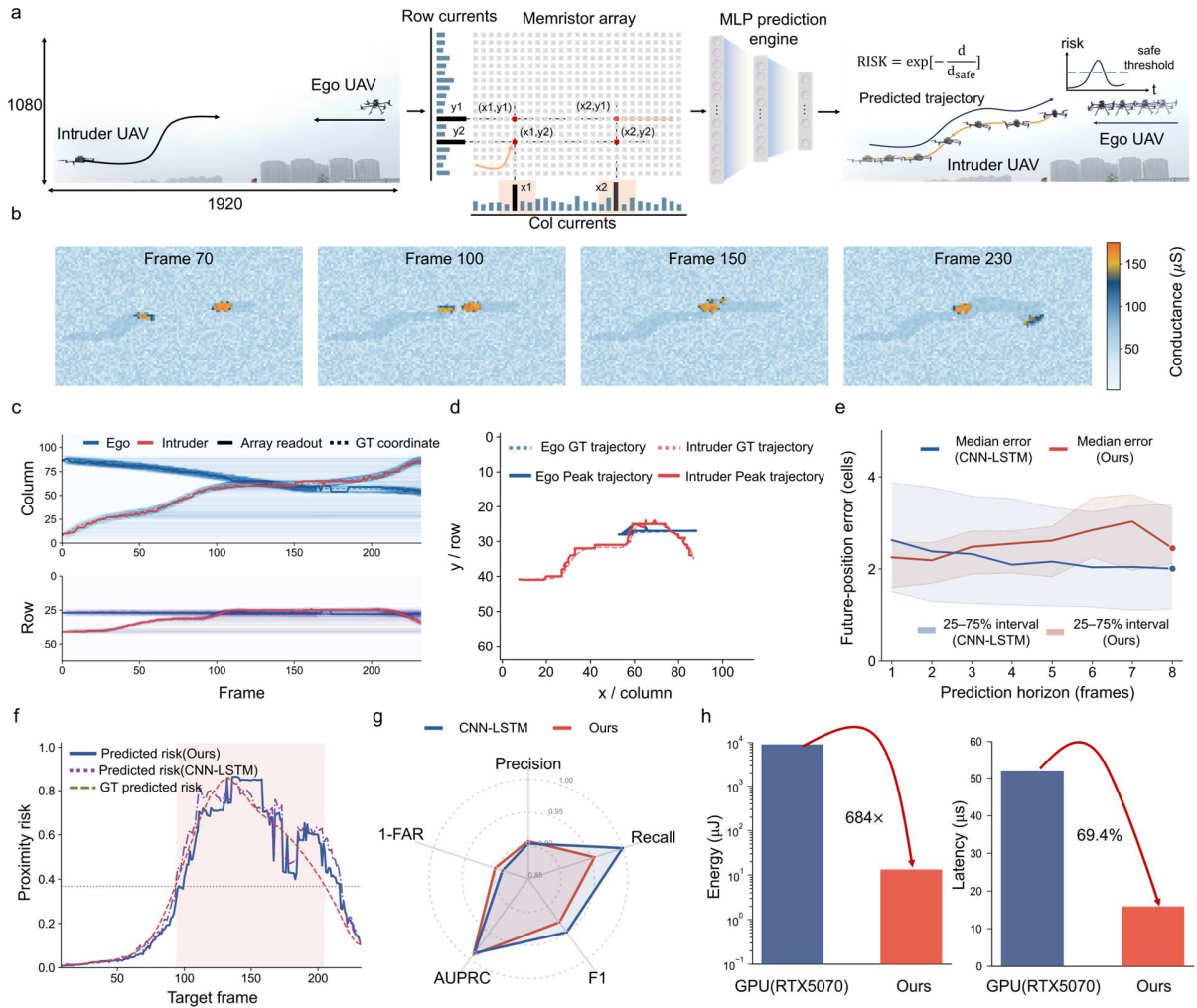


Figure 5. Hardware multi-object trajectory and proximity-risk prediction. **a** Hardware framework for two-object localization, multi-step trajectory prediction and proximity-risk assessment. **b** Experimentally measured conductance states of the memristor array at four representative frames during the interaction between two unmanned aerial vehicles (UAVs). **c** Column-current and window-conditioned row-current responses and the corresponding coordinates of the ego and intruder UAVs, compared with independently tracked references. **d** Two-dimensional trajectories reconstructed from the projection-current peaks and the corresponding ground-truth trajectories. **e** Median future-position error over prediction horizons of one to eight frames; shaded regions indicate the 25%-75% percentile intervals. **f** Predicted image-plane proximity risk derived from the future trajectories, compared with the

359 tracker-derived reference and baseline prediction. **g** Comparison of proximity-risk
360 identification performance using precision, recall, F1 score, area under the precision-
361 recall curve (AUPRC) and one minus the false-alarm rate (1-FAR). **h** Comparison
362 of energy consumption and prediction latency between the proposed hardware and
363 GPU baseline, showing 684× lower energy consumption and 69.4% lower latency.

364

Discussion

In this work, we have developed a memristive spatiotemporal compression architecture for efficient visual motion prediction. At the temporal-representation level, successive visual changes are integrated into an evolving conductance state, allowing recent motion history to be retained without explicit multiframe storage. At the spatial-representation level, analogue row and column current projection compresses the two-dimensional conductance state into compact one-dimensional currents before prediction. The resulting projection-current representation is then processed by a lightweight memristive multilayer perceptron for future-position inference. We experimentally implemented the architecture on a 180-nm memristive platform. In single-object prediction, the system achieves a mean coordinate error of 1.31 array cells, close to the 1.04 cell error of the GPU baseline, while reducing latency by 58.6% and energy consumption by $228\times$. Hardware experiments in two-object scenes further demonstrate target association, trajectory prediction and proximity-risk assessment, with prediction accuracy comparable to the GPU baseline while reducing latency by 69.4% and energy consumption by $684\times$. These results show that visual motion prediction can be shifted away from retaining and repeatedly processing high-dimensional multiframe observations towards extracting motion-relevant information in the physical sensory state before inference. By coupling temporal memory, spatial readout and predictive computation within a common hardware pathway, spatiotemporal compression before predictive inference provides a route towards fast and energy-efficient predictive vision in dynamic environments.

Methods

Memristor crossbar array fabrication

A 64×128 (8 Kb) 1T1R crossbar array was fabricated via a hybrid integration process that combines 180-nm complementary metal-oxide-semiconductor (CMOS) field-effect transistors with bipolar $TiN/TaO_x/HfO_x/TiN$ memristors. The front-end transistor array was first fabricated in a commercial CMOS foundry, using transistors with a channel width and length of $5\ \mu m$ and $350\ nm$, respectively. The memristor stack was then integrated in-house onto the drain side of each transistor. A 40-nm TiN bottom electrode was deposited onto a polished W plug by reactive sputtering. Subsequently, a 10-nm HfO_x layer and a 50-nm TaO_x layer were formed as the switching stack using atomic-layer deposition and physical vapor deposition, respectively. A 30-nm TiN top electrode was deposited to complete the $TiN/TaO_x/HfO_x/TiN$ structure.

The resulting 1T1R cells were defined by dry etching, yielding an effective device area of approximately $0.5 \times 0.5 \mu\text{m}^2$.

Hardware platform

A mixed-signal hardware platform was developed to support analogue operation of the memristor chips (Fig. 4b). An XCKU040 FPGA on the main control board provides global timing and system control, while XC7K325T FPGAs on the array sub-cards perform local addressing and operation sequencing. The platform integrates analogue-to-digital converters (ADCs, AD9650 and AD7606B), digital-to-analogue converters (DACs, DAC5682Z and AD5532), and analogue multiplexers (TMUX1112 and MUX36S16) for array programming and readout. The DACs generate the multilevel voltage waveforms required for SET, RESET and read operations, whereas the ADCs digitize the sensed array outputs for subsequent processing. The platform communicates with a host computer through a high-speed Ethernet interface. A software application programming interface (API) translates task-level commands into hardware operations, including array addressing, pulse generation, current acquisition and dataflow control, enabling different computational tasks to be deployed on the same platform.

FPGA implementation of temporal-difference processing

Input videos were converted into 8-bit grayscale sequences and downsampled by pooling to 64×114 , matching the active sensory-array region used in both single and dual-object experiments. Temporal differencing was implemented using a 64-pixel-parallel FPGA pipeline. Spatially corresponding pixels from consecutive frames were buffered and subtracted in parallel, and the resulting intensity changes were compared with preset thresholds to generate mutually exclusive SET and RESET commands. The FPGA outputs the programming addresses, operation polarities and voltage-level codes, whereas the corresponding analogue pulses are generated by the DAC and array-driving circuitry (see Supplementary Note 7).

Motion-state evolution and projection-current processing

Before each visual sequence, the initial conductance state of the sensory array was measured as a reference. During sequence processing, the array evolved continuously without frame-wise global resetting: suprathreshold temporal changes triggered level-coded SET or RESET operations at the corresponding cells, whereas subthreshold changes produced no programming pulse. The conductance state after each frame

therefore served directly as the initial state for the subsequent frame, enabling motion history to accumulate physically in the array.

Row and column currents were measured at a read voltage of 0.2 V. For the 64×114 array, the resulting 64 row-current and 114 column-current values were concatenated into a 178-dimensional feature vector for coordinate extraction and motion prediction. Reference coordinates were obtained from the centres of independently tracked target bounding boxes and mapped to array coordinates; prediction error was calculated as the Euclidean distance between predicted and reference positions in array cells.

For dual-object scenes, two horizontal peak positions were first identified from the global column-current profile. Window-conditioned row-current profiles were then obtained within 9-column windows centred on these peaks, and the maximum of each profile was paired with its corresponding horizontal position. This procedure yielded two associated two-dimensional target positions directly from the projection currents without reconstructing the complete conductance map.

Hardware implementation of the projection-current predictor

The single-object predictor used a $178 \times 32 \times 2$ MLP network, with the two outputs representing the horizontal and vertical displacements relative to the current position obtained from the projection-current peaks. The predicted next-frame position was obtained by adding these displacements to the current coordinates. For eight-step prediction in dual-object scenes, the output dimension was extended to 16, corresponding to eight future two-dimensional positions.

Signed weights were represented using differential conductance pairs. Accordingly, the 178×32 first layer was mapped to a 178×64 physical conductance matrix and partitioned into 64×64 , 64×64 and 50×64 blocks across PE2 and PE3. The single-object 32×2 output layer was mapped to a 32×4 conductance matrix, whereas the dual-object 32×16 output layer was mapped to a 32×32 matrix. Partial currents from different first-layer partitions were digitized and accumulated before nonlinear activation.

Input features were normalized and quantized to 8-bit unsigned integers for bit-serial array inference. Differential column currents were digitized and subtracted to recover signed outputs, while bit-plane accumulation, nonlinear activation and inter-layer sequencing were performed by the FPGA. Offline-trained weights were scaled to the available conductance range and programmed using closed-loop tuning. Hardware inference fidelity was evaluated by comparing the measured array outputs with those from ideal

numerical matrix-vector multiplication under identical inputs. Further details of weight mapping, bit-serial execution and hardware characterization are provided in [Supplementary Notes 8 and 10](#).

Data Availability

The authors declare that the main data supporting the findings of this study are available within the article and its Supplementary Information files. Extra data are available from the corresponding author upon request.

Code Availability

The codes for plotting the data are available from the corresponding author via email.

References

1. Amala Arokia Nathan, R. J. et al. An autonomous drone swarm for detecting and tracking anomalies among dense vegetation. *Commun. Eng.* **4**, 205 (2025).
2. Yan, X., Zou, Z., Feng, S. et al. Learning naturalistic driving environment with statistical realism. *Nat. Commun.* **14**, 2037 (2023).
3. Dürr, P. et al. Outplaying elite table tennis players with an autonomous robot. *Nature* **652**, 886–891 (2026).
4. Liang, J., Jiang, L., Niebles, J. C., Hauptmann, A. G. & Fei-Fei, L. Peeking into the future: predicting future person activities and locations in videos. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 5725–5734 (IEEE, 2019).
5. Rudenko, A., Palmieri, L., Herman, M., Kitani, K. M., Gavrila, D. M. & Arras, K. O. Human motion trajectory prediction: a survey. *Int. J. Robot. Res.* **39**, 895–935 (2020).
6. Nazari, K., Mandil, W., Santello, M. et al. Bioinspired trajectory modulation for effective slip control in robot manipulation. *Nat. Mach. Intell.* **7**, 1119–1128 (2025).
7. Berry, M. J. II, Brivanlou, I. H., Jordan, T. A. & Meister, M. Anticipation of moving stimuli by the retina. *Nature* **398**, 334–338 (1999).
8. Palmer, S. E., Marre, O., Berry, M. J. II & Bialek, W. Predictive information in a sensory population. *Proc. Natl Acad. Sci. USA* **112**, 6908–6913 (2015).

- 475 9. Hénaff, O. J. et al. Primary visual cortex straightens natural video trajectories. *Nat. Commun.*
476 **12**, 5982 (2021).
- 477 10. Shi, X. et al. Convolutional LSTM network: a machine learning approach for precipitation
478 nowcasting. In *Proc. Advances in Neural Information Processing Systems* **28**, 802–810
479 (NeurIPS, 2015).
- 480 11. Wang, Y., Long, M., Wang, J., Gao, Z. & Yu, P. S. PredRNN: recurrent neural networks for
481 predictive learning using spatiotemporal LSTMs. In *Proc. Advances in Neural Information*
482 *Processing Systems* **30**, 879–888 (NeurIPS, 2017).
- 483 12. Wang, Y., Gao, Z., Long, M., Wang, J. & Yu, P. S. PredRNN++: towards a resolution of the
484 deep-in-time dilemma in spatiotemporal predictive learning. In *Proc. 35th International*
485 *Conference on Machine Learning* **80**, 5123–5132 (PMLR, 2018).
- 486 13. Wang, Y., Jiang, L., Yang, M.-H., Li, L.-J., Long, M. & Fei-Fei, L. Eidetic 3D LSTM: a model
487 for video prediction and beyond. In *Proc. International Conference on Learning*
488 *Representations* (ICLR, 2019).
- 489 14. Wang, Y., Zhang, J., Zhu, H., Long, M., Wang, J. & Yu, P. S. Memory in memory: a predictive
490 neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In
491 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 9154–9162 (IEEE,
492 2019).
- 493 15. Gao, Z., Tan, C., Wu, L. & Li, S. Z. SimVP: simpler yet better video prediction. In *Proc.*
494 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* 3170–3180 (IEEE,
495 2022).
- 496 16. Tan, C. et al. Temporal attention unit: towards efficient spatiotemporal predictive learning. In
497 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 18770–18782
498 (IEEE, 2023).
- 499 17. Tan, C. et al. OpenSTL: a comprehensive benchmark of spatio-temporal predictive learning.
500 In *Proc. Advances in Neural Information Processing Systems Datasets and Benchmarks Track*
501 (NeurIPS, 2023).
- 502 18. Horowitz, M. Computing’s energy problem (and what we can do about it). In *Proc. 2014 IEEE*
503 *International Solid-State Circuits Conference* 10–14 (IEEE, 2014).

19. Wan, W. et al. A compute-in-memory chip based on resistive random-access memory. *Nature* **608**, 504–512 (2022).
20. Zhong, Y. et al. A memristor-based analogue reservoir computing system for real-time and power-efficient signal processing. *Nat. Electron.* **5**, 672–681 (2022).
21. Dang, B., Zhang, T., Wu, X. et al. Reconfigurable in-sensor processing based on a multi-phototransistor–one-memristor array. *Nat. Electron.* **7**, 991–1003 (2024).
22. Deng, M. et al. Memristive crossbar array-based hardware framework for compressed sensing and event-driven neuromorphic processing. *Nat. Commun.* **17**, 5717 (2026).
23. Wang, S., Zhao, J., Pu, T. et al. Ultrafast visual perception beyond human capabilities enabled by motion analysis using synaptic transistors. *Nat. Commun.* **17**, 1215 (2026).
24. Yu, Y. et al. Efficient and accurate neural-field reconstruction using resistive memory. *Nature* **654**, 642–651 (2026).
25. Aguirre, F. et al. Hardware implementation of memristor-based artificial neural networks. *Nat. Commun.* **15**, 1974 (2024).
26. Zhou, Y. et al. Computational event-driven vision sensors for in-sensor spiking neural networks. *Nat. Electron.* **6**, 870–884 (2023).
27. Gehrig, D. & Scaramuzza, D. Low-latency automotive vision with event cameras. *Nature* **629**, 1034–1040 (2024).
28. Mennel, L., Symonowicz, J., Wachter, S., Polyushkin, D. K., Molina-Mendoza, A. J. & Mueller, T. Ultrafast machine vision with 2D material neural network image sensors. *Nature* **579**, 62–66 (2020).
29. Guan, Y. et al. Photon-efficient camera with in-sensor computing. *Nat. Commun.* **16**, 3201 (2025).
30. Wang, Y., Jodoin, P.-M., Porikli, F., Konrad, J., Benezeth, Y. & Ishwar, P. CDnet 2014: an expanded change detection benchmark dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops* 393–400 (IEEE, 2014).
31. Alahi, A. et al. Social LSTM: human trajectory prediction in crowded spaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 961–971 (IEEE, 2016).

32. Salzmann, T., Ivanovic, B., Chakravarty, P. & Pavone, M. Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020* 683–700 (Springer, 2020).
33. Wojke, N., Bewley, A. & Paulus, D. Simple online and realtime tracking with a deep association metric. In *Proc. IEEE International Conference on Image Processing* 3645–3649 (IEEE, 2017).
34. Xie, X., Zhang, C., Zhu, Y., Wu, Y. N. & Zhu, S.-C. Congestion-aware multi-agent trajectory prediction for collision avoidance. In *Proc. IEEE International Conference on Robotics and Automation* 13693–13700 (IEEE, 2021).

ACKNOWLEDGEMENTS

The authors acknowledge the financial support from the National Natural Science Foundation of China (62671599, 62674245, 62605477, U23A20322, 62404253), Hunan Provincial Innovation Foundation for Postgraduate (XJJCB2026042), the science and technology innovation Program of Hunan Province (2026RC3256).

AUTHOR CONTRIBUTIONS

A.X., Y.N.W., Q.L. and Q.J.L. conceived the idea and designed the research. A.X., Y.N.W., L.Q., C.L.C. fabricated and characterized the memristor devices and arrays. A.X., Y.N.W., L.L., M.X.D. developed the mixed-signal memristive hardware platform and FPGA-based control system. A.X., Y.N.W., Q.J.L. developed the spatiotemporal compression, projection-current readout and motion-prediction methods. A.X., Y.N.W., L.L., Y.P.W., C.L., C.L.C., X.F., H.Y.X., B.H.D., H.X., Q.L. and Q.J.L. analysed and interpreted the results. A.X., Y.N.W., Q.L. and Q.J.L. cowrote the manuscript. Y.N.W., Q.L. and Q.J.L. supervised the research. All authors discussed the results and commented on the manuscript. These authors contributed equally: Ao Xu, Lun Lu.

COMPETING INTERESTS

The authors declare no competing interests.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [sivideo2.zip](#)
- [si.pdf](#)
- [sivideo1.zip](#)