Molecular convolutional neural networks with DNA regulatory circuits

Hao Pei (peihao@chem.ecnu.edu.cn)
East China Normal University
Xiewei Xiong
East China Normal University
Tong Zhu
East China Normal University
Yun Zhu
East China Normal University
Mengyao Cao
East China Normal University
Jin Xiao
East China Normal University
Li Li
East China Normal University https://orcid.org/0000-0001-8494-4997
Fei Wang
Shanghai Jiao Tong University
Chunhai Fan
Shanghai Jiao Tong University https://orcid.org/0000-0002-7171-7338

Article

Keywords: ConvNet, DNA switching gate architecture, parallel multiply-accumulate operations

Posted Date: October 21st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-926838/v1

Version of Record: A version of this preprint was published at Nature Machine Intelligence on July 4th, 2022. See the published version at https://doi.org/10.1038/s42256-022-00502-7.

Abstract

Complex biomolecular circuits enable cells with intelligent behavior for survival before neural brains evolved. Synthesized DNA circuits in liquid phase developed as computational hardware can perform neural-network-like computation that harness the collective properties of complex biochemical systems, however the scaling up in complexity remains challenging to support more powerful computation. we present a systematic molecular implementation of the convolutional neural network (ConvNet) algorithm with synthetic DNA regulatory circuits based on a simple DNA switching gate architecture. We experimentally demonstrated that a DNA-based ConvNet based on shared-weight architecture of a 3×6 sized kernel can simultaneously implement parallel multiply-accumulate (MAC) operations for 144 bits inputs and recognize patterns up to 8 categories autonomously. Furthermore, it can connect with another DNA circuits to construct hierarchical networks, which can recognize patterns up to 32 categories with a two-step classification approach of performing coarse classification on language (Arabic numerals, Chinese oracles, English alphabets and Greek alphabets) and then classifying them into specific handwritten symbols. With a simple cyclic freeze/thaw approach, we can decrease computation time from hours to minutes. Our approach shows great promise in the realization of high computing power molecular computer with ability to classify complex and noisy information.

Introduction

Categorization, a fundamental element of thinking, is an important mechanism for rapid information processing with neural circuits in the brain[1–3]. DNA components based artificial neural circuits[4–8] developed to mimic such categorization function can classify molecular inputs into discrete patterns, however it remains challenging to scale up in complexity to support more powerful computational systems. A pioneering study in this area from Qian and Winfree *et. al.* demonstrated a network of interacting DNA strands that can act as artificial neurons and remembers 4 molecular patterns by implementing Hopfield neural network strategy[5]. Then, Qian *et. al.* scaled up the molecular pattern recognition of DNA neural networks that enable recognition of 9 molecular patterns by implementing more powerful winner-take-all neural networks strategy with DNA strands[7]. These results indicate that the sophistication and elegance of algorithms implemented experimentally with reactive orthogonal DNA molecules play a key role on determining computation efficiency of DNA neural network circuitry, which is analogous to electronic circuitry [9,10].

Convolutional neural network (ConvNet) is a powerful computational model for categorization in deep learning[11–16], in which the connection pattern between neurons resembles the organization of animal visual cortex[17]. They are known as translation-invariant artificial neural networks, which based on shared-weight structure of the convolution kernels that slide along input features and subsequently provide translation equivariant output as feature maps. Compared to fully-connected networks resulting in overly-complex network structures, ConvNet is on the lower extreme on a scale of connectivity and complexity, since it features a sparse topology to effectively reduce network connections and weight parameters[13,14], thus holding great promise in allowing simpler molecular implementation at vastly smaller scale. Although ConvNet algorithms have been proposed enabling efficient hardware implementation in electronic computing devices as well as in photonic and quantum computing devices[12, 18–23], it has not yet been demonstrated in molecular computing systems.

Here we present a systematic strategy for computationally designed networks of reactive orthogonal DNA molecules that is capable of implementing the convolutional neural network (ConvNet) algorithm. We show that the DNA-based ConvNet can simultaneously implement multiple parallel multiply-accumulate (MAC) operations and recognize patterns up to 8 categories. Each pattern comprises of up to 43 distinct DNA strands selected tracing individual handwritten symbols, which is chosen from the set of 144 that represents the 144 bits in 12×12 patterns. By connecting the upstream logic circuits that activates a specific set of weight molecules, the DNA-based ConvNet can recognize patterns up to 32 categories with a two-step classification approach of performing coarse classification on language (Arabic numerals, Chinese oracles, English alphabets and Greek alphabets) and then classifying into specific handwritten symbols. Moreover, we show that a simple cyclic freeze/thaw approach can significantly accelerate large-scale DNA neural network reactions, which decreases computation time from hours to minutes. Our approach leads towards the realization of high computing power molecular computer with ability to classify complex and noisy information.

Results

A DNA switching gate architecture design used for ConvNet circuit. A ConvNet basically consists of input layer, convolutional layer, nonlinear layer, and output layer. In each layer, an intermediate array of pixels, referred to as feature map, is produced from the previous layer. Fig. 1a illustrates the operation principle of the ConvNet for recognition tasks, where a n × n input symbol is convolved with a k × k kernel function (stride=1) to compute a feature map of dimensions (n-k+1) × (n-k+1). When operating ConvNet, the input symbol is grouped into $(n-k+1)^2$ receptive regions (blue dashed line marked area of Fig. 1a) of dimensions k × k. The elements of these receptive regions share the same weights, which could enable a sparse topology to effectively reduce network connections. Mathematically, a convolution operation requires multiple MAC operations ($y=\sum w_i \times x_i$) with shared weights. To implement the weight-sharing MAC operation using DNA molecules, we proposed a switching gate architecture[24]. Each switching gate is associated with a gate base strand (for example, domains *T*Ssi*Si*T** in Fig. 1b) that has a recognition domain (*Si** in Fig. 1b) and a weight tuning domain (*Ssi** in Fig. 1b) flanked with two toehold domains (*T** in Fig. 1b), and these domains are functionally independent. Varying the sequence of the recognition domain at 3'end (or 5'end) would enable it to be connected to different downstream gates (or upstream gates), leading to different signal transmission pathways. Varying the sequence of the weight tuning domain would enable to determine the weights assigned to the input. The cascaded circuit could thus allow independent control of signal transmission functions and weight assignment functions during computation. In this way, we can implement weight-sharing MAC operations at the molecular level and construct molecular convolutional neural networks with reactive orthogonal DNA molecules.

DNA implementation of MAC and convolution operation. We started experimental demonstration with weight multiplication function ($y=w_i \times x_i$), in which x_i is a binary input and w_i is an analogue weight. These weights are implemented by designing the switching gate with one weight tuning domain and two recognition domains.Weight multiplication is implemented with cascaded reactions (*Supplementary Fig.3a*) wherein input species X_i convert an activated weight substrate molecule $N_{i,i,j}^{*}$ to an intermediate product $P_{i,j}$. $N_{i,i,j}$* is implemented with reactions that $N_{i,i,j}$undergoes a spontaneous intramolecular-conformational-switch upon hybridization with weight tuning molecule W_i. In the absence of X_i, no $P_{i,j}$ will be generated; in the presence of X_i, then the final concentration of $P_{i,j}$ will be determined by the concentration of $N_{i,i,j}^{*}$, thus setting the value of the weighted multiplication y. Different weights can be implemented by varying the concentration of W_i (*Supplementary Fig.3b, and Supplementary Figs.4-10*). Then, we can compute the sum of weighted inputs within the same neuron ($y=\sum x_i$). This is implemented with reactions wherein all intermediate species $P_{i,j}$ stoichiometrically convert summation gate ($Sd_{j,k}$) to common weighted-sum species $Ss_{j,k}$ (*Supplementary Fig.3c,d*). It should be noted that weights with negative values are implemented by different DNA strands. To complete the summation, the positive weighted-sum species $Ss_{j,k}$ and negative weighted-sum species $Ss_{i,n}$ need to be subtracted from one another (*Supplementary Fig.3e*). Specifically, all positive weighted-sum species $Ss_{j,k}$ can convert the double-stranded complex $Dd_{k,m}$ to an intermediate species $Ds_{k,m}$. All negative weighted-sum species $Ss_{i,n}$ generated from the previous step can bind to the toehold of inhibitory strand In_i and branch-migrate to form inert waste species, producing reactive annihilation species $Sub_{i,n}$* through intramolecular-conformational-switch.The subtraction can thus be realized wherein $Sub_{i,n}$* and $Ds_{k,m}$ annihilate each other. Only remaining $Ds_{k,m}$ will interact with the downstream reporting gate (*Supplementary Fig.3f*) to read the output signal; otherwise reaction is terminated (*Supplementary Fig.3g*).

Subsequently, we demonstrated that simple MAC operations can be implemented by combining weight multiplication and summation subfunctions (*Supplementary Figs. 11 and 12*). For example, a two-species MAC operation ($y=w_1 \times x_1 + w_2 \times x_2$) is implemented by adding one summation gate to two parallel weight multiplications (*Supplementary Fig. 11a, b*). We can vary the concentration of respective weight tuning molecules (W_1 and W_2) to obtain different weights for corresponding multiplication reaction, then we can compute the sum of all weighted inputs by using the same recognition domain connected to summation gate. As expected, the circuit exhibits a stoichiometric behaviour with the output level and the concentration of W_i (*Supplementary Fig. 11c, d*).

Then, we combined two MAC operations to demonstrate convolution of a 2×2 input pattern using a 2×1 kernel. Each receptive region of input patterns (x_1 and x_3; x_2 and x_4) multiply with weights (w_1 and w_2) to obtain weighted pixels ($x_1 \times w_1$ and $x_3 \times w_2$; $x_2 \times w_1$ and $x_4 \times w_2$), respectively. Feature maps (y_1 and y_2) are then exported by summing up the weighted pixels in the same receptive region (Fig. 2a). We built a small-scale DNA regulatory circuit using three subfunctions: multiplication, summation, and reporting, to perform the convolution operation (Fig. 2b). The 2×2 input pattern is encoded with four DNA input strands (X_1, X_2, X_3, and X_4). The convolution kernel is encoded in the sequence of weight tuning domains (see green domains *Ss1* and *Ss2* in Fig. 2b) of weight tuning molecules (W_1 and W_2). To complete the multiplication with shared weights, we designed four weight substrate molecules ($N_{1,1,4}$ and $N_{1,2,6}$, $N_{2,3,4}$ and $N_{2,4,6}$), and two of which (for example, $N_{1,1,4}$ and $N_{1,2,6}$ in Fig. 2b) have the same weight tuning domain corresponding to the pixels that interacts with the same kernel in different local receptive regions, but have different recognition domain at 3' end to connect to the downstream summation gates ($Sd_{4,5}$ and $Sd_{6,7}$). Each input patterns were binary patterns, in which 1 or 0 represent the presence or absence of input strands. The value of analogue weights determined from convolution kernel is implemented with the concentrations of $N_{i,i,j}$. To compute convolution, each DNA sub-circuit runs independently and parallelly to compute MAC operation in each receptive region. For a specific pattern, the corresponding weight tuning molecules W_i would activate respective weight substrate molecules $N_{i,i,j}$. Thus the assignment of shared weights by the convolution kernel is implemented with the activated weight substrate molecule $N_{i,i,j}$* and X_i through DNA strand displacement reaction, resulting in releasing of intermediate species $P_{i,j}$. Two summation gates $Sd_{j,k}$ converts $P_{i,j}$ in the same receptive regions to weighted-sum species $Ss_{j,k}$, leading to the trigger of downstream reporting gates. For experimental demonstration, we chose 6 input patterns and all two outputs achieved their correct 'on' or 'off' states, indicating that the DNA circuit correctly implemented the convolution computation (Fig. 2c). For example, with inputs X_1 X_2 X_3 X_4 = 1001, the concentration of output strands and corresponding fluorescence signal y_1 (or y_2) is proportional to $X_1 \times W_1$ (or $X_4 \times W_2$) as designed.

A DNA-based ConvNet for molecular pattern recognition. Having shown that the DNA circuit is capable of processing the convolutions, we next built a DNA-based ConvNet that can 'remember' two handwritten symbols: Chinese oracles 'fire' and 'earth' (Fig. 3a). The training set consists of 48,000 patterns of handwritten symbols from the Sinica oracle database. In silico, all original symbols were converted to 144-bit binary patterns for network training by rescaling them to 12×12 grids, and setting each pixel to 1 when exceeding the threshold (*Supplementary Fig. 13*). The convolution kernel (a 6×6 matrix) slides along input patterns with a stride of 6 and subsequently generates a corresponding output feature map (Fig. 3b and *Methods* '*Neural-network training and testing*'). We evaluated the network performance on a reference dataset after training, reaching a 97% accuracy (Fig. 3c). We implemented this ConvNet model by encoding the convolution kernel in the sequence of weight tuning domain, and implementing the value of weights with the concentration of weight substrate molecule $N_{i,i,j}$. The test input binary patterns were encoded with single strands, wherein each 1 or 0 represents the presence or absence of input strand (Fig. 3d). A DNA-based ConvNet implements pattern recognition by comparing its local feature to all memories and identifying the most similar memory (*Supplementary Fig. 14*). For example, each receptive region of a 'fire' can simultaneously interact with same kernel function to export feature maps through DNA strand displacement cascades. As the network runs, a subset of weight tuning molecules W_i could activate corresponding weight substrate molecule $N_{i,i,j}$ in four receptive region at the same time to enable multiple weight-sharing MAC operations to perform parallelly (*Supplementary Fig. 15a*). This allows DNA circuits to be able to activate a specific reaction pathway in the convolution layer when exposed to a specific pattern, which can enhance network robustness (*Supplementary Fig. 16*). Then, max-pooling process is applied to reduce feature map size by annihilating the smaller one between two pixels through cooperative hybridization[7] (*Supplementary Fig. 15b*). As shown in the experimental data (Fig. 3e and *Supplementary Fig. 17*), the input patterns of two handwritten symbols each triggered desired outputs, indicating that two handwritten symbols are classifiable. When each oracle pattern was rotated with the angle increment of 30° from 0° to 360°, the circuit still yielded the desired output for all 26 test input patterns, indicating that the circuit correctly classified rotated patterns. In total, 177~250 distinct molecules were used for all test patterns (*Supplementary Fig. 17b*). As expected, we showed that the DNA-based ConvNet can also remember eight 144-bit molecular patterns simultaneously (*Supplementary Figs. 18-22*).

A hierarchical neural network for 32 patterns recognition. Our ConvNet has the feature that inputting of weight tuning molecules can selectively activate specific set of weights to allow the same set of DNA molecules to be used for different tasks. The use of weight tuning molecules as outputs of the upstream circuit shows up possibilities for building hierarchical networks for more sophisticated categorization tasks. To validate this approach, we proposed a two-step classification approach that first uses a logic gate to perform coarse classification and then uses a ConvNet to perform finer classification. To demonstrate this approach experimentally, we choose a task of recognizing 32 handwritten symbols that can be divided into 4 groups: including 8 Chinese oracles (Sinica oracle database), 8 Arabic numerals (MNIST database), 8 English alphabets and 8 Greek alphabets (Kaggle website). In silico, we converted all original handwritten symbols to binary patterns with two layers (Fig. 4a). Layer 1 is on a 1×4 grid and acts as an input for logic circuits to perform coarse classification on languages (for example, oracle is 1000), yielding the outputs to selectively activate the downstream ConvNet subnetwork to perform fine classification into specific handwritten symbols using Layer 2 on a 12·12 grid as inputs. 4 groups in Layer 2 can be separately trained in silico with respective datasets to obtain the optimal model (Fig. 4b), thus yielding values of four convolution kernel with dimension of 3×6 (stride = 3×6) (*Supplementary Fig. 23, Methods 'Neural-network training and testing'*). This network performed well in the reference dataset, reaching >84.0% accuracy in each group (Fig. 4c). We implemented two-step classification approach experimentally by designing different switching gates to encode four convolution kernels. Both the tags in Layer 1 and the inputs in Layer 2 are binary patterns, in which each 1 or 0 indicates the presence or absence of a tag strand (or an input strand), respectively (*Supplementary Fig. 24a*). The pattern classification can be completed with following steps (Fig. 4d): (i) The tag strand in Layer 1 will react with the reporter gate to generate fluorescence signal y^i, which can be recognized as corresponding coarse category (*Supplementary Fig. 24b, c*). Meanwhile, the tag strand will react with the fan-out gate to release a set of weight tuning molecules W_i, which can then activate the downstream DNA neural networks (*Supplementary Fig. 24b, d*). (ii) The W_i generated from upstream logic circuits (*Supplementary Fig. 25*) then activates corresponding neural network to implement the recognition (Fig. 4e), in which each output y_i is uniquely correlated to specific handwritten symbols to enable the fine classification. Two fluorescence signals are collected from Layer 1 and Layer 2 respectively to determine the recognition results. For example, a 'fire' is recognized if and only if y^1=1 and y_1=1 (where y^1 is the output identifying the coarse category of ‘oracle’ and y_1 is the output identifying the fine category of ‘fire’). In total, constructing the DNA-based ConvNet that can remember 32 molecular patterns requires 368~512 distinct molecules for all test patterns. As expected, the circuit yielded the desired pair of outputs for 32 representative example patterns with group identities (Fig. 4f, *Supplementary Figs. 26 and 27*). In general, with this hierarchical approach, constructing a DNA-based ConvNet that can recognize b×m distinct n-bit patterns (b is number of groups and m is number of patterns in each group) with e-bit kernel size requires n+5m+(m+1)×b×e molecules (*Supplementary Fig. 28*).

A cyclic freeze/thaw approach accelerate DNA circuits. The speed of execution of DNA computing remains a challenge in large-scale DNA neural network reactions. For example, our computation of 2 categories took longer than 20 h (*Supplementary Fig. 17c*), and computation increased to over 36 h for 32 categories (*Supplementary Fig. 27*). To accelerate DNA circuits, we developed a simple cyclic freeze/thaw approach (Fig. 5a). The cyclic freeze/thaw approach iteratively drives the strand displacement reaction towards thermodynamic equilibrium, which can accelerate the basic strand displacement reaction by ~75-fold (*Supplementary Figs. 29 and 30*). For a larger-scale circuit, 160 test patterns of 144 bits can be recognized with less than 30 min through 5 freeze/thaw cycles, which would otherwise require hours. (Fig. 5b, c).

Conclusion

We have experimentally demonstrated a DNA-based ConvNet that can robustly accomplish information categorization function, holding great promise for further scaling up molecular computation. With a DNA switching gate architecture that allows independent control of signal transmission functions and weight assignment functions during computation, we can implement weight-sharing MAC operations at the molecular level and construct molecular convolutional neural networks with reactive orthogonal DNA molecules. The massive parallelism feature inherent to DNA molecules could enable parallelizing convolution operations autonomously, which are particularly well suited for more scalable information processing. With a hierarchical network that first uses a logic gate to perform coarse classification and then uses a ConvNet to perform finer classification, we have demonstrated that the circuits can be scaled up to classify patterns into 32 categories. This could also provide the possibility of integrating multiple circuit architecture[7,25] to enhance computational power. Importantly, we have extended the key feature of ConvNet-sparse topology-to a DNA neural network, effectively reducing complexity of network architecture through sparsely connected neurons, which could allow more complex information processing and potentially provide the molecular circuits with ‘intelligent’ behavior that resembles a biological neural network. Furthermore, by interfacing non-nucleic-acid sensory inputs[26–33], the DNA-based ConvNet can be adapted for a wide variety of molecular pattern classification tasks[34–38], which have potential applications including disease diagnostics, profiling expression patterns, and precision medicine.

Methods

Sequence design. Five types of molecular structures (*Supplementary Fig. 1*) were used in this work: (1) Weight substrate molecules $N_{i,i,j}$ and annihilation species $Sub_{i,n}$ consist of three single strands; (2) Summation gates $Sd_{j,k}$ and complexes $Dd_{k,m}$ consist of two single strands; (3) Reporters Rep_m consist of single strands modified with uorophore and quench groups. All DNA single strands used in this work were composed of long recognition domains and short toehold domains, except for the weight tuning domains used for weight multiplication, subtraction and reporting, which were composed of short stem domains and long loop domains. Note that these domains are functionally independent. On this basis, the sequence design was conducted at the domain level.

We have generated several pools of domain sequences with different lengths according to a series of design heuristics[39,40]. To reduce secondary structures and undesired interactions, all domain sequences were produced by using a three-letter code (A, C, and T). To reduce synthesis errors, no more than 4 A's or T's, and no more than 3 C's were used in a row; 30–70% C-content was kept to ensure comparable melting temperatures. For any two sequences in the pool, the longest length of matching sequence was no more than 35% of the domain length, and all sequences were formed by at least 30% different nucleotides. Sequences of single strands were generated by directly concatenating these domains together.

Finally, a 15-nucleotide sequence pool used for recognition domains and a 27-nucleotide sequence pool used for weight tuning domains were generated. We have checked the two sequence pools to ensure the same pairwise criteria. To reduce the gate-gate leak[40], we have used two-nucleotide clamps in all bottom strands, which are complementary to the first two-nucleotide in the tail of the molecular species. We have used three universal toeholds for all DNA strands expect for the DNA circuits used for 32 patterns recognition, in which the branch migration cannot be initiated by toehold domains without matching recognition domains. $A_{i,i}$ had 6-nucleotide toeholds, which was composed of the 5-nucleotide universal toehold and 1-nucleotide extension G, and was used in the weight multiplication layers to ensure the effective strand displacement reaction rate. $Sub_{i,n}$ had 7-nucleotide toeholds, which was composed of a 5-nucleotide universal toehold and a 2-nucleotide extension that is complementary to the 2-nucleotide next to the toehold of the upstream complexes. All weight tuning molecule W_i shared a 7-nucleotide universal toehold domain. All the other molecular complexes shared a 5-nucleotide universal toehold domain. To reduce the side reaction caused by the universal toehold binding in the DNA circuits used for 32 patterns recognition, the toehold of W_i was different for switching different convolution kernels. Two nucleotides "TT" were inserted between the toehold domains and recognition domains in input strands X_i and single strand $Ds_{k,m}$ to ensure the effective strand displacement reaction rate.

Designed DNA strands were verified by NUPACK[41] to ensure the binding energy and specicity. Note that bottom strands of complexes in the network are complementary to corresponding domains and thus contain A, G and T. We also validated the correct formation of the hairpin loop structures in the presence of W_i by NUPACK.

All DNA sequences used in this experiment are listed in *Supplementary Table*.

Neural network training and testing. The convolutional neural networks (ConvNets) were trained to recognize four categories of symbols: handwritten Chinese oracles ('▯(fire)', '▯(earth)', '▯(tree)', '▯(water)', '▯(gas)', '▯(sky)', '▯(human)', and '▯(life)'), handwritten English alphabets ('A' to 'H'), handwritten Greek alphabets ('α' to 'θ'), and handwritten Arabic numerals ('1' to '8'). 112 images of Chinese oracles were obtained from the Sinica oracle database (http://xiaoxue.iis.sinica.edu.tw/jiaguwen*).* 11,164 images of handwritten English alphabets and 112 images of handwritten Greek alphabets were obtained from the Kaggle website (https://www.kaggle.com/*).* 16,000 images of handwritten Arabic numerals were obtained from the MNIST dataset (http://yann.lecun.com/exdb/mnist/*).*

To ensure the reliability of the ConvNet model and avoid overfitting, one must ensure that sufficient data is available in the training/test dataset. However, the number of images for the Chinese oracles and handwritten Greek alphabets is far from sufficient. Thus, the Augmentor software[42] was used to augment the dataset. The expanded four datasets had 1,000 different images for each character. A separate dataset was constructed for Chinese oracle 'fire' and 'earth'. Initially, there were 1,000 images for each oracle. Then, each image was rotated 24 times with 15 degrees per time. Finally, the dataset contains 48,000 images. For each recognition task, 80.0% of datasets were put in the training set while the rest were put in the validation set.

Each original handwritten symbol was rescaled as a grayscale image with 60·60 pixels by the Pillow software with DOI [10.5281/zenodo.5394534]. The pixel values in each image that exceeded the threshold were set to 1, and the rest were set to 0. The value of threshold can be adjusted in specific circumstances. In 32 handwritten symbols recognition experiments, each input symbol was replaced by a pair of input symbols (Layer 1 and Layer 2 in Fig. 4a). Binary tags were attached to Layer 1 to mark coarse categories (1000 corresponds to Chinese oracles, 0100 corresponds to Arabic numerals, 0010 corresponds to English alphabets, and 0001 corresponds to Greek alphabets). Layer 2 was kept as 12·12 binary pattern to be classified at the finer level.

Identifying rotated handwritten symbols is more demanding on the ConvNet model. Here we tested the ability of the ConvNet to recognize rotated symbols of two Chinese oracles. First, the kernel size and the stride were set as 6·6 and 6. After the convolution operation, we could obtain an output feature map with shape 2·2. The following step was the operation of the ReLU (Rectified Linear Unit) activation function. The max-pooling process reduced the 2·2 matrix to 1·2 through setting the pooling size and pool stride as 2·1 and 1. Eventually, this model's accuracy can reach 96.8%, and the recognition accuracy can reach 97.0%.

For eight handwritten symbols classification, the inputs of symbols were convolved by a 3·6 sized kernel. The stride was set to 3·6, then the size of the first feature map should be 4·2. The ReLU activation function was then used to zero out any value less than zero in the feature map. After that, the 4·2 sized feature map was flattened to a 1·8 sized matrices, which was finally activated by a softmax activation function. After sufficient epochs, all four models have achieved or exceeded an accuracy of 0.9 on both raining and validation sets. In addition, the performance of models on training and validation sets was very similar, indicating no overfitting. The recognition accuracy of the model for each image all exceeded 85.0%.

The training process was performed on the Keras platform (https://keras.io*).* During the model compilation process, the Adam[43] optimizer was used to compute the gradient. The learning rate was set to 0.001 and the exponential decay rate β1 and β2 were set to 0.9 and 0.999, respectively. The constant epsilon was set to 10^{-8} and the decay value of the learning rate was set to 0 after iteration. The sparse categorical cross-entropy was chosen as the loss function and sparse categorical accuracy function was chosen as metrics. The batch size was set to 150 and the number of training epochs was set to 1000. All of parameters were selected after a series of comparisons and tests and the final values were chosen because they can balance the network size and predictive power.

DNA oligonucleotide synthesis. With design, ULTRAPAGE purified oligonucleotides and HPLC purified oligonucleotides modified with fluorophores were provided by Sangon Biotech and were used without further purification. All strands were shipped lyophilized and resuspended at 200 μM in 1× TAE buffer with 12.5 mM Mg^{2+}, pH 8.0, and stored at 4°C for further use.

Annealing protocol and buffer condition. All triplexes were prepared for annealing at 50 μM with top and bottom strands in a 1:1:1 ratio, and all duplexes were prepared for annealing at 50 μM with top and bottom strands in a 1:1 ratio, while reporters were prepared at 50 μM with top quencher strands in 20% excess of bottom strands. The buffer for all experiments and annealed complexes was 1× TAE with 12.5 mM Mg^{2+} (pH 8.0). Complexes were annealed in a thermal cycler (Life Technologies) by heating to 95°C for 5 min and then cooling to 20°C at a rate of 0.1°C per 8 s, then were kept at 4 ℃. The hybridized molecules were purified by 12% polyacrylamide gel electrophoresis (PAGE).

Fluorescence spectroscopy. Kinetics experiments were performed with a spectrofluorometer (Fluorolog-max, Horiba) at 25 ℃. The instrument allows running four experiments in parallel. Fluorescence kinetics data were collected every 30 s, 60 s, or 10 min, depending on the overall experimental time length. A maximum four excitation / emission pairs in one experiment can be measured in parallel. Excitation (emission) wavelengths were 495 nm (520 nm) for dye FAM, 530 nm (563 nm) for dye HEX, 585 nm (605 nm) for dye ROX, and 685 nm (705 nm) for dye Cy5.5 (*Supplementary Fig. 2*). For circuits using only one reporter, FAM was used; for circuits using two reporters, FAM and ROX were used. Before experiment, all cuvettes were successively washed with distilled water for 6 times, 70% ethanol for 1 time, and distilled water for 5 times. In the 8 patterns recognition and 32 patterns recognition, 8 output trajectories were recorded using four distinct fluorophores. Every experiment was conducted twice, each containing half of outputs correlated to fluorophore-labelled reporters and the other half correlated to non-fluorophore-labelled reporters. We can observe all 8 outputs simultaneously by combining all output trajectories in one plot.

Fluorescence data normalization. All raw fluorescence data were normalized to standard concentration of output signals, and the difference in fluorescence readout caused by the instrument is neglectable. We conducted each set of parallel experiments for the same circuit with different inputs, and these experiments were normalized together for data analysis. For a given fluorophore, in the sets of parallel experiments where at least one of output signals that increased high and reached plateau at the end of the experiment, the maximum level (output=1) was determined by the highest completion signal. In the sets of parallel experiments where none of output signals increased all the way to completion, the standard concentration (1×) of output fluorescence level was obtained from the highest signal produced from the reporter Rep_m at the same time.

Declarations

Code availability. The algorithm code used for network training of this work is available in the Code Ocean (and https://github.com/tongzhugroup/DNAcode), or from the corresponding author upon reasonable request.

Data availability. The associated data and experimental protocols for this work are available within Supplementary Information, or from the corresponding author upon reasonable request. **Supplementary Information** is available in the online version of the paper.

Acknowledgements This work was supported by the National Science Foundation of China (Grant No. 21722502, 22074041), National Key Research and Development Program of China for International Science & Innovation Cooperation Major Project between Governments (2018YFE0113200), Shanghai Rising-Star Program (19QA1403000).

Author Contributions H.P. conceived and supervised the research. X.X. designed and perform the experiments. H.P., T.Z., and X.X. discussed the design. Y.Z. and M.C. carried out experiments and interpreted data. J.X. and T.Z. developed the model and performed the *in silico* training. All authors analyzed data. X.X., L.L., F.W., C.F. and H.P. wrote the manuscript.

References

1. Freedman, D. J. & Assad, J. A. Experience-dependent representation of visual categories in parietal cortex. *Nature* 443, 85–88 (2006).
2. Zhong, L. et al. Causal contributions of parietal cortex to perceptual decision-making during stimulus categorization. *Nat. Neurosci.* **22**, 963–973 (2019).
3. Reinert, S., M Hübener, Bonhoeffer, T., & Goltstein, P. M. Mouse prefrontal cortex represents learned rules for categorization. *Nature* **593**, 411–417 (2021).
4. Kim, J., Hopfeld, J. & Winfree, E. Neural network computation by in vitro transcriptional circuits. Adv. *Neural Inf. Process. Syst.* **17**, 681–688 (2005)
5. Qian, L., Winfree, E., & Bruck, J. Neural network computation with DNA strand displacement cascades. *Nature* **475**, 368–372 (2011).

6. Genot, A. J., Fujii, T., & Rondelez, Y. Scaling down DNA circuits with competitive neural networks. *J. R. Soc. Interface* **10**, 20130212 (2013).
7. Cherry, K. M., & Qian, L. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* **559**, 370–376 (2018).
8. Kim, S. et al. Nanoparticle-based computing architecture for nanoparticle neural networks. *Sci. Adv.* **2**, eabb3348 (2020).
9. Soltoggio, A., Stanley, K. O. & Risi, S. Born to learn: the inspiration, progress, and future of evolved plastic artifcial neural networks. *Neural Netw.* **108**, 48–67 (2018).
10. Stanley, K. O., Clune, J., Lehman, J., & Miikkulainen, R. Designing neural networks through neuroevolution. *Nat. Mach. Intell.* **1**, 24–35 (2019).
11. Lecun, Y., Boser, B., Denker, J. S., Henderson, D., & Hubbard, W. Backpropagation applied to handwritten zip code. *Neural Comput.* **1**, 541–551 (1989).
12. Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems* **25**, 1090–1098 (2012).
13. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
14. Al-Saffar, A., Hai, T., & Talab, M. A. Review of deep convolution neural network in image classification. 2017 *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET) IEEE* 26-31 (2017).
15. Luo, R., Sedlazeck, F. J., Lam, T. W., & Schatz, M. C. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. *Nat. Commun.* **10**, 998 (2019).
16. Sahraeian, Sme. et al. Deep convolutional neural networks for accurate somatic mutation detection. *Nat. Commun.*<bvertical-align:super;> </bvertical-align:super;><background-color:#FFCC66;bvertical-align:super;>10</background-color:#FFCC66;bvertical-align:super;>, 1041 (2019).
17. Hubel, D. H., & Wiesel, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **160**, 106–154 (1962).
18. Yao, P. et al. Fully hardware-implemented memristor convolutional neural network. *Nature* **577**, 641–646 (2020).
19. Cong, I., Choi, S., & MD Lukin. Quantum convolutional neural networks. *Nat. Phys.* **15**, 1273–1278 (2019).
20. Xu, X. et al. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* **589**, 44–51 (2021).
21. Wang, Y., H Tang, Xie, Y., Chen, X., & Bao, W. An in-memory computing architecture based on two-dimensional semiconductors for multiply-accumulate operations. *Nat. Commun.* **12**, 3347 (2021).
22. Feldmann, J. et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589**, 52–58 (2021).
23. Wu, C., Yu, H., Lee, S., Peng, R., & Li, M. Programmable phase-change metasurfaces on waveguides for multimode photonic convolutional neural network. *Nat. Commun.* **12**, 96 (2021).

24. Lai, W. et al. Programming chemical reaction networks using intramolecular conformational motions of DNA. *ACS. Nano.* **12**, 7093–7099 (2018).
25. Wilhelm, Daniel, Bruck, Jehoshua, Qian, & Lulu. Probabilistic switching circuits in DNA. *Proc. Natl Acad. Sci. USA* **115**, 903–908 (2018).
26. Seelig, G., Soloveichik, D., Zhang, D., & Winfree, E. Enzyme-free nucleic acid logic circuits. *Science* **314**, 1585–1588 (2006).
27. Kunihiko, et al. Small molecule release and activation through DNA computing. *J. Am. Chem. Soc.* **139**, 13909–13915 (2017).
28. Bertucci, A., Porchetta, A., Grosso, E. D., Patio, T., & Ricci, F. Protein-controlled actuation of dynamic nucleic acid networks using synthetic DNA translators. *Angew. Chem. Int. Ed.* **59**, 20577–20581 (2020).
29. Zhou, J., & Rossi, J. Aptamers as targeted therapeutics: current potential and challenges. *Nat. Rev. Drug. Discov.* **16**, 181–202 (2017).
30. Xiao, M., Lai, W., Wang, F., Li, L., & Pei, H. Programming drug delivery kinetics for active burst release with DNA toehold switches. *J. Am. Chem. Soc.* **141**, 20354–20364 (2019).
31. Xiong, X. et al. Optochemical control of DNA switching circuits for logic and probabilistic computation. *Angew. Chem. Int. Ed.* **60**, 3397–3401 (2021).
32. Tang, Q. et al. Multi-mode reconfigurable DNA-based chemical reaction circuits for soft matter computing and control. *Angew. Chem. Int. Ed.* **60**, 15013–15019 (2021).
33. Xiao, M., Lai, W., Yu, H., Yu, Z., & Pei, H. Assembly pathway selection with DNA reaction circuits for programming multiple cell−cell interactions. *J. Am. Chem. Soc.* **143**, 3448–3454 (2021).
34. Lopez, Randolph, Wang, Ruofan, Seelig, & Georg. A molecular multi-gene classifier for disease diagnostics. *Nature Chem.* **10**, 746–754 (2018).
35. Zhang, C., Zhao, Y., Xu, X., Xu, R., & D Han. Cancer diagnosis with dna molecular computation. *Nat. Nanotechnol.* **15**, 709–715 (2020).
36. Xiao, M., Lai, W., Man, T., Chang, B., & Pei, H. Rationally engineered nucleic acid architectures for biosensing applications *Chem. Rev.* **119**, 11631–11717 (2019).
37. Douglas, S. M., Bachelet, I. & Church, G. M. A logic-gated nanorobot for targeted transport of molecular payloads. *Science* **335**, 831–834 (2012).
38. Benenson, Y., Gil, B., Ben-Dor, U., Adar, R. & Shapiro, E. An autonomous molecular computer for logical control of gene expression. *Nature* **429**, 423–429 (2004).
39. Thubagere, A. J., Thachuk, C., Berleant, J., Johnson, R. F., & Qian, L. Compiler-aided systematic construction of large-scale dna strand displacement circuits using unpurified components. *Nat. Commun.* **8**, 14373 (2017).
40. Qian, L. & Winfree, E. Scaling up digital circuit computation with DNA strand displacement cascades. *Science* **332**, 1196–1201 (2011).

41. Zadeh, J. N., et al. NUPACK: Analysis and design of nucleic acid systems. *J. Theor. Comput. Chem.* **32**, 170–173 (2011).

42. MD Bloice, Roth, P. M., & Andreas, H. Biomedical image augmentation using augmentor. *Bioinformatics* **35**, 4522–4524 (2019).

43. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *Proc. International Conference on Learning Representations* (ICLR, 2015).

Figures

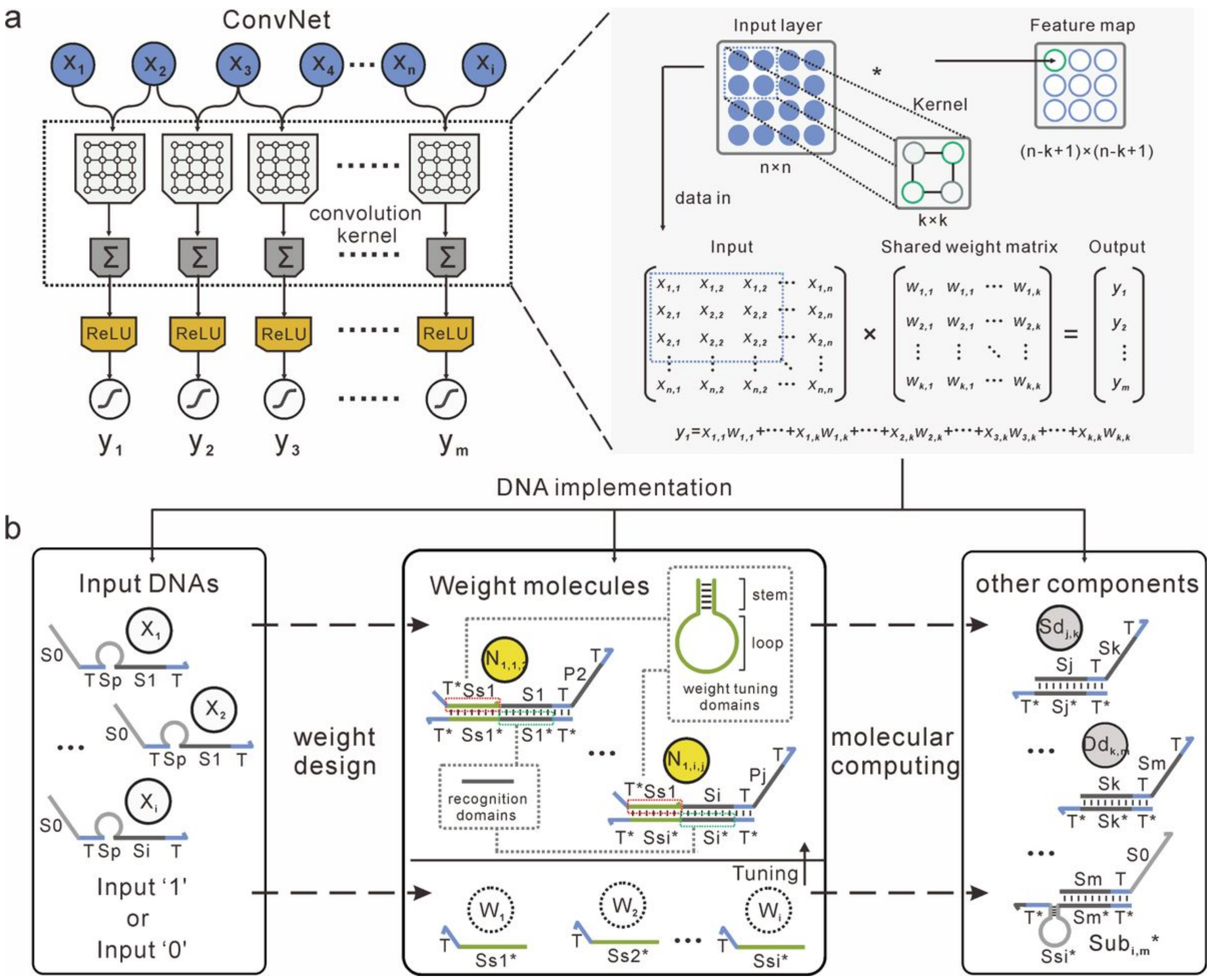

Figure 1

ConvNet and its molecular implementation with DNA regulatory circuit systems. a, Left: The architecture of ConvNet. Note that x1 to xi and y1 to ym are binary inputs and outputs, respectively. Right: Schematic of the operation principle of ConvNet for recognition tasks. The individual receptive region of inputs multiplies with the kernel to perform MAC operation. By this way, the inputs can be convolved with the same kernel by mapping convolution computation into a series of MAC operations. b, Implementation of ConvNet with DNA regulatory circuit systems. The inputs can be encoded by a set of single strands, wherein 1 or 0 represent the presence or absence of input strands. The weight matrix can be stored by programming a set of weight molecules with a simple switching gate architecture.

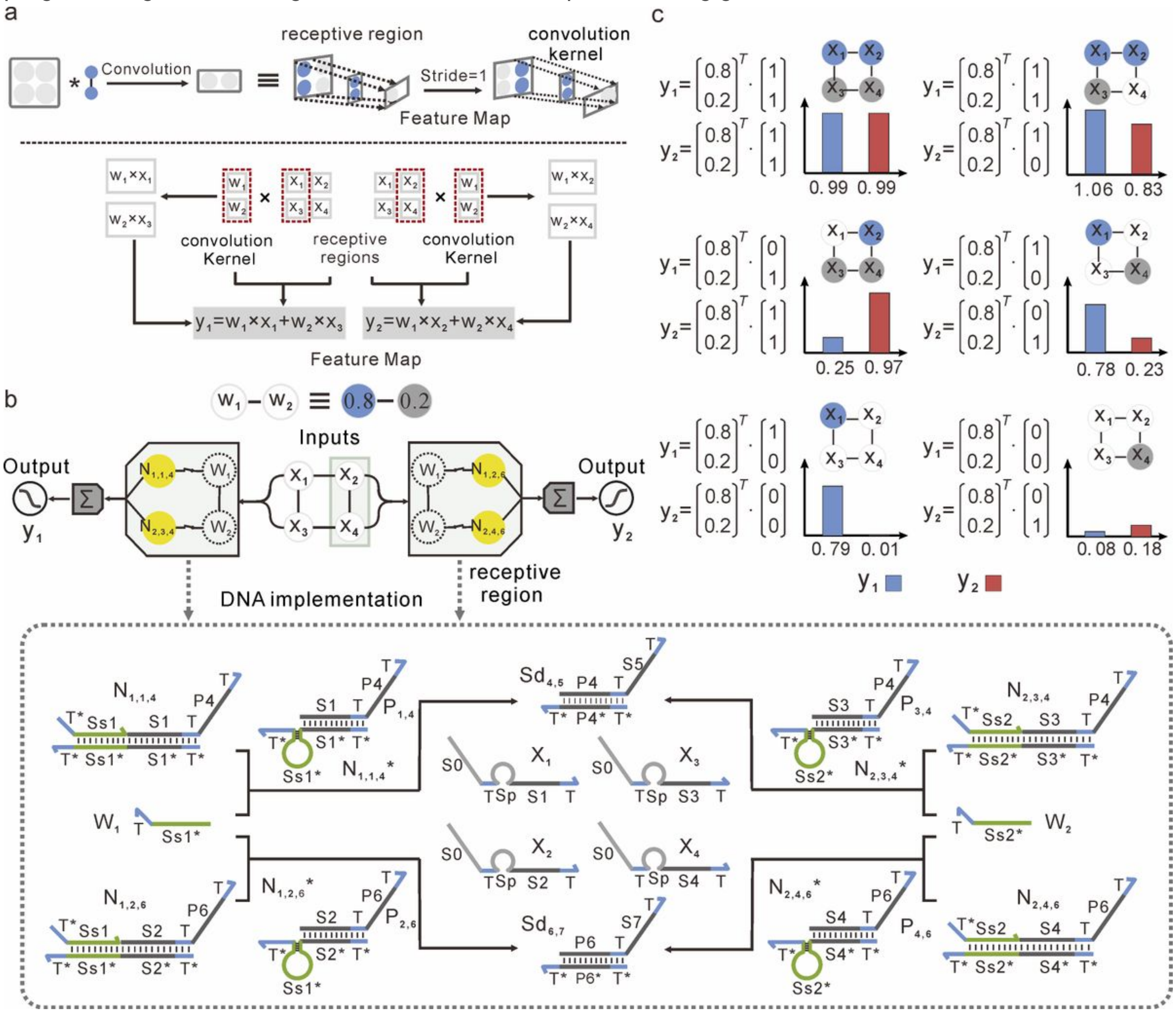

Figure 2

Convolution computation via multiple parallel MAC operations. a, Detailed calculation process of convolution with an input matrix of 2×2, a kernel size of 1×2, and a stride of 1. The feature map is generated from interactions between kernel function and different receptive regions. b, Abstract schematic diagram of convolution (top). The weight substrate molecule Ni,i,j can be activated by corresponding weight tuning molecule Wi, then activated weight substrate molecule Ni,i,j* can interact with local receptive region and export the computation results. The value of weights determined from convolution kernel is 0.8× (blue dot) and 0.2× (gray dot), respectively. DNA implementation of convolution (bottom). The 2×2 input pattern is encoded with four DNA single strands (X1, X2, X3, and X4). The circuit consists of four Ni,i,j, two Sdi,j, two Repi and two Wi. The relative concentrations of Xi, Ni,i,j, Repi and Sdi,j are 2×. The relative concentrations of Wi are 0.8× and 0.2×, receptively. The standard concentration is 1× = 50 nM. c, Characterization of convolution with six different input patterns after 3 h. The blue and red histograms correspond to the fluorescence kinetic outputs of y1 and y2, respectively. Blue dots indicate that the weight W1 that is multiply with corresponding input (X1 or X2), while gray dots indicate that the weight W2 that is multiply with corresponding input (X3 or X4).

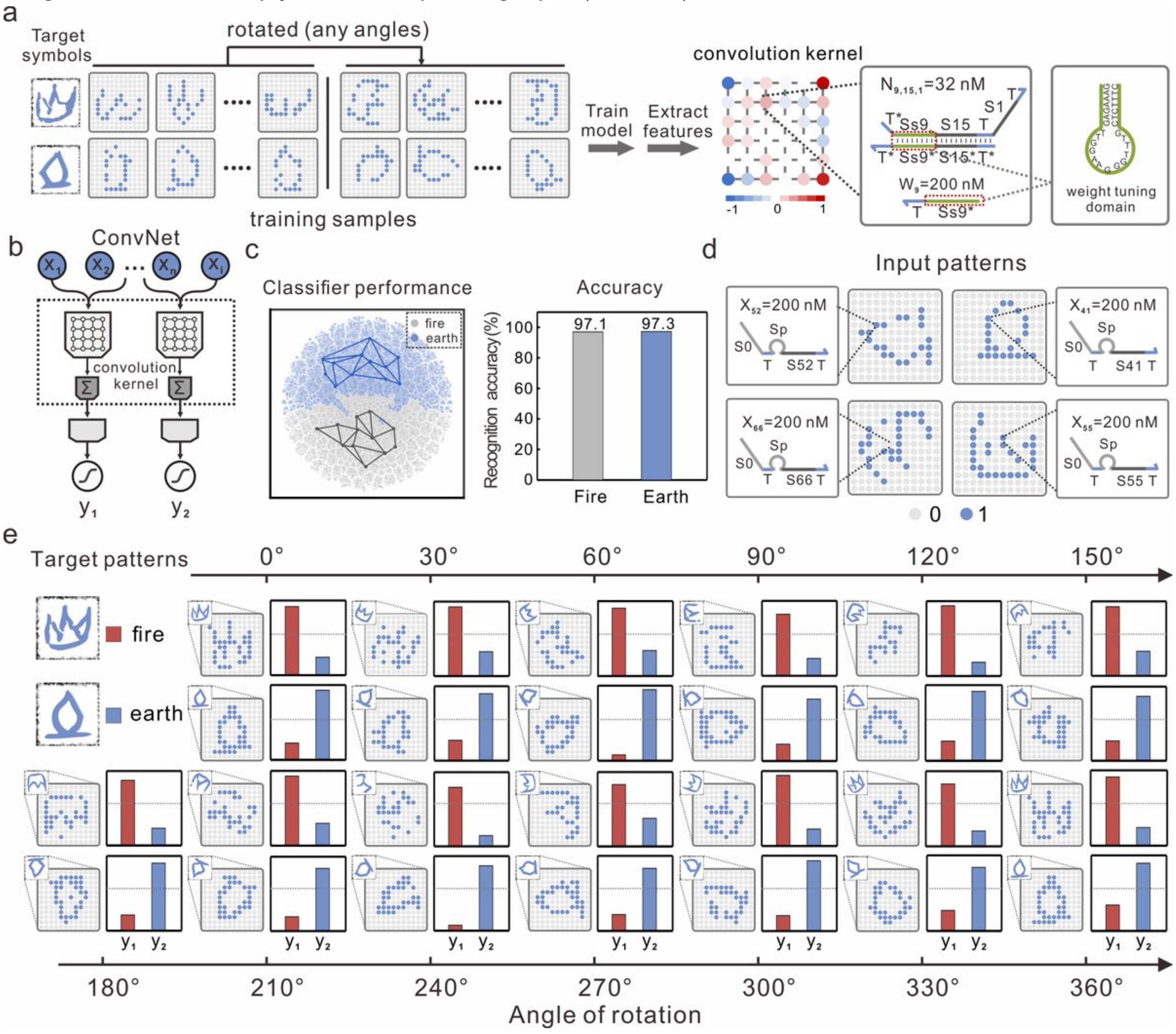

Figure 3

A DNA-based ConvNet for one of two rotated molecular patterns recognition. The training examples ‘fire’ and ‘earth’ are obtained by rescaling and adjusting to 144-bit binary patterns from the Sinica oracle database. The weights are determined from the convolution kernel with dimensions of 6×6. Each pixel of kernel function is encoded in the sequence of weight tuning domains of weight tuning molecules and weight substrate molecules. The value of each pixel (for example, 0.32 for the 9th pixel) in convolution kernel is used to determine the concentration of each weight substrate molecule Ni,i,j, relative to a standard concentration of 100 nM (for example, [N9,15,1] = 32 nM). The concentration of weight tuning molecule Wi that activates the weight multiplication reactions is 2× (for example, [W9] = 200 nM). Detailed process of neural-network training and testing can be seen in Methods 'Neural-network training and testing'. b, Circuit diagram of a DNA-based ConvNet. c, Performance of the ConvNet in silico for dataset with 48,000 oracles, where 97.1% and 97.3% of 'fire' and 'earth' were recognized correctly in theory. d, Four example binary input patterns with 1 (blue dots) and 0 (grey dots), corresponding to inputs with or without DNA strand, respectively. The concentration of each input strand is 2×. e, Characterization of the recognition behavior of 26 representative example 'fire' and 'earth' after 20 h, with rotation angle incremented from 0° to 360° by steps of 30°. Red and blue histograms correspond to the fluorescence kinetic outputs of 'fire' and 'earth', respectively. The gray dotted line marks the threshold value of 0.6. Corresponding fluorescence kinetics data are shown in Supplementary Fig. 17c.

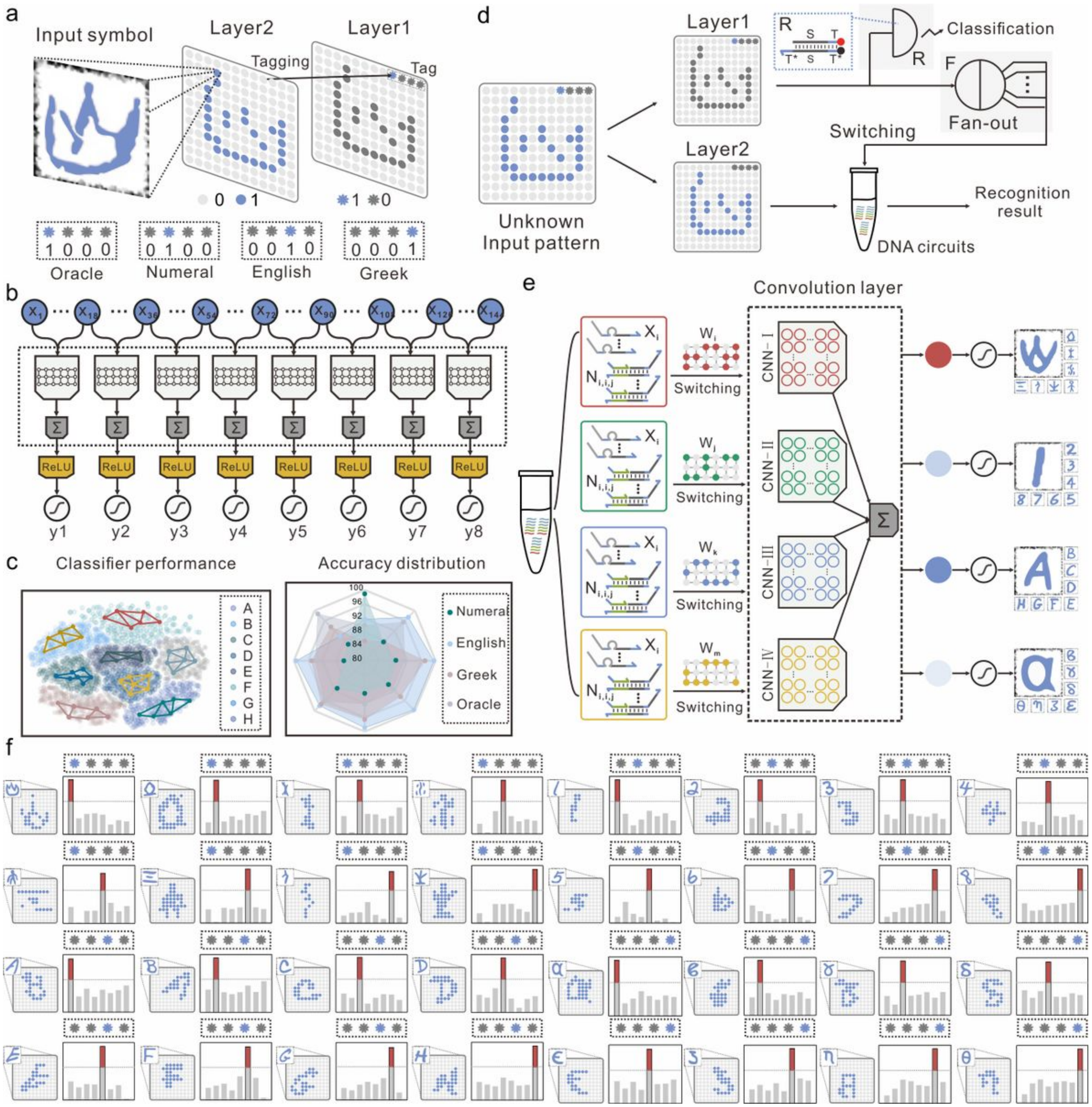

Figure 4

The two-step classification approach based on hierarchical network architecture for 32 molecular patterns recognition. a, Each input pattern is replaced by a pair of input patterns (layer1 and layer2). Layer 1 was attached with tags to classify four groups, in which 1000 corresponds to Chinese oracles, 0100 corresponds to Arabic numerals, 0010 corresponds to English alphabets, and 0001 corresponds to Greek alphabets. Both the tags in Layer 1 and the inputs in Layer 2 are binary patterns, in which each 1 or 0 indicates the presence or absence of a tag strand (or an input strand), respectively. b, Circuit diagram for recognizing 8 distinct patterns of each group. c, Performance of the ConvNet in silico for dataset with 8 English alphabets. d, Example pattern recognition process of ‘fire’ with a pair of input patterns. Layer1 is the input of logic circuit, which is composed of reporter gates R and fan-out gates F. The outputs of the fan-out gates can activate downstream ConvNet circuit to complete the pattern recognition, while the outputs of the reporter gates can be readout to complete the coarse classification. e, Circuit diagram. Specific weight tuning molecules Wi can switch on corresponding circuit components for specific molecular pattern groups. f, Characterization of the recognition behavior of 32 representative input patterns after 36 h. The tag highlighted above the bar graph marks the group of each input pattern. The light gray dotted line marks the threshold value of 0.6 (ON: red histogram; OFF: gray histograms). Corresponding fluorescence kinetics data are shown in Supplementary Figs. 26 and 27.

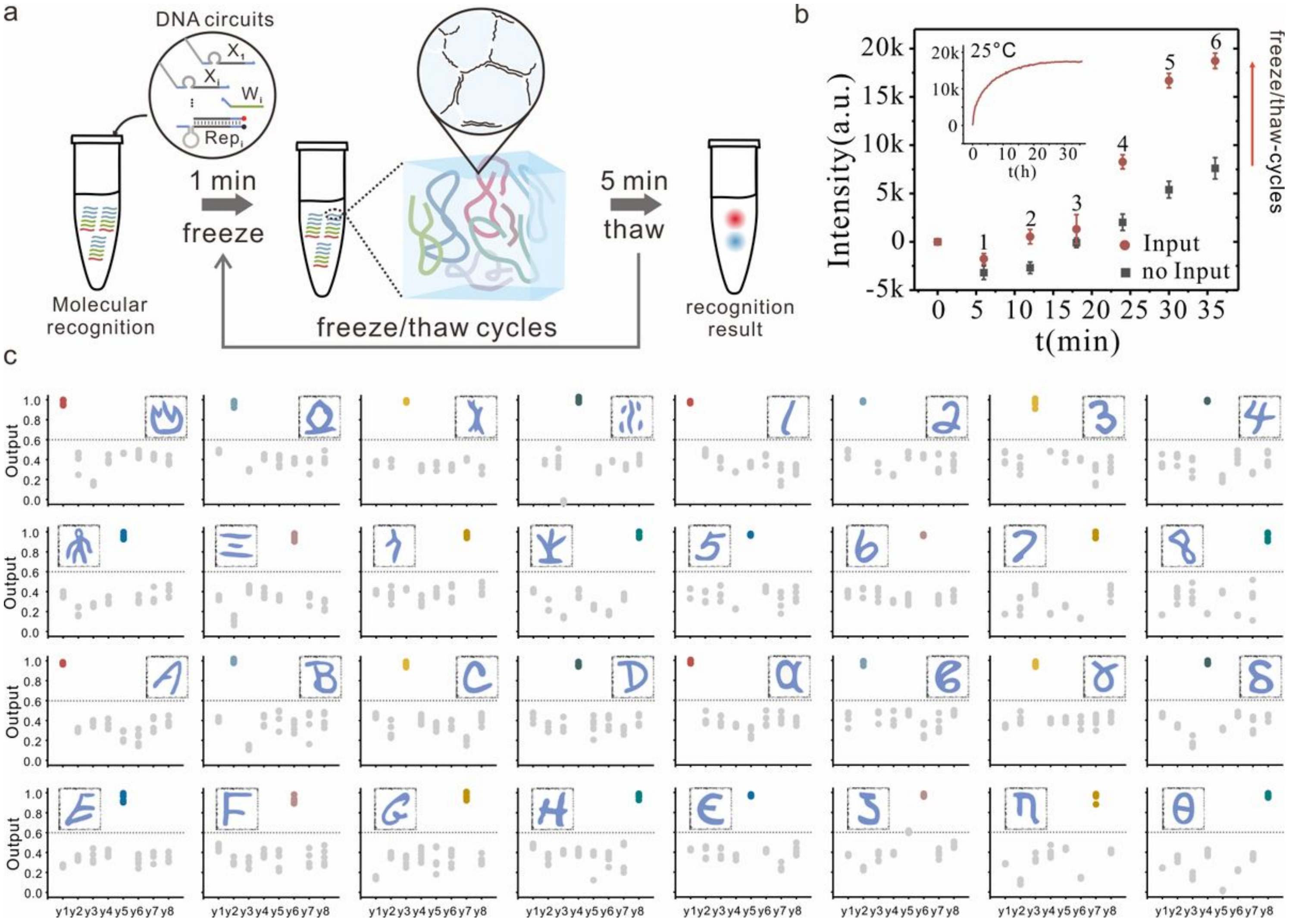

Figure 5

A cyclic freeze/thaw approach accelerate DNA circuits for molecular patterns recognition. a, Operation of cyclic freeze/thaw approach. The circuits were kept in cryopreservation tube (1.5 mL) during repeated freeze/thaw cycling: thawing at 37 °C, followed by cooling to −196 °C. b, Fluorescence levels of DNA circuits for recognizing ‘G’ after different freeze/thaw cycles (inset: the same circuit with continuous incubation at 25 °C). Only reporter Rep7 was readout. c, Fluorescence levels for characterizing the recognition behavior with 160 representative input patterns after 5 freeze/thaw cycles. The light gray dotted line marks the threshold value of 0.6 (ON: colored dots; OFF: gray dots).

Supplementary Files

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

- SupplementarytableConvNet.xlsx
- SIConvNet.docx