nature portfolio

https://doi.org/10.1038/s41586-025-09479-w

Supplementary information

Supervised learning in DNA neural networks

In the format provided by the authors and unedited

Supervised learning in DNA neural networks

Supplementary information

Kevin Cherry and Lulu Qian*

Contents

1 Materials and methods **3**
1.1 Sequence design . . . 3
1.2 DNA oligonucleotide synthesis . . . 6
1.3 Concentration measurements . . . 6
1.4 Annealing protocol and buffer condition . . . 7
1.5 Purification . . . 7
1.6 Fluorescence kinetics experiments . . . 7
1.7 Multi-step learning experiments . . . 8

2 Modeling and simulation **9**
2.1 Activatable weight motif . . . 9
2.2 Learning motif . . . 11
2.3 Learned weights . . . 12

3 DNA implementation design criteria **14**
3.1 Design complexity . . . 14
3.2 Reaction specificity . . . 17
3.3 System-level three-letter code . . . 19

4 Design 1 **22**
4.1 Activatable weight motif . . . 22
4.2 Learning motif . . . 24
4.3 General-purpose implementation of chemical reaction networks . . . 27
4.4 Modular drain . . . 28
4.5 Unreacted drains occluding weight activation . . . 32
4.6 Undesired reversibility of learning . . . 35
4.7 Leak at the interface of learning and testing . . . 38
4.8 Crosstalk in weight activation . . . 41
4.9 Moving to a better design . . . 43

5 Design 2 **47**
5.1 Activatable weight motif . . . 47
5.2 Learning motif . . . 50

5.3 Impact of domain length on the interface leak 53
5.4 Scaling up to a 100-bit learning neural network 56
5.5 Label occlusion 59
5.6 Annealing ratio and complex purity 61
5.7 Effectiveness of fuel 63
5.8 Improved performance in a 100-bit learning neural network 65
5.9 Input leak in learning 67
5.10 Further improvement in a 100-bit learning neural network 70
5.11 System bias 73
5.12 Good teacher 76
5.13 Sample evaporation 79

6 DNA sequences **83**

References **116**

1 Materials and methods

1.1 Sequence design

Toehold pool

A 7-nt toehold pool for Lj on the learning gates and Tj on the weight gates was designed using StickyDesign.[1] We generated a pool of five orthogonal toeholds. Four were required and an additional one was made for backup in case an unexpected spurious reaction was discovered during later experiments. The provided function 'easyends' was used to design a set of roughly equivalent energy, orthogonal toeholds. In order to also avoid unintended interactions with components in other layers of the neural network, the 5-nt universal toehold T was pre-populated in the initial sequence array using the 'oldends' parameter. For use in this function, it was artificially extended to 7 nucleotides by adding two As to the 5' end, although this was later found to be unnecessary. The 'energeticsDAOE' energy function was used with a temperature of 25 °C and 'coaxparams' was set to 'protozanova'. The 'easyends' function generated five new 'DT' style toeholds with the alphabet set to H (A, C, or T), both adjacent nucleotides set to A, 'fdev' set to 0.05, and 'maxspurious' set to 0.30. The function was run until we obtained a set of toeholds that did not contain any consecutive subsequences of the same nucleotide longer than length 4. The average Gibbs' free energy of the resulting 7-nt toehold pool was −9.87 kcal/mol with standard deviation 0.038 kcal/mol.

```
en=EnergeticsDAOE(enclass='EnergeticsDAOE', temperature=25,
coaxparams='protozanova', danglecorr=True, singlepair=False, version='0.6.0')

stickydesign.easyends('DT', 7, number=5, energetics=en, alphabet='h',
adjs=['a','a'], fdev=0.05, maxspurious=0.30, oldends=oldends)
```

The 9-nt AT-only toehold U was also selected using StickyDesign. We generated all toehold sequences of length 7, 8, 9, or 10 consisting only of A and T nucleotides that did not contain a run of more than 4 of the same nucleotide. All energies were calculated using the 'EnergeticsDAOE' energy class, with a temperature of 25 °C, and 'protozanova' set to coaxparams. The 'energy_array_uniform' function was used to calculate the Gibbs' free energy of each sequence. Our initial chose was a 7-nt AT-only toehold U that had $\Delta G = -6.31$ kcal/mol. To match the strength of the 7-nt toeholds with A, T, and C nucleotides used in earlier experiments, which had energy ranging from −9.36 to −11.85 kcal/mol, we updated the toehold U to 9 nucleotides (Fig. S19c). The ten strongest AT-only toehold sequences of 9 nucleotides had a calculated free energy of −9.59 kcal/mol and were selected for further analysis. The ten candidate toehold sequences were appended to an array of the four chosen 7-nt Lj and Tj toeholds and the 'energy_array_uniform' function was used to calculate all pairwise binding energies. We selected the toehold sequence that was most orthogonal to the four 7-nt toeholds by having the highest minimum calculated free energy indicating the least interaction.

Xi pool

The 15-nt Xi sequence pool was designed through iterative optimization subject to heuristic design constraints. The new sequence pool was pre-populated with the six domains (P1, P2, S1, S2, Y1, and Y2) used in the downstream layers of the network, which were selected from a previously

published sequence pool.[2] We have successfully utilized these six domain sequences in winner-take-all DNA neural networks trained *in silico*,[3] and therefore they were not edited during the pool optimization process. The previously published sequence pool only contained 56 sequences, and thus was not suitable for the 100 Xi domain sequences needed here.

The Xi pool was initialized with 100 new sequences that met some basic selection criteria. First, all sequences must contain between 30% and 70% Cs. Second, the 3' end cannot be 'AA' in order to avoid stronger occlusion between the activator strand and the input strand when the bulge domain B∗ is complementary to the first two nucleotides on the 3' end of the Xi domain. Third, the 5' end was set to 'CA' to be compatible with the universal clamp sequence used in the seesaw gates.[2] Fourth, the ninth and tenth nucleotides were set to 'C' and 'A' in order to lock the T1 or T2 toehold loops into place (Fig. S21f). Finally, for synthesis reasons sequences cannot contain five consecutive Ts, five consecutive As, and four consecutive Cs, and for occlusion reasons, they cannot contain U∗ or any 9-nt subsequences of label inhibitor strands Inh_1 and Inh_2. New sequences in the iterative procedure described below can be added to the initial sequence pool if they meet these criteria.

Combining the 6 pre-populated sequences and the 100 new Xi sequences, an initial pool of 106 sequences was created. We then optimized the pool to minimize the similarity of the sequences. To measure similarity, three scores were computed for each pair of sequences including the count of matching bases at each nucleotide index, the longest length run of these matching bases, and the length of the longest common subsequence. An additional penalty of five was added to the second score to increase its importance compared to the other scores.

The scores for each sequence were converted to a polynomial where the coefficients were the frequencies of a specific score across the pool and the exponent was the score value. These polynomials were easy to manipulate mathematically and allowed us to keep the different scores separate. Commonly, a sequence in the initial pool had a few high-score similarities with the rest of the pool, but dozens more low-score similarities. We chose to reduce higher scores at the expense of increasing the frequency of lower scores. This optimization approach allowed us to preferentially reduce the number of severe sequence interactions while still attempting to reduce minor sequence interactions.

Sequences were ranked by total similarity to the rest of the pool with the largest polynomial ranked first. New sequences were chosen to replace old sequences according to a probability function $1/x^{2+i/n}$, where x is the ranking index of the sequence, i is the current number of iteration, and n is the maximum number of iterations. At the beginning of the optimization process the function was $1/x^2$, and became $1/x^3$ at the last iteration. Thus, the sequence with the worst score is always replaced with probability equal to one. The probability of replacing the next worst sequence begins as 1/4, and the probability of replacing the next few sequences in the ranking quickly approaches zero. As the sequence pool becomes more highly optimized near the end of the iteration process, the probability of replacing more than only the worst sequence decreases. At each iteration, all sequence scores were summed into a pool score which was compared to the previous minimum pool score. If the new pool score was lower, that pool was kept. The optimization procedure was run for ten million iterations.

Ai pool

The 9-nt Ai sequence pool was optimized via the same iterative procedure as the Xi sequence pool. However, the process for creating the initial sequence pool and the constraints on new sequences

were different. New candidate sequences must meet three criteria: First, same as the Xi sequence pool, all sequences must contain between 30% and 70% Cs. Second, the 3' nucleotide was set to C, which helps avoid stronger occlusion between the activator strand and the label inhibitor strand when the clamp domain c is complementary to the first two nucleotides on the 3' end of the Ai domain. Finally, for synthesis reasons sequences cannot contain five consecutive Ts, five consecutive As, and four consecutive Cs (same as the Xi sequence pool), and for occlusion reasons, they cannot contain any 6-nt subsequences of toeholds T1 and T2.

Acquiring an optimized pool of 100 Ai sequences was expedited by requiring eighty sequences to have eighty different four nucleotide tuples of A, C, and T on their 5' end. Out of $3^4 = 81$ possibilities, the last tuple consisting of four Cs was excluded for synthesis reasons. These four nucleotides of each sequence was immutable. Using the eighty different tuples on the 5' end guaranteed the initial branch migration of the Ai domain would be as orthogonal as possible without having to randomly explore that part of the sequence space. The remaining five nucleotides of those eighty sequences as well as the twenty unspecified sequences were filled in randomly, provided they met the above criteria for new candidate sequences. The optimization routine was run with ten million iterations using the same sequence similarity scoring and sequence replacement function described for the Xi sequence pool.

1.2 DNA oligonucleotide synthesis

Except for experiments that compared strand purity (for example, Fig. S21d), all unmodified DNA strands were ordered unpurified (standard desalting) from Integrated DNA Technologies (IDT) with the LabReady service, shipped at a 100 μM concentration in IDTE Buffer, pH 8.0. Strands modified with fluorophores or quenchers were ordered HPLC purified.

1.3 Concentration measurements

DNA concentration was measured using two methods (Fig. S1). A Nanodrop (Thermo Fisher) was used for low-throughput measurements such as a few tubes of stock DNA or tubes of complexes recovered from in-house PAGE purification. A Take3 microvolume plate and a Synergy H1 (Biotek) microplate reader were used for high-throughput absorbance measurements, such as measuring concentration of stock DNA stored in 96-well plates (IDT deep well plates or Echo source plates). In order to determine accurate concentrations for short DNA strands, the extinction coefficient of a single strand or two-stranded complex was calculated for each sample.

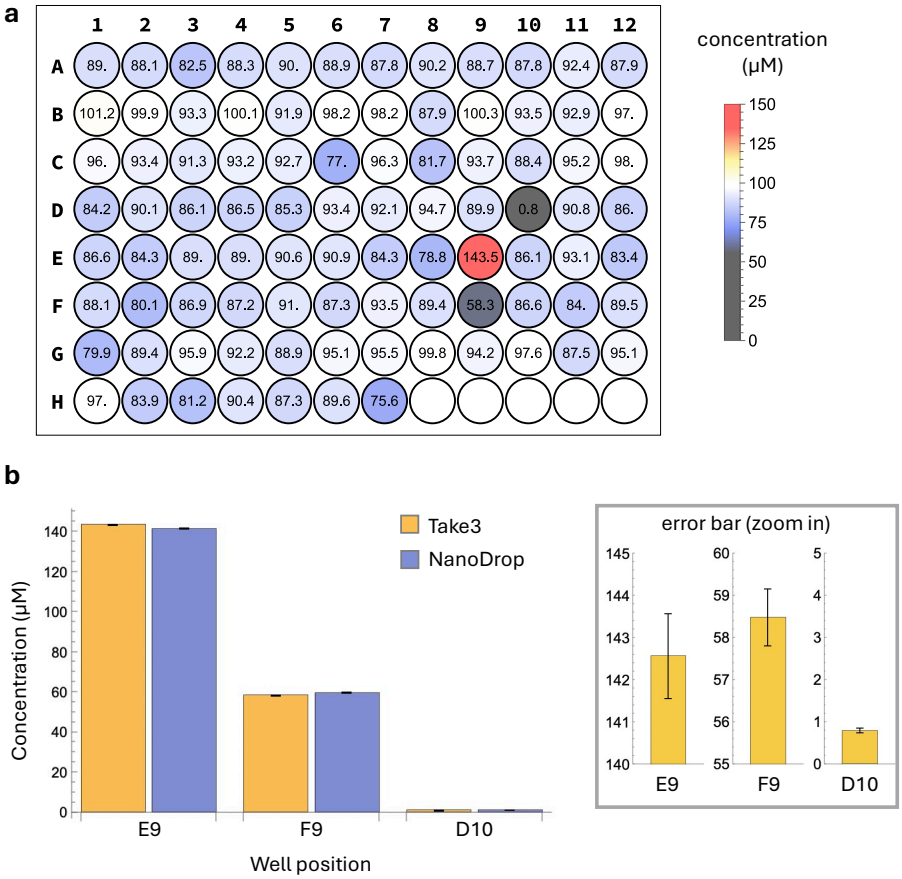


Fig. S1 | Concentration measurements. **a,** Measured concentrations of DNA strands from a 96-well deep well plate upon arrival from Integrated DNA Technologies, ordered with LabReady at a nominal concentration of 100 μM. **b,** Comparing the high-throughput measurements using a Take3 microvolume plate and the low-throughput measurements using a NanoDrop.

1.4 Annealing protocol and buffer condition

Complexes were prepared by mixing stock DNA strands with buffer and then annealed at the highest concentration possible in either 0.5 mL tubes or in conical 96-well plates. Strands were ordered at a nominal concentration of 100 μM, but in practice were typically at lower concentrations (Fig. S1a). For two-stranded complexes annealed from 100 μM strands at a 1:1 ratio, 10% of the final sample volume was a TE buffer with 10× Mg^{2+} at 125 mM, yielding a final solution of complex at 45 μM in TE buffer with 1× Mg^{2+} at 12.5 mM. With typical stock concentrations lower than 100 μM, complex concentrations around 41 μM were more common. Volumes of stock DNA strands were adjusted accordingly if the annealing ratio was different, such as for 1:1.2 (Fig. S24).

The mixing procedures were performed either by hand pipetting (using Eppendorf Research Plus pipettes) when making only a few complexes. When the 200 learning gates (inhibited activators) and 200 weight gates (inhibited weights) were annealed, we used an Echo liquid handler (Beckman Coulter) which transferred stock strands and buffer into 96-well plates. Since all of the stock strands had different starting concentrations, the liquid handler was programmed to adjust for the concentration of each strand. It was important all of these complexes remained relatively at the same concentration after combining the mixes and before in-house PAGE purification, so the Echo also transferred a specific amount of TE buffer. In the end, all plate wells contained complex mixes with the same volume and concentration. Plates were briefly centrifuged to collect liquid at the bottom of the wells, and then they were covered with a metallic sticker cover to prevent evaporation.

Annealing was performed in a thermocycler (Thermo Fisher). The samples were held at 90 °C for 5 minutes before ramping down from 90 °C to 20 °C in steps of 1 °C per minute. The temperature was then held at 20 °C until samples were collected.

1.5 Purification

DNA complexes were purified using electrophoresis (Cytiva SE 600 Ruby) on 12% polyacrylamide gels formed with TAE buffer. Annealed DNA samples were mixed with a 10× loading buffer containing 25% Ficoll-400 and bromophenol blue dye. Between 40 and 180 μL of sample was loaded into each well. Electrophoresis occurred at a constant 150 V for between 9 and 15 hours while immersed in a 20 °C water bath. Target bands were cut from the gel while visualizing their shadow under UV light, cut into small pieces, and incubated in TE buffer with 12.5 mM Mg^{2+} for 24 hours. Buffer containing the purified complexes was extracted from the incubation tube and transferred to a new tube. The concentration of the purified complexes was determined by Nanodrop (Thermo Fisher). The summation gates, annihilator, and restoration gates were purified individually. For the inhibited activators and inhibited weights, all complexes were annealed individually then mixed at equal ratios into four pools before purification. Each pool contained 100 gates of the same type for one of the two memories. The average extinction coefficient was used to calculate the concentration of each pool of gates purified in one pot.

1.6 Fluorescence kinetics experiments

Fluorescence kinetics experiments were performed with standard concentration equal to 50 nM or 100 nM in glass-bottom plates (Corning) in a microplate reader (Synergy H1, Biotek). Diagnostic tests were conducted in 50 μL or 100 μL volume in 96-well plates. High-throughput experiments, for

example, those showing learned values for 100 weights, were conducted in 30 μL volume in 384-well plates. Early experiments used a 20T strand at 1 μM, but we switched to using 0.01% Tween 20 (Sigma-Aldrich) which more effectively prevented strand loss during liquid handling events (Fig. S2). Fluorescence reads were taken every 2, 4, or 8 minutes. Four fluorophores were used with distinct excitation and emission wavelengths of 496 nm and 525 nm for ATTO488, 555 nm and 582 nm for ATTO550, 598 nm and 629 nm for ATTO590, and 639 nm and 669 nm for ATTO647, respectively.

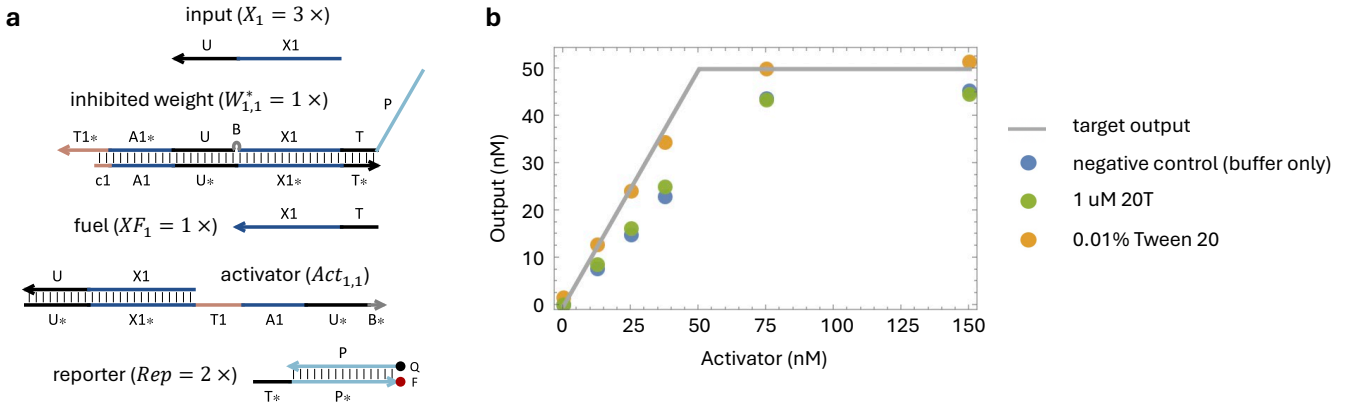


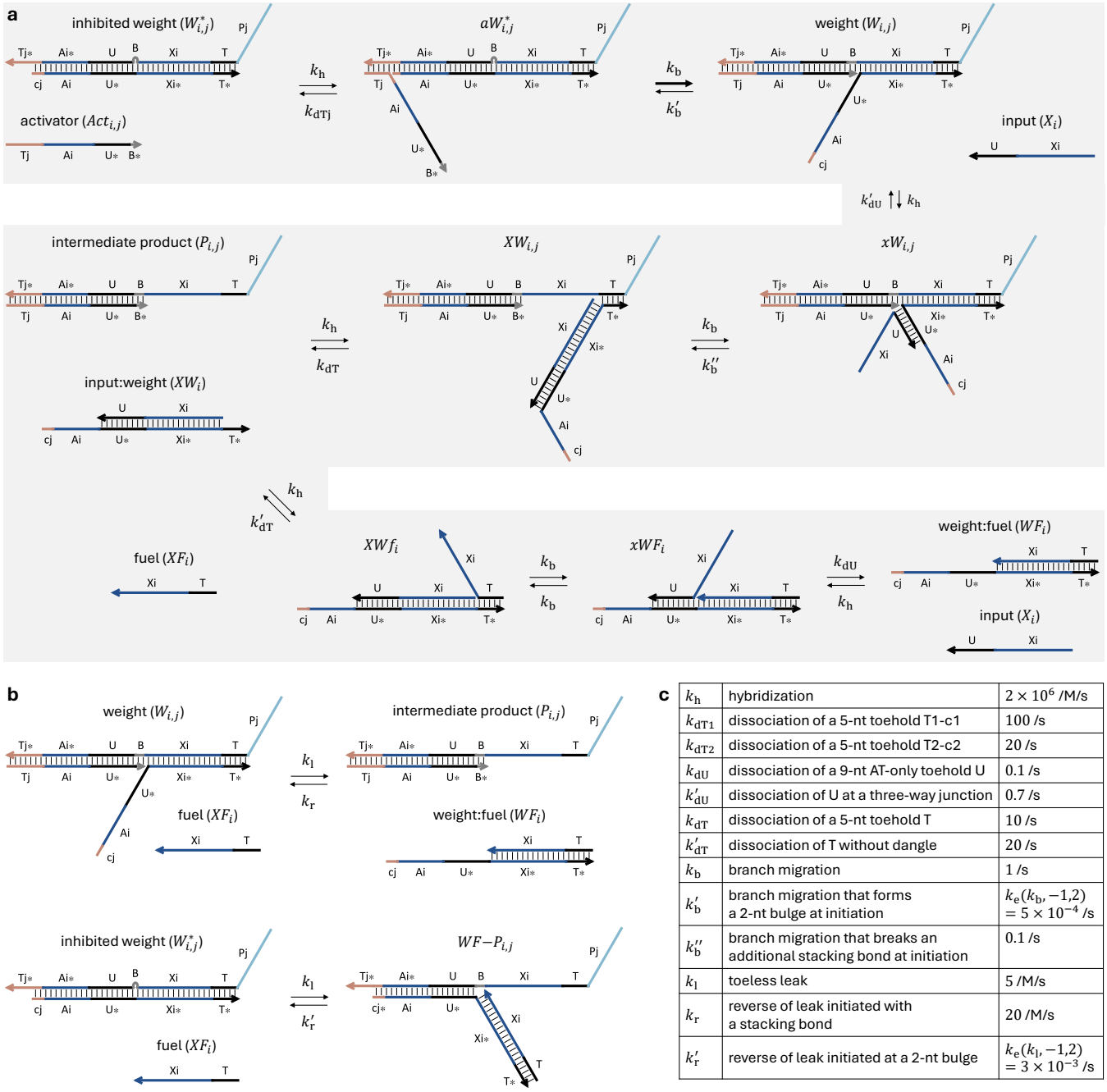
Fig. S2 | Comparing the effectiveness of Tween 20 and a 20T DNA strand as a passivating agent. a, Domain-level diagrams of molecules used in the experiments. Standard concentration $1\times = 50$ nM. **b,** End-point fluorescence data of measured output concentration versus activator concentration.

1.7 Multi-step learning experiments

Learning experiments required components to be added in multiple stages for sequentially presenting the training patterns and for transition from learning to testing. Learning was performed in a master mix incubated at 25 °C. That master mix was then distributed into multiple wells on a plate, one for each test pattern. The inhibited weights and other molecules in the downstream layers were also made in a single master mix and distributed. This process allowed for all sample wells to be filled with the same trained network. Unique test patterns were finally added to each individual sample well. The volume and concentration of molecules changed when the two master mixes were combined and the test pattern was added. The reported concentrations for all molecules were calculated at the final sample volume. Experiments that required training had a sequential learning master mix at 30% of the final volume, or two parallel learning master mixes each at 30% volume. The testing master mix volume was also 30%. The remaining 10% was reserved for test patterns. This means learning always occurred at $3.33\times$ the concentration of the final experiment. Adding a small volume of training patterns, labels, and label inhibitors changed the learning master mix volume by approximately 15% in total.

2 Modeling and simulation

2.1 Activatable weight motif



c

k_h	hybridization	2×10^6 /M/s
k_{dT1}	dissociation of a 5-nt toehold T1-c1	100 /s
k_{dT2}	dissociation of a 5-nt toehold T2-c2	20 /s
k_{dU}	dissociation of a 9-nt AT-only toehold U	0.1 /s
k'_{dU}	dissociation of U at a three-way junction	0.7 /s
k_{dT}	dissociation of a 5-nt toehold T	10 /s
k'_{dT}	dissociation of T without dangle	20 /s
k_b	branch migration	1 /s
k'_b	branch migration that forms a 2-nt bulge at initiation	$k_e(k_b, -1,2) = 5 \times 10^{-4}$ /s
k''_b	branch migration that breaks an additional stacking bond at initiation	0.1 /s
k_l	toeless leak	5 /M/s
k_r	reverse of leak initiated with a stacking bond	20 /M/s
k'_r	reverse of leak initiated at a 2-nt bulge	$k_e(k_l, -1,2) = 3 \times 10^{-3}$ /s

Fig. S3 | Modeling of the activatable weight motif. a-b, Desired (**a**) and leak (**b**) reactions. **c,** Rate constants.

The forward rate of a reaction with known reverse rate k, a total of l nucleotides gained, and a bulge loop of size s formed was estimated as follows and applied to derive k'_b and k'_r with gas

constant $R = 1.9872$ and temperature $T = 298.15$ K (25 °C):

$$k_{\mathrm{e}}(k, l, s) = k \times e^{-\frac{(\Delta H_{\mathrm{bp}} - T\Delta S_{\mathrm{bp}}) \times l + T\Delta S_{\mathrm{bg}}(s)}{RT}}$$

where ΔH_{bp} (unit: cal per mol) is the average enthalpy of forming a base pair:

$$\Delta H_{\mathrm{bp}} = \mathrm{Mean}[\{-7.6, -7.2, -7.2, -8.5, -8.4, -7.8, -8.2, -10.6, -9.8, -8.0\}] \times 1000$$

ΔS_{bp} (unit: cal per K per mol) is the average entropy of forming a base pair:

$$\Delta S_{\mathrm{bp}} = \mathrm{Mean}[\{-21.3, -20.4, -21.3, -22.7, -22.4, -21, -22.2, -27.2, -24.4, -19.9\}]$$

$\Delta S_{\mathrm{bg}}(s)$ is the entropy of forming a bulge of size s:

$$\Delta S_{\mathrm{bg}}(1) = (\Delta G_{37,\mathrm{bg}}(1) + \Delta G_{37,\mathrm{bp}}) \times 1000/310.15$$
$$\Delta S_{\mathrm{bg}}(s) = \Delta G_{37,\mathrm{bg}}(s) \times 1000/310.15$$

$\Delta G_{37,\mathrm{bg}}(s)$ is the free energy of forming a bulge at 37°C:

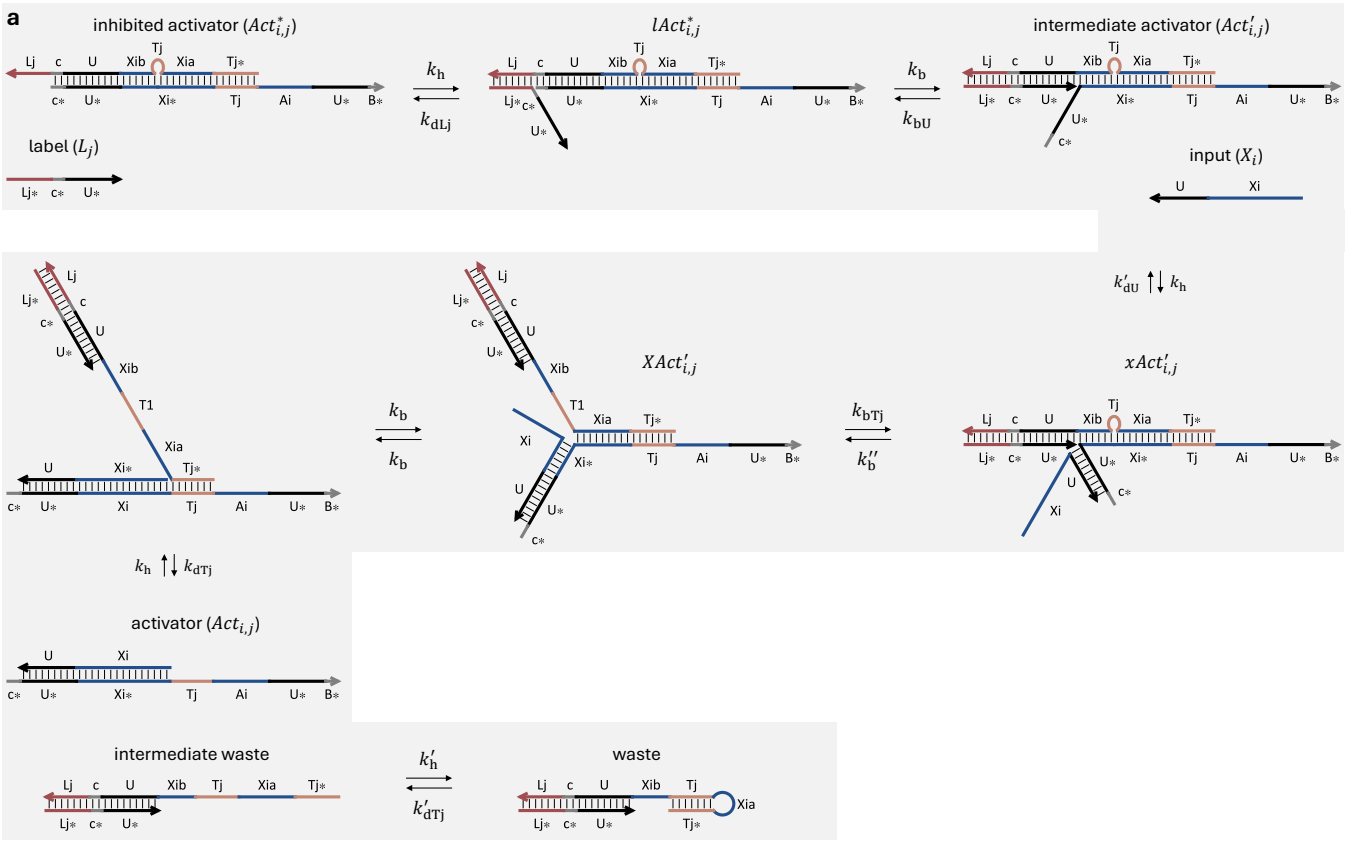
$$\Delta G_{37,\mathrm{bg}}(s) = \{4, 2.9, 3.1, 3.2, 3.3, 3.5, 3.7, 3.9, 4.1, 4.3\} \text{ for } s = 1 \text{ to } 10$$

$\Delta G_{37,\mathrm{bp}}$ is the average free energy of forming a base pair at 37°C:

$$\Delta G_{37,\mathrm{bp}} = \mathrm{Mean}[\{-1, -0.88, -0.58, -1.45, -1.44, -1.28, -1.3, -2.17, -2.24, -1.84\}]$$

Thermodynamic parameters for base pairs and bulges are taken from SantaLucia and Hicks.[4]

2.2 Learning motif



b

k_h	hybridization	2×10^6 /M/s
k_{dLj}	dissociation of a 7-nt toehold Lj	0.1 /s
k'_{dU}	dissociation of a 9-nt AT-only toehold U at a three-way junction	0.7 /s
k_{dTj}	dissociation of a 7-nt toehold Tj	1 /s
k'_{dTj}	dissociation of Tj next to a 9-nt Xia hairpin loop	0.5 /s
k_b	branch migration	1 /s
k_{bU}	branch migration rate with AT-only sequence at initiation	10 /s
k''_b	branch migration that breaks an additional stacking bond at initiation	0.1 /s
k_{bTj}	branch migration rate that forms a 7-nt Tj bulge at initiation	$k_e(k_b, 0, 7) = 2.5 \times 10^{-3}$ /s
k'_h	unimolecular hybridization rate that forms a 9-nt Xia hairpin loop	$k_{h1}(9) = 2.2 \times 10^2$ /s

Fig. S4 | Modeling of the learning motif. a, Reactions. **b,** Rate constants.

Unimolecular hybridization rate for closing a hairpin loop of size n: $k_{h1}(n) = k_h \times e^{-\frac{\Delta S_{hp}(n)}{R}}$, where $k_h = 2 \times 10^6$ /M/s is bimolecular hybridization rate, $\Delta S_{hp}(n)$ is the entropy of forming a hairpin loop of size n: $\Delta S_{hp}(n) = \Delta G_{37,hp}(n) \times 1000/310.15$, where $\Delta G_{37,hp}(n)$ is the free energy of forming the hairpin loop at 37°C:

$$\Delta G_{37,hp}(n) = 6.3 + 2.44 \times R/1000 \times 310.15 \times \ln(n/30) \text{ for } n > 4$$

and gas constant $R = 1.9872$. Thermodynamic parameters for hairpin loops are taken from SantaLucia and Hicks.[4]

2.3 Learned weights

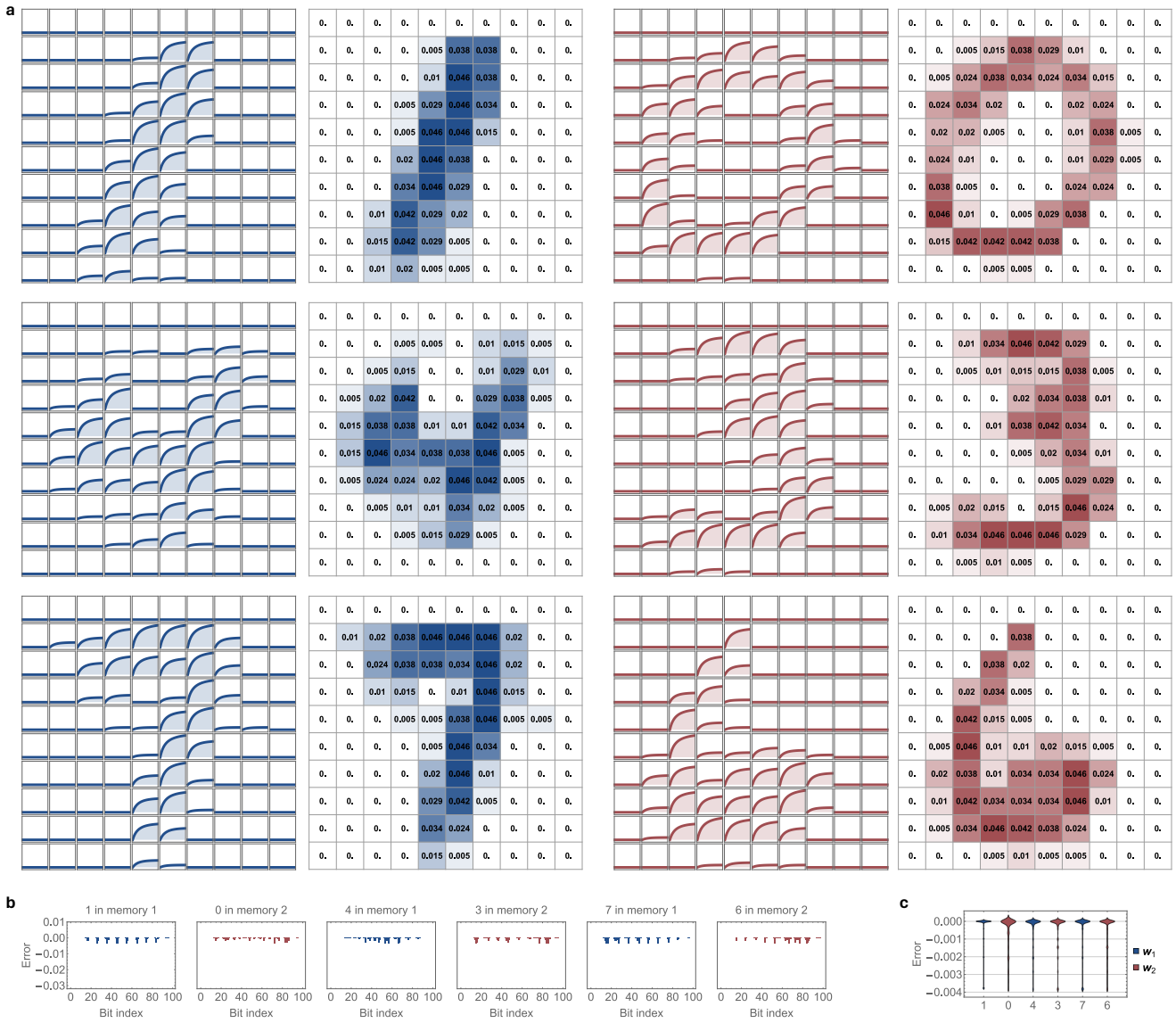


Fig. S5 | Simulations for learning 100-bit patterns. a, Mass-action kinetics simulations for 24 hours of learning and simulation-derived endpoint activator concentrations relative to a standard concentration of $1\times = 50$ nM. **b,** Simulation-derived error statistics. **c,** Distribution chart of errors for learning all six handwritten digits.

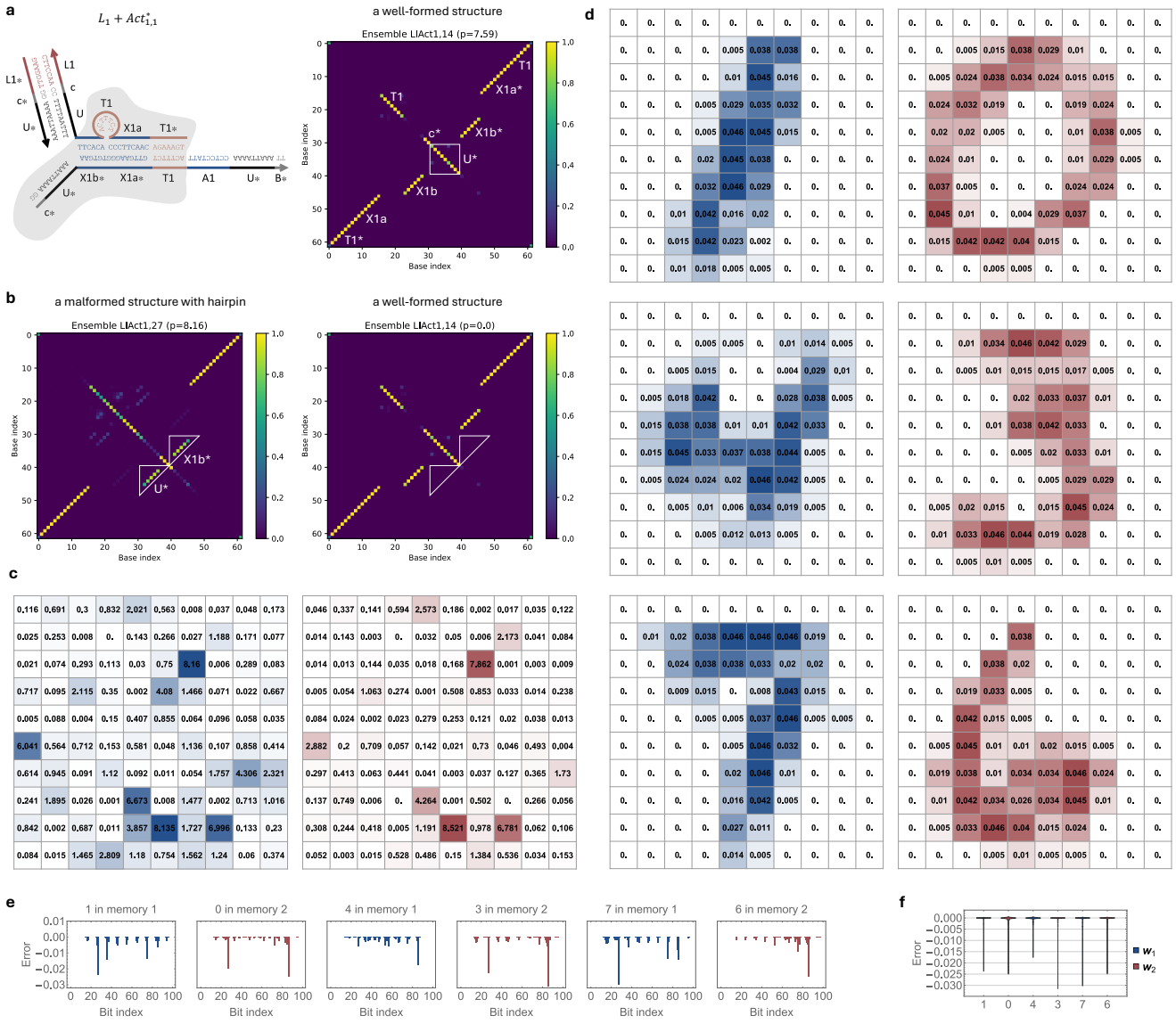


Fig. S6 | Revised simulations for learning 100-bit patterns based on NUPACK analysis. a, Sequence-level diagram of a label-bound learning gate and NUPACK analysis of a highlighted region in the structure. The expected number of unpaired bases in the toehold U∗ is estimated as 7.69 (max = 9 for a 9-nt toehold). **b,** Example NUPACK analysis of a malformed structure with a hairpin between the U∗ and X1b∗ domains on the bottom strand of the learning gate, compared to a well-formed structure without any hairpins. The expected number of paired bases in the toehold U∗ is estimated as 8.16 and 0 for the malformed and well-formed structures, respectively. **c,** Expected number of paired bases in the toehold U∗ for all 100 label-bound learning gates in each of the two memories. **d,** Simulation-derived endpoint activator concentrations relative to a standard concentration of 1× = 50 nM, using reaction rates adjusted based on the NUPACK estimation of the toehold U∗ availability for input binding. **e,** Simulation-derived error statistics. **f,** Distribution chart of errors for learning all six handwritten digits.

3 DNA implementation design criteria

3.1 Design complexity

We established three design criteria for evaluating the complexity of a DNA implementation.

Criterion 1: The fewer types of molecules the better. In a complex molecular system, besides the reactions designed to carry out desired functions, undesired spurious reactions are unavoidable. To understand what kinds of spurious reactions could occur, we typically start with a pairwise investigation across all types of molecules (each type is commonly referred to as a species), and expand it to spurious reactions that involve three or more species when necessary. Naturally, the investigation is simpler and the solutions to mitigate the spurious reactions are easier to find if there are fewer species.

Criterion 2: The fewer strands per molecule the better. A challenge that limits the scalability of a molecular system is how well each molecular structure is formed. It is impossible to achieve perfect stoichiometry when two or more strands are mixed together to assemble into a single complex, due to errors in DNA concentration measurements and pipetting. It is known that contaminates from DNA synthesis affects the absorbance at 260 nanometers (A260) in a spectrophotometer, further complicating the results of the measurements. PAGE purification of the complexes can help remove excess strands, as a solution to the stoichiometry problem. However, the yield of purification depends on the molecular concentration, and thus complexes are typically purified at a high concentration (for example, 45 micromolar), in some cases allowing for excess strands to bind to the complex via a complementary toehold or nonspecific hybridization and remain present in purified samples. With fewer strands, there are fewer ways that excess strands could form spurious structures and interfere with desired stoichiometry.

Criterion 3: The fewer nucleotides per strand the better. How well each molecular structure is formed also depends on synthesis errors. Given a success probability p of each nucleotide attachment in chemically synthesized DNA strands, the expected fraction of full-length product is $p^{(n-1)}$ for strands with n nucleotides. Truncations in the toehold would slow down the desired reaction rate, whereas truncations in the branch migration domain near the toehold would speed up the undesired leak. Purchasing PAGE or HPLC purified strands helps reduce the synthesis errors, but is cost-prohibitive for constructing complex molecular systems that contain hundreds to thousands of unique strands, especially consider the many design iterations. Utilizing shorter strands in a molecular design promotes a larger fraction of full-length products, improving the system behavior even when all strands are purchased unpurified.

Let us consider five example implementation schemes for the learning reaction $X_i + L_j \rightarrow Act_{i,j}$ and evaluate their complexities using the above criteria (Fig. S7). To be consistent, we will use the same format of activator $Act_{i,j}$ in all examples – it contains three consecutive domains Tj$*$, Ai$*$, and T$*$. The choice of the format is not important here, as a correct implementation simply requires the toehold Tj$*$ in the activator to be covered when X_i or L_j is absent and otherwise revealed after both X_i and L_j have reacted with the fuel species.

The first scheme utilizes a general-purpose chemical reaction network implementation.[5,6] In this scheme, both label L_j and input X_i have the same format of a toehold followed by a branch migration domain. Inhibited activator $Act^*_{i,j}$ is a fuel species that reversibly reacts with the label and input to produce the activator. Drain $D_{i,j}$ is an additional fuel species that irreversibly consumes a waste strand from the first reaction and drives the overall reaction pathway forward. The inhibited

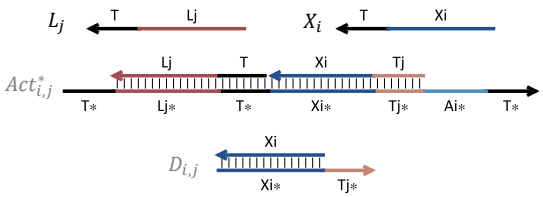
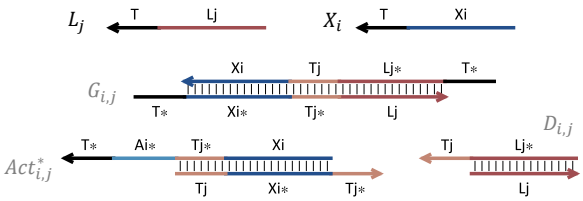
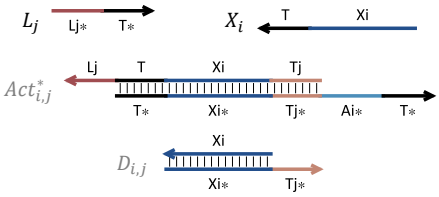
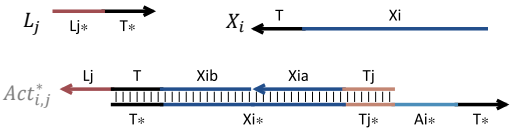
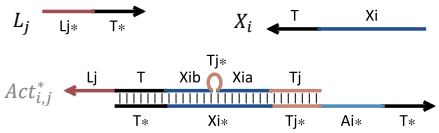
	Implementation scheme for $X_i + L_j \rightarrow Act_{i,j}$	Types of fuel molecules	Strands per molecule	Nucleotides per strand
i		2	3	67
ii		3	2	44
iii		2	2	45
iv		1	3	56
v		1	2	45

Fig. S7 | Design complexity. Five example implementation schemes are compared for reaction $X_i + L_j \rightarrow Act_{i,j}$. Species names in black indicate signal species that appear in the formal chemical reaction. Species names in gray indicate fuel species designed to facilitate the desired reaction. Types of fuel molecules are counted as the number of gray species. Strands per molecule are counted as the largest number of strands across all gray species. Nucleotides per strand are counted as the largest number of nucleotides across all strands in all gray species. For consistency, product $Act_{i,j}$ consists of a 7-nt Tj∗, a 9-nt Ai∗, and a 7-nt T∗ in all five examples. All toeholds and long domains contain 7 and 15 nucleotides, respectively, except for Xi in Scheme iv, which is split into two long domains Xia and Xib that each contains 13 nucleotides.

activator is three-stranded and the drain is two-stranded, making the largest number of strands per molecule 3. The bottom strand in the inhibited activator is the longest, containing two toeholds and two long domains besides the activator domains, totaling 67 nucleotides. All three aspects of the design complexity pose room for improvement.

The second scheme utilizes a reversible version of cooperative hybridization[7,8] to implement the "join" function for X_i and L_j using a two-stranded fuel $G_{i,j}$. Upon reacting with the input and label, two waste products are generated simultaneously, each containing an open toehold Tj or

Tj$*$. One waste proceeds to react with $Act^*_{i,j}$ and release the activator, whereas the other becomes irreversibly consumed by drain $D_{i,j}$. Compared to the first scheme, this one has fewer strands per molecule at the cost of more fuel species. The longest strands are in $G_{i,j}$, improved from 67 to 44 nucleotides by moving the activator domains to a sperate fuel species.

The third scheme simplifies the design by utilizing an allosteric toehold.[9] In this scheme, the label consists of two consecutive toeholds, different from the format of the input. Toehold Lj$*$ initiates branch migration to open up toehold T$*$ on $Act^*_{i,j}$, allowing for the input to react and reversibly release the activator. Drain $D_{i,j}$ functions the same as discussed above. Compared to the first scheme, the bottom strand in the inhibited activator is shortened by one toehold and one long domain. While the number of fuel species remains the same, improvements are achieved on the number of strands per molecule and the length of strands.

Reducing the number of fuel species requires embedding irreversibility within the inhibited activator. One way to accomplish that is shown in the fourth scheme: the top strand is broken into two strands such that the reverse reaction initiated by the second strand can not complete branch migration and free up input again once it has become bound to the bottom strand. For both top strands to be stably bound within the inhibited activator, a longer Xi domain is needed. This scheme successfully reduces the number of fuel species to the minimum possible, at the cost of reverting to a three-stranded fuel and increased strand length.

Instead of disrupting branch migration, the reverse reaction could also be inhibited if the toehold is no longer available. In the last scheme, we inserted Tj$*$ domain as a bulge loop into the top strand of the inhibited activator. Once the top strand is released, the Tj$*$ and Tj domains will hybridize to each other and form a hairpin loop, preventing the Tj toehold from initiating the reverse reaction. This scheme is arguably the simplest possible on all three aspects of the design complexity – a single fuel species that contains just two strands, the longer of which has 45 nucleotides, same as the original allosteric toehold design shown in the third scheme. Further reducing the number of strands per fuel species to one is possible, but at the cost of more species, longer strands, and more complex secondary structures.[10]

The three design criteria for evaluating the complexity of a DNA implementation may seem straightforward, but the challenge is to decide how much optimization is necessary. For example, we were initially satisfied with the third scheme that has two double-stranded fuel species, and did not think that it would be necessary to further simplify the implementation. However, even though the implementation worked well as a motif for learning, we encountered various problems when the motif was composed together with other motifs to demonstrate pattern classification using learned memories (Supplementary Note 4). Critically, every time we applied a design revision in a particular fuel species to address a problem that we discovered, it caused new problems involving other fuel species in the system. After extensive investigation, we concluded that the implementation was not simple enough and we must develop another scheme that uses only a single fuel species, which was the last scheme discussed above. Pushing the design complexity to the extreme turned out to be necessary for building a molecular system as complex as the learning DNA neural network reported in this work (Supplementary Note 5).

3.2 Reaction specificity

There is a trade-off between design complexity and reaction specificity – long domains allow for more specificity than toeholds at the cost of increased design complexity.

Let us consider three example implementation schemes for the weight activation reaction $Act_{i,j} \rightleftharpoons W_{i,j}$ (Fig. S8). In the first scheme, the activator $Act_{i,j}$ utilizes a universal toehold T, and the combination of bit information i (which bit in a weight matrix to activate) and class information j (which weight matrix is the bit in) is encoded in a long domain Aij. It reacts with a three-stranded fuel species, inhibited weight $W^*_{i,j}$, and reversibly converts it to an active weight $W_{i,j}$. This scheme has excellent specificity, as any nucleotide mismatch in the Aij domain will slow down branch migration by 10 to 100-fold.[11] However, for a neural network with m memories that each contains n bits, $n \times m$ long domains are needed, posing challenges for the scalability of sequence design given a 15-nt domain length or otherwise requiring longer strands.

In the second scheme, the bit and class information are encoded separately in two toeholds, Ai and Tj, in the activator. The inhibited weight is simplified to a two-stranded complex, and the number of distinct domains is reduced to $n + m$. A trade-off of this scheme is reduced specificity. Even though nucleotide mismatches in the Ai domain will still slow down branch migration, a partial toehold can be exposed for input binding without requiring the completion of branch migration, resulting in crosstalk in weight activation (Supplementary Note 4.8). Additionally, designing m orthogonal toeholds Tj will be challenging when m is large.

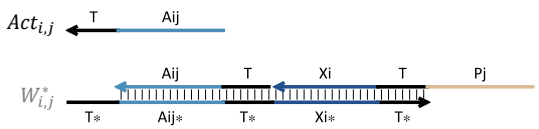
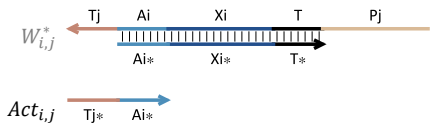
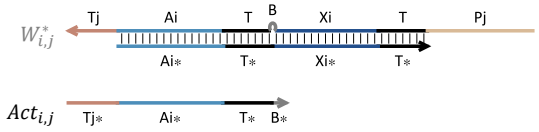
	Implementation scheme for $Act_{i,j} \rightleftharpoons W_{i,j}$	Types of fuel molecules	Strands per molecule	Nucleotides per strand	Number of toeholds	Number of long domains
i		1	3	51	1	$n \times m$
ii		1	2	51	$n + m$	0
iii		1	2	68	$m + 1$	n

Fig. S8 | Reaction specificity. Three example implementation schemes are compared for reaction $Act_{i,j} \rightleftharpoons W_{i,j}$. $W_{i,j}$ is an activated weight molecule that participates in downstream reaction $X_i + W_{i,j} \rightarrow X_i + P_{i,j}$, where X_i contains a 7-nt T (Scheme i and iii) or Ai (Scheme ii) and a 15-nt Xi domain. Species names in black indicate signal species that appear in the formal chemical reaction. Species names in gray indicate fuel species designed to facilitate the desired reaction. Types of fuel molecules is counted as the number of gray species. Strands per molecule is counted as the largest number of strands across all gray species. Nucleotides per strand is counted as the largest number of nucleotides across all strands in all gray species. Number of toeholds and long domains are counted within activator $Act_{i,j}$, where $1 \leq i \leq n$ and $1 \leq j \leq m$.

Considering the desired properties and trade-offs of the above schemes, we established the following criteria for achieving a balance between design complexity and reaction specificity in signals that contain a combination of information (like activator $Act_{i,j}$): **Encode specificity in long domains or toeholds based on the number of distinct domains.** For example, in DNA neural networks with a small number of complex memories ($n = 100$ and $m = 2$), we encode the bit specificity in long domains (Ai) and the class specificity in short toeholds (Tj), as shown in the last scheme in Fig. S8.

3.3 System-level three-letter code

A three-letter code has been commonly used for minimizing secondary structures in DNA strands.[2,12] As G-C base pairs are roughly 5 times stronger than A-T base pairs, by removing G from the sequence alphabet, no G-C base pairs could form within the same strand, effectively reducing undesired secondary structures that affect the kinetics of strand displacement reactions. In addition to secondary structures within individual strands, undesired spurious bindings could occur between single-stranded regions on separate molecules. For example, fuel species have open toeholds for reacting with signal species, and these toeholds have the opposite three-letter code with C removed from the sequence alphabet. Spurious binding between a signal and a mismatching fuel may involve G-C base pairs, but is often not a concern due to the short length of the toeholds.

In this work, the use of allosteric toehold[9] led us to discover limitations of the simple three-letter code applied at the individual strand level. As a solution, we extended the three-letter code to be applied at the system level, defined as follows: **All single strands and single-stranded regions on a double- or multi-stranded species (on the same strand or on different strands but adjacent to each other) must be non-star domains (As, Ts, and Cs only) or otherwise no longer than a toehold size, for all initial species, intermediates, and products in the entire system. Single-stranded regions on wastes can have both star and non-star domains so long as the region of star domains is no longer than a toehold size.**

There are three important features of the system-level three-letter code. First, individual strands within double- or multi-stranded species do not need to have the same three-letter code across all domains – star and non-star domains can occur within the same strand so long as the exposed single-stranded region has only non-star domains or otherwise no longer than a toehold size. We expect that secondary structures within a strand that has both star and non-star domains will not cause a problem when the strands are annealed together to form a complex, if the secondary structures are not too strong to result in kinetics traps during annealing. This rule is necessary to resolve violations of other aspects of the three-letter code, as we will discuss in an example below. Second, beyond the initial species in a system, intermediates and products must be considered, as violations of which will cause the same problems of undesired interactions when the system executes to process molecular information. Wastes have a different rule because self-occlusion within a waste is not an issue and only wastes occluding other species should be considered. Third, a single-stranded region on a double- or multi-stranded species can adopt one of two formats: consecutive domains on the same strand or adjacent domains on different strands. Violations in either format will cause undesired occlusions that reduce the effectiveness of a fuel, intermediate, or product.

As an exercise, let us evaluate six example systems shown in Fig. S9. The first example does not satisfy the three-letter code, because L_j is a single-stranded species that has two consecutive toeholds of star domains, longer than a single toehold. The consequence of this violation is that L_j can spuriously bind to other single strands in the system, causing occlusion for both. Occlusion between L_j and X_i is inevitable due to the complementary toeholds Ai$*$ and Ai needed for allosteric regulation – this occlusion is expected to be highly reversible given the short toehold size. However, spurious binding between L_j and F_i can be stronger than a short toehold, affecting the availability of both species for participating in the desired reactions. Similarly, occlusion could also occur between L_j and the single-stranded domain Pj on $W^*_{i,j}$, affecting the desired function of L_j.

Even if the problem in L_j is not considered, the system still does not satisfy the three-letter code. In the second example, L_j is removed and $Act^*_{i,j}$ is changed to $Act'_{i,j}$, which is a simplified version of

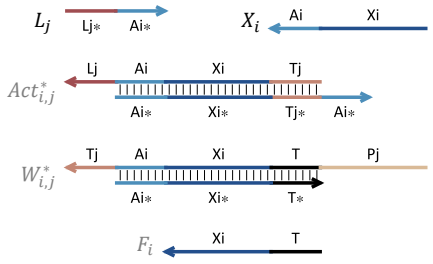
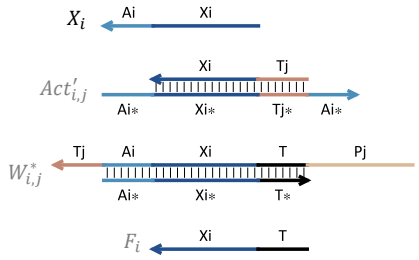
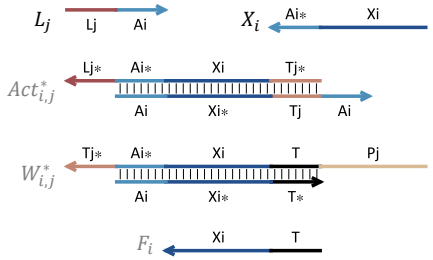
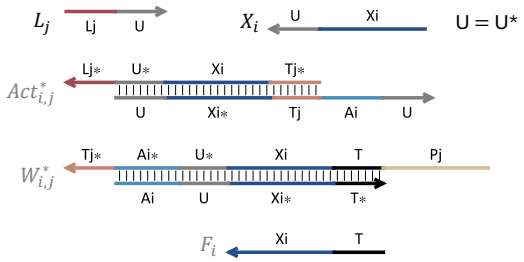
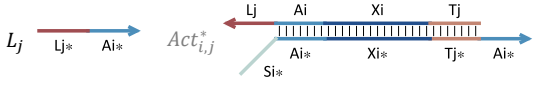
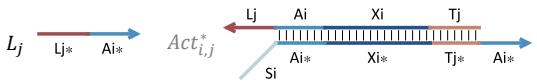
	Example DNA strand-displacement system	System-level 3-letter code	Violations
i		✘	L_j is single stranded and has star domains longer than a toehold size, allowing it to occlude and be occluded by other single strands (for example, F_i) or single-stranded domains on double- or multi-stranded species (for example, Pj domain on $W^*_{i,j}$) in the system.
ii		✘	All initial species satisfy the system-level 3-letter code, but intermediate product from X_i reacting with $Act'_{i,j}$ has single-stranded star domains (Tj* Ai*) longer than a toehold size.
iii		✘	All initial species, intermediates, and products satisfy the system-level 3-letter code except for X_i, which has both star and non-star domains in the same strand.
iv		✓	No violations. Given U = U*, U must be As and Ts only.
v		✘	Lj and Si* domains are adjacent single-stranded non-star and star domains on the same double-stranded species $Act^*_{i,j}$.
vi		✘	Intermediate product from L_j reacting with $Act^*_{i,j}$ has a single-stranded region with both star and non-star domains.

Fig. S9 | System-level three-letter code. Six example DNA strand-displacement systems are evaluated for whether they each satisfies or violates the system-level three-letter code. For consistency, sequences of all non-star domains have As, Ts, and Cs only, whereas sequences of all-star domains have As, Ts, and Gs only. Lj, Ai, Tj, T, U, and Si are toeholds. Xi and Pj are long domains.

the intermediate species in which L_j is bound and exposed the Ai$*$ toehold. In this case, all initial species satisfy the three-letter code. However, once X_i has reacted with $Act'_{i,j}$, the two-stranded product $Act_{i,j}$ will have both Tj$*$ and Ai$*$ domains exposed, making up a single-stranded region of star domains longer than a toehold. The violation in this product may give rise to occlusion by F_i or Pj on $W^*_{i,j}$, preventing it from effectively participating in desired downstream reactions.

To address the problems in L_j and $Act_{i,j}$, we can convert the star domains to non-star domains, as shown in the third example. In this case, all four strands in $Act^*_{i,j}$ and $W^*_{i,j}$ consist of both star and non-star domains, which is not a problem because they will each be annealed to form the desired two-stranded structures. However, a violation in X_i arises – it is a single-stranded species that has both star and non-star domains. This violation may cause strong secondary structure within X_i and slow down its reaction with $Act_{i,j}$.

To address the problem in X_i, we can replace Ai$*$ with a universal toehold U, and make the same change in L_j, as shown in the fourth example. The universal toehold can be designed to be self-complementary (U = U$*$) so that both L_j and X_i satisfy the three-letter code while still being able to react with $Act^*_{i,j}$. The self-complementarity determines that U must only contain common nucleotides between star and non-star domains (As and Ts). In fact, any U domain sequences with As and Ts only will allow L_j and X_i to satisfy the three-letter code, whether self-complementary or not. In that case, U on L_j or X_i will be written as U$*$, with the knowledge that a two-letter code domain can be treated as either a star or non-star domain. The two-letter code should only be applied to universal domains, otherwise sequence design challenges will limit the scalability of the system. The universal toehold alone works for L_j but not $Act_{i,j}$, because the latter must encode both the class j information and the bit i information. This issue can be resolved by an additional Ai domain between Tj and U. Accordingly, a similar change in $W^*_{i,j}$ is needed for it to be activated by $Act_{i,j}$. The fourth example now satisfies all aspects of the system-level three-letter code.

Importantly, there does not always exist a solution to satisfy the system-level three-letter code. For example, if an open toehold is needed at the 5' end of the bottom strand in $Act^*_{i,j}$, as shown in the last two examples, and if it must encode the bit i information, then it violates the three-letter code no matter it is a star or non-star domain. A star domain (Si$*$) adjacent to a non-star domain (Lj) on the same species may result in occlusion of both domains. Changing Si$*$ to Si fixes the above problem but introduces another problem in the intermediate product – when L_j is bound to $Act^*_{i,j}$, both Si and Ai$*$ will be exposed, leading to the possibility of strong secondary structure within the two domains, affecting their ability to participate in downstream reactions.

When the system-level three-letter code cannot be satisfied, more stringent sequence design criteria can be introduced to allow for a valid implementation. However, this would severely limit the scalability of the system, as checking all intermediate states of a complex molecular system becomes unfeasible. Apart from the design complexity issue discussed in Supplementary Note 3.1, the violation of three-letter code is another main reason that prompted us to switch from Design 1 (Supplementary Note 4) to Design 2 (Supplementary Note 5).

4 Design 1

4.1 Activatable weight motif

We began with an activatable weight motif that directly utilized an allosteric toehold.[9] In this design (Fig. S10a), a single-stranded activator $Act_{i,j}$ consists of two toehold domains Tj$*$ and Xit$*$, encoding the memory identity j and input identity i, respectively. Tj$*$ binds to the open toehold Tj on a double-stranded inhibited weight $W^*_{i,j}$, initiating subsequent branch migration to open the toehold Xit$*$ on $W^*_{i,j}$ and converting it to an active weight $W_{i,j}$. Input strand X_i then binds to the open toehold Xit$*$ on $W_{i,j}$ and releases an intermediate product $P_{i,j}$ that represents the output of weight multiplication and the input to downstream summation. Fuel strand XF_i displaces the input, freeing it up for reacting with more active weight molecules and resulting in catalytic behavior where the concentration of $P_{i,j}$ can be higher than the input concentration. The catalytic behavior is necessary for implementing weights that are larger than 1.

Compared to the original weight molecule in our previous work of winner-take-all neural networks,[3] the weight molecule here is extended with a 14-nt double-stranded domain (7-nt Tj and 7-nt Xit), allowing for it to reversibly switches between an inhibited state where Xit$*$ is closed and an active state where Xit$*$ is open, regulated by an activator strand containing an additional toehold Tj$*$. Moreover, unlike the universal toehold T$*$ used in the original weight molecule at both the 5' and 3' ends of the bottom strand, the toehold Xit$*$ here must encode the input identity i so that each activator strand selectively activates a weight molecule that reacts with a specific input and produces a specific output, allowing for any desired information obtained from learning to be stored in the memories represented as a collection of inhibited and activated weight molecules. Encoding input identities in toeholds has tradeoffs, which will be discussed in Supplementary Note 4.8.

We characterized the behavior of the activatable weight motif using a reporter that directly reacts with the intermediate product $P_{i,j}$ (Fig. S10b). The 3' end of the bottom strand in the reporter is modified with a fluorophore (F = ATTO590) whereas the 5' end of the top strand is modified with a quencher (Q = RQ). The quencher absorbs energy from the fluorophore, resulting in low fluorescence signal. When the output of weight multiplication (the intermediate product $P_{i,j}$) is produced, it binds to the toehold T$*$ on the reporter and releases the quencher-modified strand, resulting in increased fluorescence signal. When the inhibited weight, fuel, and reporter are in excess ($W^*_{1,1} = XF_1 = Rep_1 = 2\times$) and the activator concentration ($Act_{1,1} = 0$ to $1\times$) is no larger than the input concentration ($X_1 = 1\times$), we expect the output concentration to be determined by the activator concentration (simulations shown as solid trajectories in Fig. S10b). Fluorescence kinetics data agreed with the simulations within experimental noise (experiments shown as dotted trajectories in Fig. S10b).

However, when we composed together 8 distinct activatable weight molecules with a downstream winner-take-all circuit to implement a neural network with two 4-bit activatable memories, two out of four test patterns failed to be classified correctly (Fig. S10c): the output expected to be on (Y_2 shown in yellow) was barely above the output expected to be off (Y_1 shown in dark green) for the 2nd and 4th tests. Even for the correctly classified patterns, the kinetics was slower than expected (for example, comparing the experiment with simulation for the 3rd test). We hypothesized that the unexpected system behavior was caused by the change in Gibbs free energies for the two toeholds T1 and T2 encoding the memory identities were not sufficiently similar (-8.72 versus -7.40 kcal/mol as shown in Fig. S10c). To address this issue, we redesigned the second toehold to

reduce the difference (−8.72 versus −8.26 kcal/mol as shown in Fig. S10d). With this change in toehold sequence, we observed much improved pattern classification performance in the 4-bit neural network: clear on-off separations were achieved for all four tests and the kinetics of the experimental data semiquantitatively agreed with the simulations (Fig. S10d).

To further verify the kinetics of weight activation across memories and inputs, we performed a set of characterization experiments for each of the inhibited weights used in the 4-bit, 2-memory neural network (Fig. S10e). Minor differences were observed for distinct inputs within the same memory ($Act_{1,1}$ versus $Act_{3,1}$, and $Act_{2,2}$ versus $Act_{3,2}$) and for the same input across distinct memories ($Act_{3,1}$ versus $Act_{3,2}$), within the expected differences due to sequence variations. At this point, we decided that the differences were acceptable and moved on to the next motif.

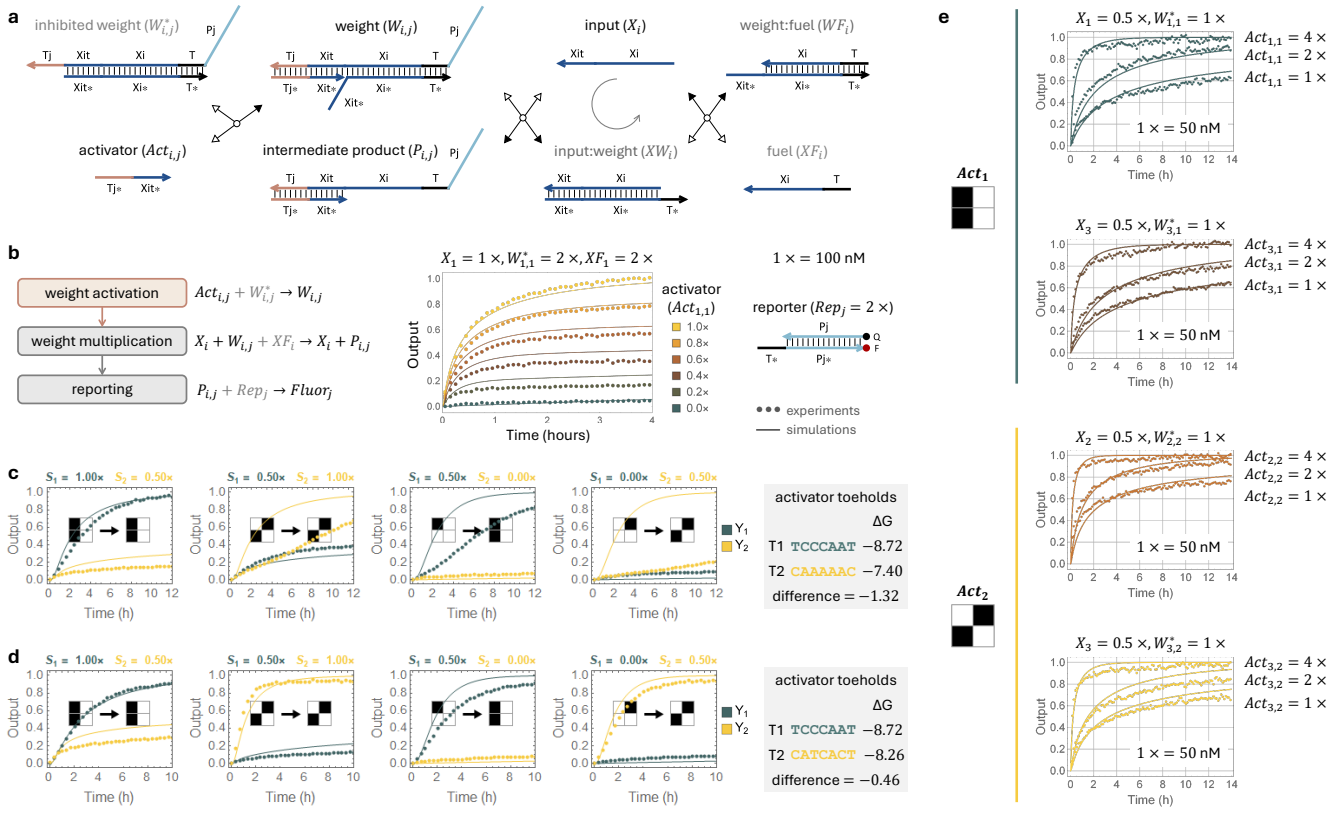


Fig. S10 | Activatable weight motif in Design 1. a, DNA strand-displacement implementation for an early design of an activatable weight motif. In the presence of activators, inhibited weights are activated and input strands along with excess fuel strands can catalytically produce output strands $P_{i,j}$ up to the lessor of the activator or inhibited weight concentration. **b,** Chemical reaction network implementation and experimental data testing the performance of a single inhibited weight with a fluorescence reporter. Species names in black indicate signal or gate species whose concentrations correspond to variable values in the abstract mathematical function. Species names in gray indicate facilitating species whose concentrations are typically in excess. Initial concentrations of all species and measured output concentrations are shown as relative concentrations to a standard concentration of 1× = 100 nM. Fluorescence data are shown in dots and simulations are shown in solid lines. **c-d,** Fluorescence kinetics experiment evaluating a 4-bit activatable neural network using activator toeholds with a larger (**c**) and smaller (**d**) difference in the change in Gibbs free energy (ΔG, unit: kcal/mol). S_1 and S_2 shown on each plot indicate the expected concentrations of the two weighted sum species that determine the position of each input pattern in the weighted sum space and its classification difficulty. **e,** Characterization of each of the inhibited weights used in the two memories of the 4-bit neural network. The rate of output strand produced depends on the concentration of activator strand as well as on the DNA sequences of the Tj and Xit toeholds.

4.2 Learning motif

The learning motif also utilized an allosteric toehold (Fig. S11a). A label strand $L_{i,j}$ consisting of two 7-nt toeholds Lj$*$ and Xit$*$ binds to the open toehold Lj on an inhibited activator $Act^*_{i,j}$, initiating branch migration to expose the toehold Xit$*$ on the inhibited activator, which in turn allows for the input strand X_i to bind and displace the top strand in the inhibited activator, uncovering the toehold Tj$*$ and completing the 14-nt single-stranded region Tj$*$-Xit$*$ for functioning as an activator for the inhibited weight shown in Fig. S10a. The released top strand in the inhibited activator becomes bound to the label strand, which is referred to as the intermediate waste. At this point, the reaction is fully reversible: the intermediate waste can displace the input strand via toehold exchange,[13] and backward branch migration occurs to kick off the label strand. The reversibility is undesired because any unconsumed label and input strands from a previous round of training will interfere with subsequent training events. To ensure irreversibly, a double-stranded drain $D_{i,j}$ was designed to absorb the intermediate waste, producing a three-stranded and a single-stranded waste molecule. Neither waste has any open toeholds to react with other molecules in the system, and thus we considered them to be inert. The drain can be modified with a quencher and a fluorophore to simultaneously function as a reporter that reads out the concentration of the three-stranded waste, which can be used to infer the concentration of the produced activator.

The learning motif (Fig. S11a) and the activatable weight motif (Fig. S10a) are considered two types of an activatable seesaw gate, as illustrated in the learning and weight multiplication layers of the neural network (Fig. 1c). The label strand in the learning motif and the activator strand in the weight motif have the same format (two consecutive toeholds), both functioning as a regulator to the seesaw gate. For the purpose of learning, the input strands are identical in both types of gates, but in general they can be two distinct signals with the same format. The output of the learning gate is in the format of a regulator, whereas the output of the weight gate has the same format as the input (a toehold and a branch migration domain), allowing for distinct functions and composability in a network. The weight gate incorporates a fuel strand for catalysis and signal amplification, which is not needed for the learning gate but in principle could be incorporated when the gate is used in other contexts. Similarly, both gates can be designed to be reversible or irreversible, with or without a drain. The two types of gates can be further generalized into a single type of activatable seesaw gate with two types of possible outputs and allow for the implementation of arbitrary chemical reaction networks (Fig. S12).

We experimentally characterized the behavior of the learning motif (Fig. S11b). Data showed that when the label strand was in excess ($L_{1,1} = 4\times$), the concentration of the produced activator ($Act_{1,1}$) was determined by the input concentration ($X_1 = 0$ to $1\times$). This mechanism ensures that the learned weights, which correspond to the produced activators, represent the total inputs accumulated from all training patterns – a more common bit in the training patterns will result in a larger weight. We also showed that lower amounts of label below the input concentration resulted in proportionally reduced activator concentration, whereas a larger excess of label enabled faster kinetics of activator production while maintaining the same activator concentration set by the input (Fig. S11c). Variation in kinetics across three distinct inhibited activators ($Act^*_{3,1}$, $Act^*_{2,2}$, and $Act^*_{3,2}$) was likely due to sequences, whereas variation in reaction completion was likely due to synthesis errors. Notably, $Act^*_{3,2}$ produced nearly $0.5\times$ of activator when $0.75\times$ of input but no label was present. This leak cannot be explained by any input-specific ($i = 3$) or memory-specific ($j = 2$) domain sequences alone, because neither $Act^*_{3,1}$ nor $Act^*_{2,2}$ had more than $0.1\times$ of leak.

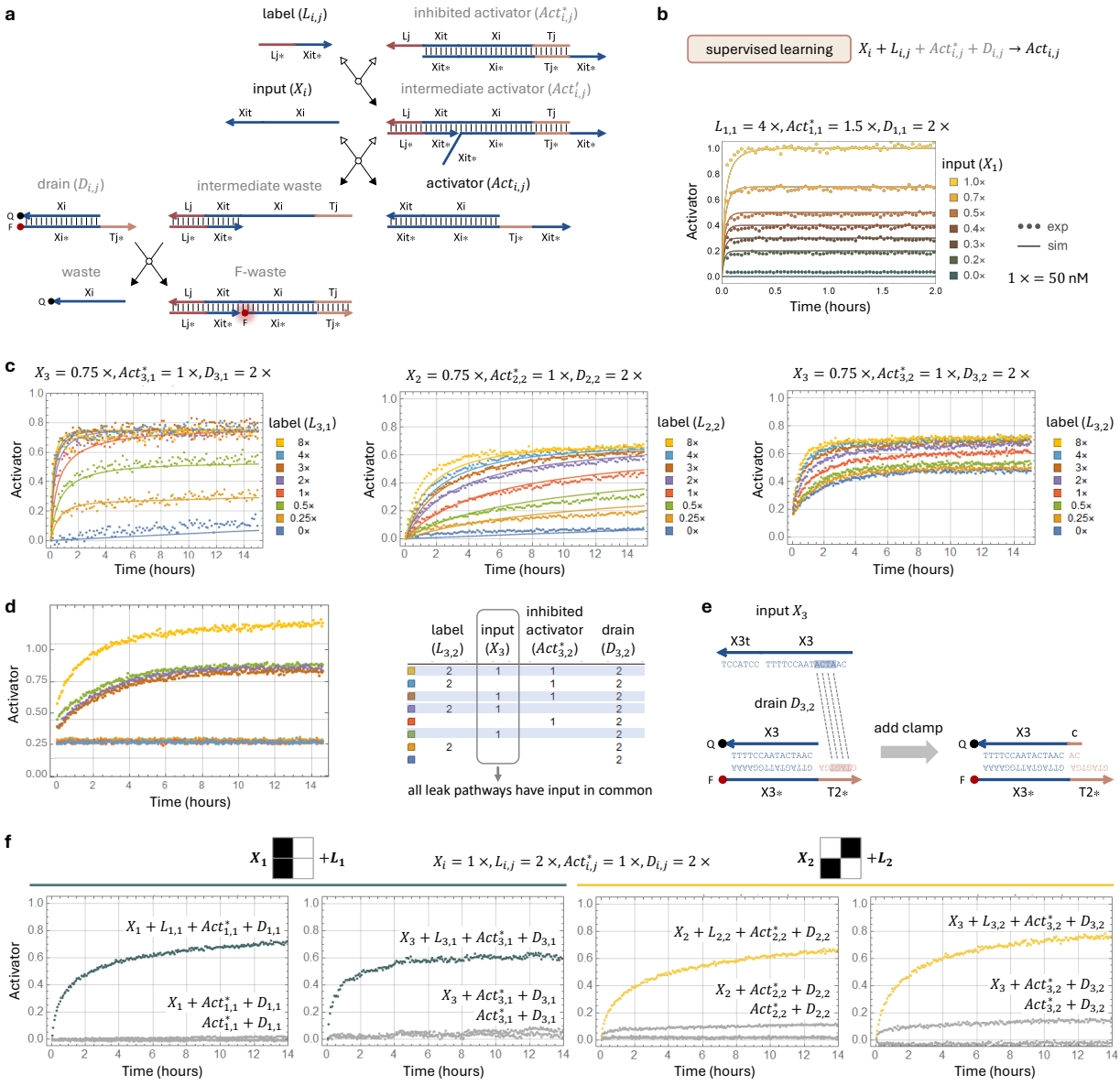


Fig. S11 | Learning motif in Design 1. **a,** DNA strand-displacement implementation for an early design of a learning motif. A drain is used to ensure irreversibility of learning. Optionally, with fluorophore and quencher modifications on the drain, the learning reaction could be observed and the activator concentration could be inferred. **b,** Chemical reaction network implementation of learning, and experimental data testing the performance of a single inhibited activator with a fluorescence drain. Species names in black indicate signal or gate species whose concentrations correspond to variable values in the abstract mathematical function.Species names in gray indicate facilitating species whose concentrations are typically in excess. The analog learned weight corresponds to the input strand concentration. **c,** Fluorescence kinetics experiments evaluating learning reaction performance with variable label strand concentration. Due to sequence variation $Act^*_{3,1}$ exhibited faster kinetics than $Act^*_{2,2}$, and $Act^*_{3,2}$ displayed strong leak when the input was present but the label was absent. **d,** Troubleshooting the strong leak reaction by testing all combinations of learning reactants. All test pathways that show leak are highlighted in blue and have input in common. **e,** Analysis of the input and drain sequences used in the experiments shown in d, revealing a complementary 4-nt region that can serve as a remote toehold to enable branch migration. Avoiding all possible short complementary regions is prohibitive, so another solution is to add a 2-nt clamp. **f,** Diagnostic tests confirming learning reaction performance with the clamped drain for four inhibited activators involved in learning two 4-bit patterns. Input-drain leak was reduced although not fully eliminated.

To investigate the source of the leak, we performed experiments with all combinations of molecules consisting the learning motif (Fig. S11d). The drain $D_{3,2}$ was present in all these experiments because of its role as a reporter for fluorescence readout. The label $L_{3,2}$, input X_3, and inhibited activator $Act^*_{3,2}$ were each either present or absent, resulting in 8 combinations. The data clearly separated into three groups: highest activator concentration was observed when all three components were present, indicating desired reaction; lowest activator concentration was observed when the input was absent, indicating desired background; medium activator concentration was observed when the input was present, regardless of the other two components, indicating undesired leak. Taking a closer look at the sequences of the input and the drain (Fig. S11e), we discovered a 4-nt complementary region between the X3 domain on the input and the open toehold T2$*$ on the drain. This complementary region could function as a remote toehold[14] that speeds up the anticipated toeless strand displacement between the input and the drain.

A possible solution is to introduce more stringent sequence design criteria that eliminate any complementary regions longer than 3 nucleotides between any input X_i and the open toehold Tj$*$ on drain $D_{i,j}$. This criteria would limit the scalability of the system as the number of inputs increases. Alternatively, adding a clamp on the drain could slow down the initiation of branch migration regardless of any complementary regions, providing a simpler solution for mitigating the observed leak (Fig. S11e). Utilizing the revised drain with a clamp, we evaluated the behavior of four inhibited activators involved in learning two 4-bit patterns (Fig. S11f). For $Act^*_{3,2}$, the leak between input X_3 and drain $D_{3,2}$ was reduced to below 0.2$\times$. Even with a clamp, initiation of undesired branch migration could still occur from the 3' end of the X3 domain, explaining the remaining leak. Nonetheless, clear on-off separations of produced activators were observed for all four input and label combinations, establishing the basis for a 4-bit learning system.

4.3 General-purpose implementation of chemical reaction networks

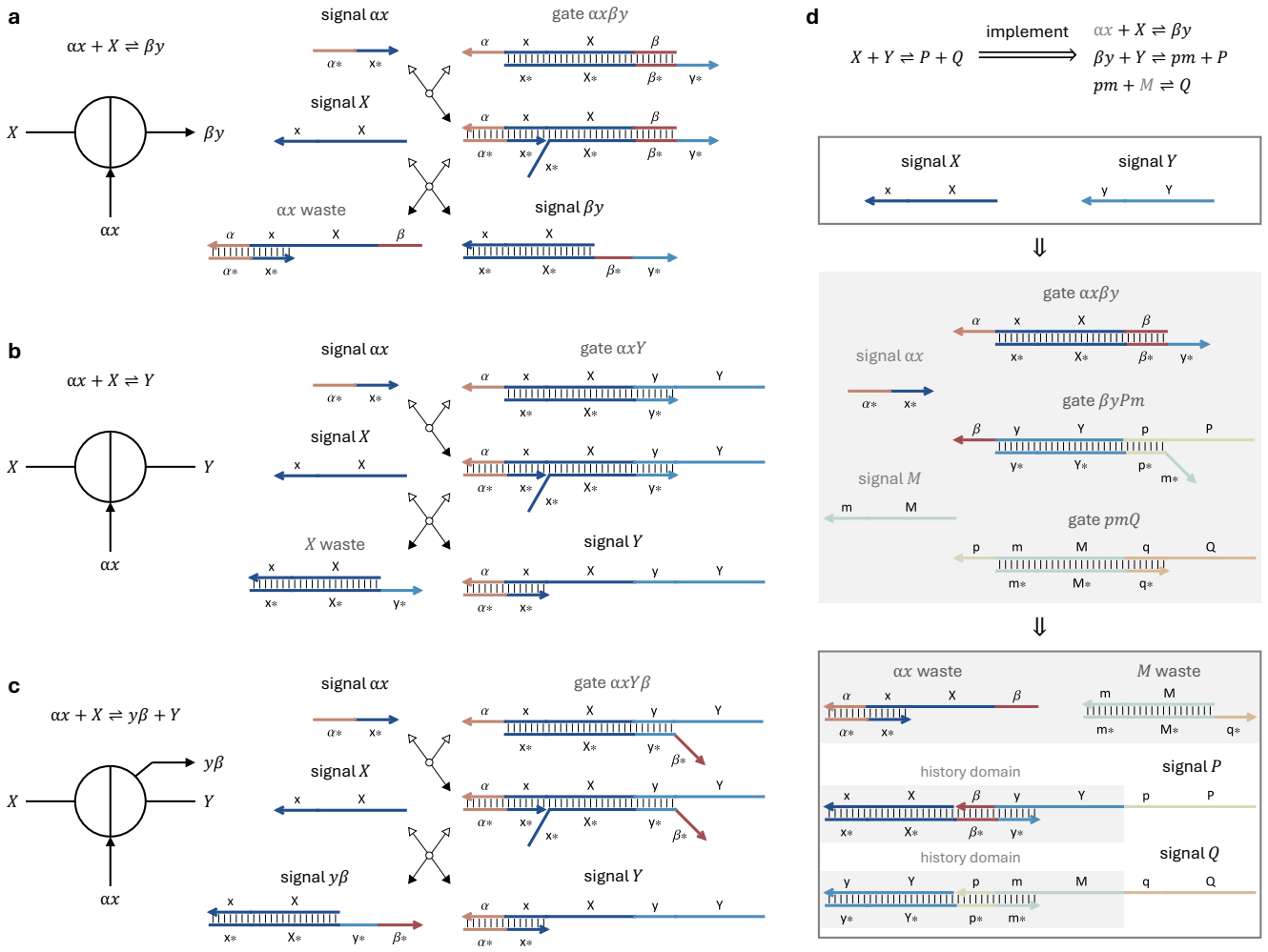


Fig. S12 | General-purpose implementation of chemical reaction networks using activatable seesaw gates. a-b, Activatable seesaw gate generalized from the learning motif (**a**) and the weight motif (**b**). A two-sided node indicates a gate that takes inputs on one side, produces outputs on the other side, and is activated by a regulator that points to the center. A wire indicates a signal, which can be an input, an output, or a regulator. A wire without an arrowhead indicates a standard signal (for example, X) that consists of a toehold and a branch migration domain. A wire with an arrowhead indicates a regulator signal (for example, αx) that consists of two consecutive toeholds, the second of which must be complementary to the toehold on the input for the gate that it regulates. An output can be a regulator (a) or an input (b) to a downstream gate. **c,** Activatable seesaw gate integrating a and b, allowing for two output signals, one of which is a standard signal and the other is a regulator. **d,** Implementation of a two-reactant, two-product chemical reaction $X + Y \rightleftharpoons P + Q$ using the three types of gates shown in a-c. Species in black indicate formal signals or intermediates; species in gray indicate fuels that are specific to this reaction or to a formal signal species in this reaction and held at a constant high concentration. For formal signals, all reactants and products have the same format, except for history domains, but distinct and independent sequences indicated by unique domain names. Reactants X and Y are shown in the top box. Five fuel species in the middle box facilitate the desired reaction, two of which are signals and the other three are gates defined in (a)-(c). Four products are shown in the bottom box, two of which are wastes and the other two are formal signals P and Q with history domains. The implementation of a single reaction is verified to be correct using Peppercorn.[15] The compilation of multiple reactions using Nuskell[16] would fail because history domains on the product species are reaction specific. To address this problem, a pair of reactions is needed to include the fuels and wastes in both directions $X + Y \rightleftharpoons P + Q$ and $P + Q \rightleftharpoons X + Y$. Alternatively, a standard seesaw gate could be used as a translator to remove the history domains.

4.4 Modular drain

To monitor the production of activators during learning, we used drains with fluorophore and quencher modifications (Fig. S11a). For learning in a 100-bit two-memory neural network, we would need 100 unique quencher strands and 200 unique fluorophore strands, which becomes cost prohibitive. We thus investigated a cost-effective approach using a modular drain (Fig. S13a), similar to the previously developed modular hybridization probes.[17,18] Compared to the regular drain $D_{i,j}$, the top and bottom strands of the modular drain $M_{i,j}$ each has an extended region (S6-T and S25∗, respectively) bound to a complementary strand modified with a fluorophore or a quencher. The fluorophore and quencher strands are each taken from a standard reporter (Rep_6 and Rep_{25}, respectively), which remain the same for arbitrary $M_{i,j}$ with distinct Xi and Tj domains. When no signal is present, the fluorophore and quencher are in close proximity, resulting in low fluorescence. When a signal containing Xi and Tj domains (for example, drain trigger V1) arrives, it binds to the open toehold Tj∗ on the modular drain and displaces the top half of the 4-stranded complex, separating the fluorophore from the quencher and resulting in increased fluorescence. While the fluorophore and quencher are covalently linked to the regular drain, the extended domains on the modular drain (colored in gray) function as linkers that non-covalently attach the fluorophore and quencher to the drain, allowing for hybridization of distinct drains that have common linkers but varying signal detection regions.

To evaluate how well the fluorescence readout of the modular drain reflects the signal concentration that it absorbs, we measured the fluorescence for 7 signal concentrations between 0 and 1× and 2× excess drain (Fig. S13a). The experiments included four test cases: i) A simple signal (drain trigger V1) that directly triggers the drain. This is a straightforward test for the performance of the drain. ii) An input used in learning and a simplified intermediate activator (Lj and Xit domains removed from the intermediate activator shown in Fig. S11a), producing the simple signal. This test simultaneously evaluates the second reaction step in learning and introduces a competing reaction for the drain after the signal is produced (reverse reaction of the second step). Because the competing reaction is fully reversible, it is not expected to affect the performance of the irreversible drain. iii) The input, label, and inhibited activator, producing the intermediate waste in two steps (Fig. S11a), which is equivalent to the simple signal but extended with double-stranded Lj and Xit domains. This test incorporates the first and second reaction steps in learning and evaluates the robustness of the drain when the signal has additional components that are expected to be nonreactive. iv) A single-stranded signal including the Lj and Xit domains (drain trigger V2). This test is similar to iii but the learning reaction steps are eliminated, allowing for the robustness evaluation in isolation, with a small variation that the additional component is single rather than double-stranded.

Ideal performance of the drain corresponds to a perfect linear fit of fluorescence versus signal concentration for all 7 data points between 0 and 1×. Near-ideal performance was observed for the regular drain but not for the modular drain (Fig. S13a). To visualize how the slope of the data points changes over the signal range, a gray line was fit to each pair of adjacent data points. For the regular drain $D_{i,j}$, all gray lines had similar slopes. For the modular drain $M_{i,j}$, the slope at low signal concentration was nearly zero and increased with higher concentration. This phenomenon occurred even in the simplest test case, suggesting that the modular drain had a thresholding effect where low concentrations of signal strands were consumed without producing fluorescence. Similar but not worse phenomena was observed in the other three more complex test cases, suggesting

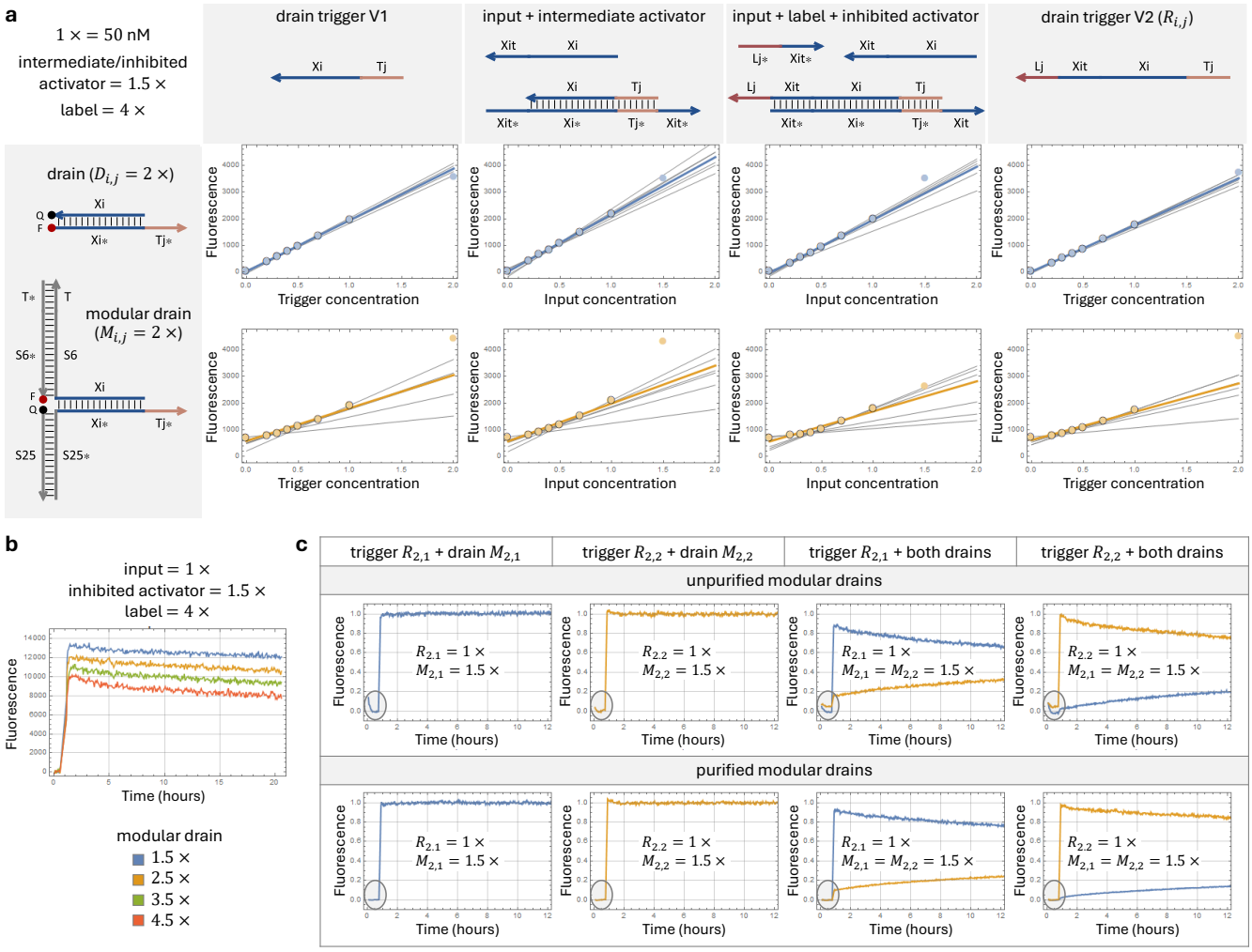


Fig. S13 | Experimental challenges using a modular drain. a, Four different reaction pathways were used to compare the performance of a simple 2-stranded drain and a most cost-effective 4-stranded modular drain. For each reaction pathway, a range of input concentrations were used to establish a fluorescence-concentration calibration curve. These curves should be linear over the tested input range. In all four cases the simple 2-strand drain performed better than the 4-strand modular drain. A colored line (blue or orange) is fit to seven data points outlined with black over the input range 0–1×. The last data point without a black outline is the result of 4× input activation. **b,** Experiments revealing that the concentration of the modular drain was inversely correlated with the amount of signal loss. The modular drain was annealed with excess fluorophore-modified strand, so all samples were normalized to have zero initial fluorescence units in order to compare the signal change upon reaction. **c,** Experiments uncovering that the modular drains can be reversible and non-specific with respect to the Tj toehold sequence. When just a single drain is present, only the desired reaction can occur. However, when both drains are present, the trigger strand can react with both drains. The drains also exhibit undesired reversibly: the fluorescence trends towards a steady state which favors the drain with a matching toehold, but toehold complementarity is not sufficient for complete reaction orthogonality in the presence of both drains.

that the learning reaction performed well and despite the thresholding effect the modular drain was robust enough to handel competing reactions as well as additional components on the signal strand.

To verify that the observation was not due to unexpected molecular defects that led to low effective concentration of the drain or another reactant, an 8th data point was obtained with 4× signal concentration to trigger all of the 2× drain (first and fourth test cases) or all of the 1.5×

intermediate and inhibited activators (second and third test cases). In all cases, the 8th data point was close to or above the best linear fit for the other 7 data points, indicating that the drain was indeed in excess. Unlike the regular drain, the 8th data point for the modular drain was consistency above the best linear fit, agreeing with the thresholding effect discussed above.

To evaluate the cause of the thresholding effect, we repeated the experiments for the third test case but with a fixed 1× input concentration and varying modular drain concentration from 1.5 to 4.5× (Fig. S13b). Lower fluorescence was observed with a higher drain concentration, suggesting that a fraction of the drain was consuming the signal without producing fluorescence, as oppose to a fraction of the signal was unable to react with the drain properly. Naturally, an excess of the bottom strand in the modular drain could function as a threshold for the signal. However, this is unlikely because we intentionally added excess of the top strand: the ratio of the bottom, top, fluorophore, and quencher strands were at $1 : (1 + x) : (1 + 2x) : (1 + x)$, where $x = 0.1$ (Fig. S13a) or 0.2 (Fig. S13b). With these annealing ratios, we expected that the only partial structures were the top:fluorophore complex and the fluorophore and quencher strands. Having the fluorophore strand in excess was desired to ensure no partial 3-stranded complex. However, it could also create spurious bindings to signal strands, because it contains long star domains of sequences with As, Ts, and Gs whereas a signal strand contains long non-star domains of sequences with As, Ts, and Cs. To address this issue, we added an additional quencher strand with the S6 domain (also taken from the standard Rep_6 reporter) at twice the concentration of the excess fluorophore strand. With this change, we expected that the excess fluorophore strand to be bound to the additional quencher strand, resulting in two excess quencher strands that have the same 3-letter code as the signal strands and are generally expected to not interfere with the system behavior. Given that the experimental observation did not agree with our expectation, we decided to remove the excess strands with gel purification.

Furthermore, the fluorescence kinetics exhibited a fast increase followed by a slow decrease, where a higher drain concentration resulted in a larger decrease (Fig. S13b). This type of kinetics is commonly seen in reversible reactions where the forward rate is much faster than the backward rate.[19,20] We hypothesized that the reversibility was due to the fluorophore-quencher interaction[21] and the blunt-end stacking[22] between the 5' end of the double-stranded S6 domain and the 3' end of the double-stranded Xi domain. To reduce this reversibility, we made three changes to the modular drain design. First, a 1-nt bulge loop was introduced near the center of the Xi domain on the drain. When the signal reacts with the drain, elimination of the bulge will help drive the reaction forward, similar to a nucleotide mismatch elimination.[20] Second, a 2-nt dangle was introduced between the fluorophore and the S6∗ domain, weakening the fluorophore-quencher interaction by increasing the distance between them. Third, a 2-nt spacer was introduced between the S6 and Xi domains on the top strand, disrupting the stacking bond.

To evaluate the impact of gel purification and the above design changes, we performed an additional set of experiments with two modular drains that have the same Xi domain but different Tj toeholds (Fig. S13c). For both pairs of trigger and drain ($R_{2,1} + M_{2,1}$ and $R_{2,2} + M_{2,2}$), nearly no slow fluorescence decrease was observed, indicating improved reversibility. However, when each trigger was allowed to react with both drains, the reversibility issue reappeared, accompanied by undesired crosstalk between the trigger and the drain with a mismatched toehold. We suspect this was due to nucleotide truncations on the 5' end of the top strand in the modular drain, creating a small toehold for signal invasion even without the Tj toehold. The drain with a mismatched toehold could then slowly drive the reverse reaction of the trigger and its matching drain.

Another phonomania was observed during the first hour of the experiments before any trigger was added (highlighted in the circled region in Fig. S13c): the fluorescence decreased without any reaction taking place, presumably due to the excess top:fluorophore complex sticking to the walls of wells in the plate, reducing the concentration of fluorescent molecules within the light path of the plate reader. Encouragingly, gel purification of the modular drains removed the initial fluorescence decrease, suggesting successful removal of the excess top:fluorophore complex. Presumably, the excess quencher strands were removed as well. With purified drains, the reversibility and crosstalk were improved, but not enough to meet our standard for a robust building block that can be utilized in complex systems.

The lesson we learned here was that the modular drain has trade-offs. While providing a cost-effective approach for fluorescence readout, it is not precise enough for accurately measuring signals at low concentrations. Moreover, it is not robust enough for avoiding crosstalk in complex systems. The accuracy issue could be explained by the number of strands in the modular drain: four strands are involved in the target complex, and as discussed above one more strand is required to inhibit excess fluorophore strand if gel purification of the complex is not performed. By contrast, only two strands are involved in the regular drain, and excess top strand can be present without causing problems. The robustness issue could be explained by the length of strands in the modular drain: both the top and bottom strands are roughly twice the length of those in the regular drain, leading to more synthesis errors. In particular, DNA synthesis starts from the 3' end of a strand, and thus truncations at the 5' end become worse with increasing strand length. As discussed above, these truncations on the top strand in the modular drain could result in undesired crosstalk that allows a signal with matching Xi domain but mismatching Tj domain to react. Overall, the challenges with a modular drain once again confirmed our design criteria: the shorter strands, and the fewer strands per complex, the better.

4.5 Unreacted drains occluding weight activation

After concluding that the modular drain was not well suited for irreversibly driving the learning reaction forward while simultaneously reading out the concentration of learned weights (Supplementary Note 4.4), we returned to the regular two-stranded drain and made a compromise to temporally skip the readout of leaning and go straight to the integration of learning and testing using "silent drains" that do not have fluorophore and quencher modifications. In this case, the result of learning is indirectly reflected in the result of testing: if the activators produced from learning (Fig. S11a) have desired concentrations that represent the averaged training patterns for each memory, they will collectively activate a desired selection of weight molecules encoding the learned memories, which then react with the input strands from a test pattern to perform weight multiplication (Fig. S10a). Subsequent summation and winner-take-all computation will turn on one of the two outputs read by a standard fluorescent reporter (Fig. 1c), indicating the classification decision.

Unfortunately, our first attempt of learning and testing with 4-bit patterns failed (Fig. S14a). We expected output Y_1 to turn on for the first and third tests, and output Y_2 to turn on for the second and fourth tests (simulations shown as solid trajectories). Experiments suggested that the rate of output production was very slow – both outputs essentially remained off for the second and fourth tests, whereas slightly larger separations between the two outputs were observed for the first and third tests (experiments shown as dotted trajectories).

To investigate the cause of the slow output production rate, we performed a set of experiments comparing the full learning and testing system with a subset of the molecules involved using the third test pattern $\boldsymbol{X} = \{1, 0, 0, 0\}$ that was supposed to turn on Y_1 (Fig. S14b). When only the required weights (two per memory) and their activators were present (no training, minimal testing), the network performed the best, showing desired kinetics of output production similar to the simulation prediction. When all possible weights (four per memory) and the required activators were present (no training, full testing), the kinetics was slightly slower – this was expected because activators can be temporally occluded when they bind to inhibited weights that have a matching Tj toehold but mismatching Xit toehold. When inhibited activators for the unrequired weights were included (inhibited activators, full testing), the kinetics became faster, presumably due to leak between inhibited activators and inhibited weights (Supplementary Note xx). However, the output production turned off when drains from learning were included (drains, full testing), at a level similar to when all learning components were included (full training, full testing).

The above observation revealed a problem for the integration of leaning and testing: drains must be in excess to effectively drive the learning reaction forward, and drains for all possible input and label combinations must be present to enable the learning of arbitrary patterns. Thus, a large fraction of unreacted drains will be left over after learning, affecting the performance of testing. To gain a better understanding of how the drains affect testing, we performed a set of experiments on weight activation using a specific weight and three distinct drains (Fig. S14c). A drain for a matching input i but mismatching memory j had nearly no impact on the activation of weight $W_{i,j}$ (top plot). A drain for both matching input and memory most strongly suppressed weight activation (middle plot). A drain for a mismatching input but matching memory was almost as problematic as the fully matched drain (bottom plot). Matching memory means that the Tj$*$ toehold on the drain is complementary to the Tj toehold on the inhibited weight. To evaluate the impact of this toehold, we measured the kinetics of weight activation comparing three distinct toehold lengths from 7 to 5 nucleotides (Fig. S14d). Reducing the toehold to 6 nucleotides allowed

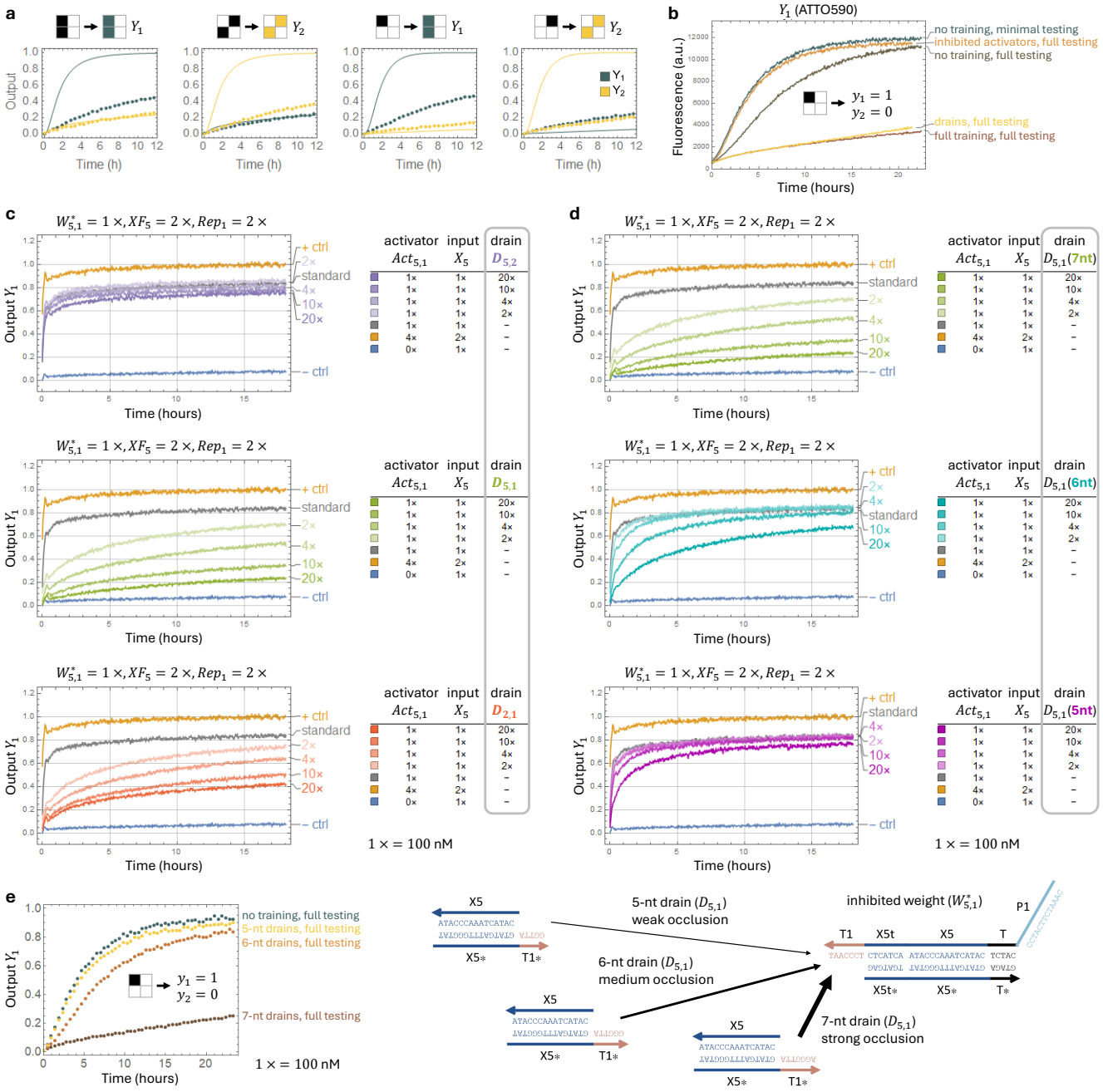


Fig. S14 | Evaluating the effect of unreacted drains occluding weight activation. a, Fluorescence kinetics experiments of 4-bit pattern classification after training. Solid and dotted trajectories indicate simulations and experiments, respectively. Patterns to the left and right of an arrow indicate input pattern and output classification, respectively. **b,** Comparing the full network with a subset of molecules involved in learning and testing, using the third test pattern in (a). Output trajectories are the fluorescent response of signal Y_1 only, which was expected to go on for all cases. **c,** Experiments on weight activation using an inhibited weight $W^*_{5,1}$ and three distinct drains $D_{5,2}$, $D_{5,1}$, and $D_{2,1}$. A standard 1× activation of the inhibited weight is shown in gray. No activator is used as a negative control shown in blue. A positive control of excess activator and input is shown in orange. **d,** Experiments on weight activation comparing three distinct toehold lengths from 7 to 5 nucleotides. The top plot in (d) is the same as the middle plot in (c). **e,** Experiments on 4-bit pattern classification in the presence of drains with varying toehold lengths, using the third test pattern in (a), and sequence-level diagrams of three drains and the inhibited weight that they occlude. The drains were at 8×. All experiments were performed with 1× = 100 nM.

for the recovery of weight activation kinetics at low (2 and 4×) drain concentrations (middle plot). Further reducing the toehold to 5 nucleotides allowed for a similar recovery but even at high (10×) drain concentrations (bottom plot). High drain concentrations are required for learning interesting patterns. For example, after learning 100-bit patterns with 20 bits of 1s, the total unreacted drains will be at 9× for each memory. We thus concluded that a 5-nt toehold on the drain was most effective for maintaining the desired weight activation kinetics.

Finally, we re-evaluated the testing performance of the 4-bit 2-memory neural network using drains with varying toehold lengths (Fig. S14e). The two cases without any drains and with drains that have 7-nt toeholds (abbreviated as "7-nt drains") were the same as those shown in Fig. S14b – they represent the desired and undesired behaviors, respectively. When 5-nt drains were present, the testing performance was nearly as as good as no drains, indicating that the occlusion of weight activation was minimal (termed "weak occlusion"). When 6-nt drains were present, the half reaction completion time increased from 4 to 7 hours, indicating a 1.75-fold slowdown in kinetics as a result of the toehold occlusion on the inhibited weight molecule – we term it as "medium occlusion" compared to an over 10-fold slowdown in kinetics caused by the "strong occlusion" of the 7-nt drains. The kinetics difference may not immediately make sense when the toehold sequences are considered: for example, the 6-nt drain $D_{5,1}$ has a reduction of an A-T base pair compared to the 7-nt drain, whereas the 5-nt drain has a reduction of a G-C base pair compared to the 6-nt drain. From these reductions alone, we would expect the difference between the 6-nt and 7-nt drains to be smaller than that between the 5-nt and 6-nt drains. However, a key issue here is that the binding between the 7-nt drain and the inhibited weight not only result in a double-stranded Tj toehold but also two additional stack bonds at the interface with the Xi domain on the drain and the Xit domain on the inhibited weight. This is why the observed occlusion was so strong despite that we normally do not expect a 7-nt toehold occlusion to be disruptive for the overall system behavior. Shortening the toehold from 7 to 6 nucleotides on the drain simultaneously eliminates one of the stacking bonds, allowing for an effective improvement of the occlusion.

Does using a shorter toehold on the drains provide a good solution to the problem of integrating learning and testing? Not necessarily. We will move on to discover another major problem in the next section (Supplementary Note 4.6).

4.6 Undesired reversibility of learning

An immediate problem that we encountered with the 5-nt drains shown in Fig. S14e was the undesired reversibility of learning. In a single-bit sequential learning experiment (Fig. S15a), a pair of inhibited activators $Act^*_{5,1}$ and $Act^*_{5,2}$ and a pair of drains $D_{5,1}$ and $D_{5,1}$ were present in the test tube. The designed function is that label $L_{5,1}$ enables the single-bit training pattern, input X_5, to be stored in memory 1 by selectively producing $Act_{5,1}$ that will later activator weight $W_{5,1}$ during testing. After the first round of leaning is completed, inhibitor $Inh_{5,1}$ binds to and cleans up any excess label. A new label $L_{5,2}$ could then enable the same single-bit training pattern to be stored in memory 2. Experiments showed that input and the first label collectively produced the desired activator (blue trajectory turning on within the first 2 hours). However, when the inhibitor was added, the fluorescence signal started to decrease, indicating reduced activator concentration. Moreover, when the new label was added without any input, undesired production of activator $Act_{5,2}$ was observed (orange trajectory turning on after 4.5 hours), suggesting nonspecific learning in the absence of any training patterns.

We hypothesized that the reserve production of $Act_{5,1}$ and spurious production of $Act_{5,2}$ were due to the reversibility of the 5-nt drains. It is known that fluorophore-quencher interaction increases the free energy of DNA hybridization, observed as increased melting temperature of fluorophore and quencher modified double strands when the modifications are adjacent to each other.[21] Typically, this interaction is not a problem in standard fluorescent reporters when they are used to monitor reactions that are largely irreversible. When used for the readout of reversible reactions, a sufficiently strong toehold on the reporter or a sufficiently high concentration of the reporter would allow for minimal impact of the fluorophore-quencher interaction. A strong toehold results in a large difference in the forward and reverse rate constants, whereas a high reporter concentration creates a large difference in the reactant and product concentrations, both of which can effectively drive the reaction forward. In our case, the drains with fluorophore and quencher modifications had relatively weak 5-nt toeholds (for example, two G's in the T1$*$ toehold on $D_{5,1}$, as shown in Fig. S14e), and for cost-saving purposes, were in moderate (2×) rather than large excess.

To verify the reversibly of the drains, we measured the kinetics of toeless strand displacement initiated by the fluorophore-quencher interaction (Fig. S15b). By comparing simulations with experimental data, we estimated a rate constant of $10^{3.5}$ /M/s. The rate constant of strand displacement initiated by the 5-nt toehold T1$*$ can be estimated as $10^{4.5}$ /M/s based on the toehold sequence, resulting in only a 10-fold difference in the forward and reverse rates of the drain $D_{5,1}$. Reversibly of the drains directly lead to the undesired reversibility of learning: the intermediate waste shown in Fig. S11a will be present at a low concentration (for example, 0.1×), reacting with the activator and reverting it to intermediate activator while releasing the input strand. The intermediate activator is not in a stable state and can quickly revert to release the label strand through unimolecular branch migration. When the label inhibitor is present, it will consume any released label strand and drive the learning reaction backward. The released input strand then become available to interfere with subsequent training events – this is why we observed nonspecific learning in the absence of any training patterns when the second label was added (Fig. S15a).

Two caveats should be considered in estimating the strength of fluorophore-quencher interaction. First, the unmodified strand used in the experiment shown in Fig. S15b was unpurified and expected to have more truncations compared to the HPLC-purified quencher strand. We annealed the two-stranded reactant with a 20% excess of the unmodified top strand, hoping that copies of

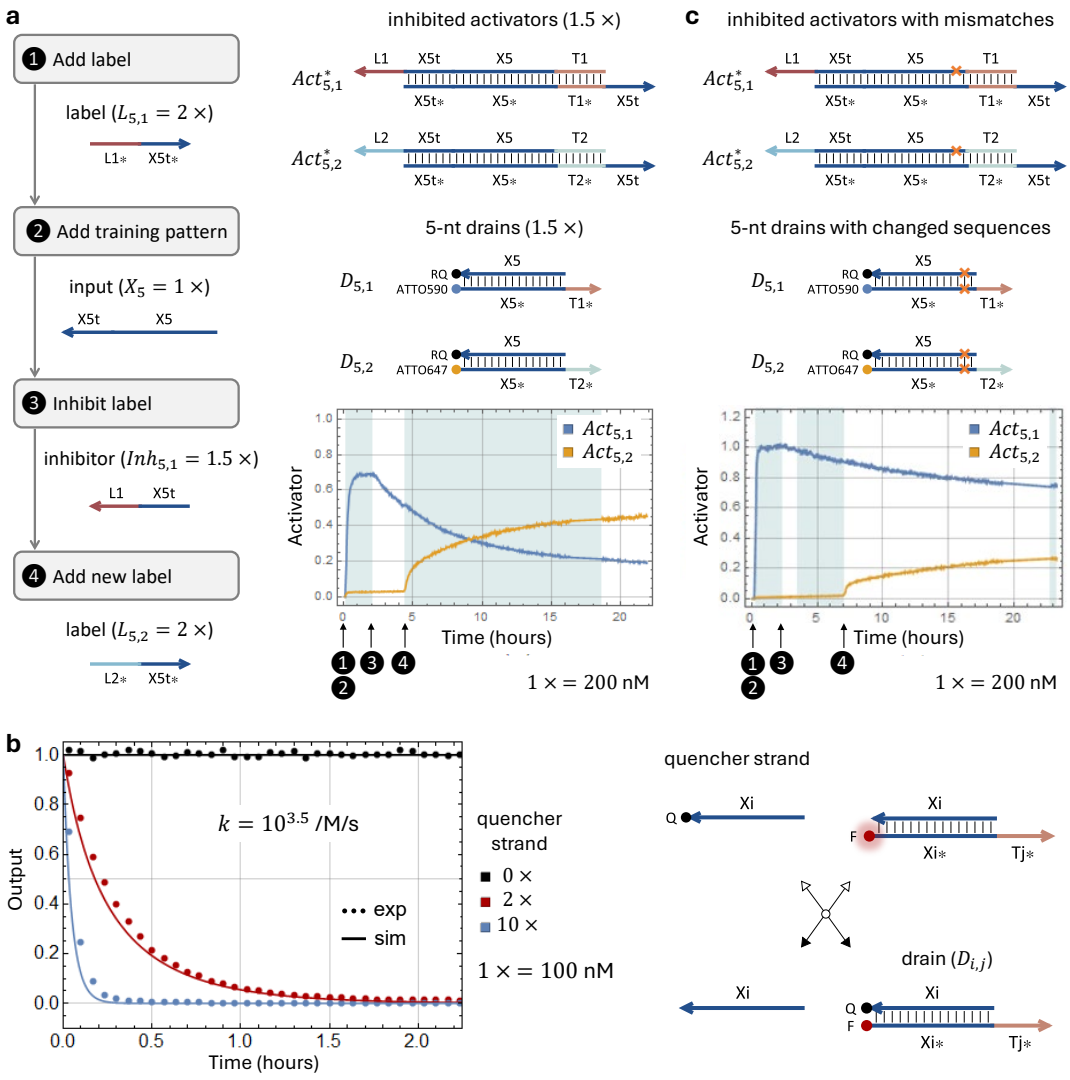


Fig. S15 | Undesired reversibility of learning. a, Fluorescence kinetics experiments of training with a single-bit pattern that can be stored in two memories. Fluorophore- and quencher-modified drains are used to measure the concentrations of activators produced during learning. Toehold T1∗ and T2∗ on the drains are complementary to the 5 nucleotides from the 3' end of the 7-nt toehold T1 and T2 on the inhibited activators. The desired learning behavior at a lower equilibrium is observed when label and training pattern are added in steps 1 and 2 at $t = 0$ hour. In step 3 inhibitor is added at $t = 2$ hours and a reverse effect of learning is observed. Upon addition of a new label in step 4 at $t = 4.5$ hours, fluorescence is produced from $D_{5,2}$ presumably because input strand X_5, which was added and consumed in step 2 but reversibly recovered in step 3, can participate in the second round of learning along with the new label. **b,** Measuring the rate of fluorophore-quencher interaction-initiated strand displacement. A general scheme for the rate measurement is illustrated (right). For the specific experimental data (left), F = ATTO590 and Q = RQ. Addition of a quencher strand shows a decrease in fluorescence, indicating strand displacement restoring the two-stranded drain without the aid of a complementary toehold sequence. Mass-action simulations indicate the fluorophore-quencher interaction functions similar to a toehold of 3.5 nucleotides. **c,** Same experiments as in (a) but using a revised design that includes a mismatched base pair in the inhibited activators. The mismatch is located at the second nucleotide position from the 5' end of the X5 domain, indicated as an orange cross and a missing vertical line. The same nucleotide sequence change was applied to the drains, with a complementary nucleotide on the bottom strand, indicated as a pair of orange crosses with a vertical line in between. Mismatch elimination shifts the reaction equilibrium to favor products of learning. Drain reversibility is improved but not eliminated.

the top strand without truncations will be preferred for binding to the bottom strand at a higher temperature or otherwise displace copies of the top strand with truncations at a lower temperature. Nonetheless, we expect the estimated strand displacement rate to be higher than what is purely enabled by fluorophore-quencher interaction without any truncations involved. Second, we expect the rate to vary across distinct fluorophore-quencher pairs. For example, the change in melting temperatures of fluorophore and quencher modified double strands exhibited a 2-fold difference across 5 distinct fluorophores.[21]

To address the reversibility problem, we explored using nucleotide mismatch elimination as a hidden thermodynamic drive[20] for the learning reaction (Fig. S15c). In this design, a mismatch is introduced in the top strand of the inhibited activators near the 5' end of the Xi domain. As a result, the reverse reaction requires a mismatch creation near the beginning of branch migration, which was shown to have the strongest impact on reducing strand displacement rate compared to all possible mismatch positions.[19,23] To avoid any negative impact on the drain, the same mismatch nucleotide sequence was used in its top strand whereas the complementary nucleotide sequence was used in its bottom strand, making sure that no mismatches exist in the drain and the intermediate waste can react with the drain without any undesired slowdown. Note that further utilizing a mismatch elimination in the drain is not a good option here, as the 15-nt Xi domain is not long enough for the thermodynamic drive to be sufficiently hidden and to prevent undesired leak between the input and the drain. With the mismatch design, we observed a higher reaction completion for the desired learning behavior (blue trajectory reaching 1× within the first 2 hours), reduced signal decrease after the inhibitor was added (blue trajectory after 2 hours), as well as reduced nonspecific learning (orange trajectory after 7 hours).

Does introducing a mismatch on the inhibited activators provide a reasonable solution to the problem of undesired reversibility in learning and allow for a better integration between learning and testing? Not necessarily. We will move on to discover yet another major problem in the next section (Supplementary Note 4.7).

4.7 Leak at the interface of learning and testing

Like there are unreacted drains from leaning, there are also unreacted inhibited activators from learning. The introduction of a mismatch on the inhibited activators (Fig. S15c) led to serious leak in testing (Fig. S16a), where the learned memories were obscured by background noise. Critically, the Tj$*$ toehold on the activator strand is covered up in the inhibited activator, preventing it from binding to the inhibited weight and faultily activate a bit in the memory without going through the proper training process. This toehold inhibition is not perfect, because blunt-end stacking[22] can provide the energy nearly as strong as a base pair to initiate toeless strand displacement. The rate of stacking initiated leak between double-stranded gates was measured to be 20 per molar per second in seesaw logic circuits.[2] However, unlike the gate-gate leak in the original seesaw circuits, the reserve reaction here is a fast unimolecular branch migration, and thus we anticipated the issue of leak to be less problematic.

Unfortunately, the mismatch on the inhibited activator altered the reserve rate of the leak reaction (Fig. S16a): there is only one base pair adjacent to the mismatch in the leak product, which will be highly unstable. When that base pair is open, the reverse rate of branch migration will be slowed down, similar to the effect of a remote toehold.[14] Moreover, because the goal of training is for the DNA neural network to recognize similar molecular events in future operation, the format of the input strands used in test patterns is the same as that in training patterns – input X_i present for testing has a chance to invade the leak product at the mismatch location (indicated by the gray arrow), converting the inhibited activator to an activator and essentially locking in the error.

We experimentally measured the leak with varying inhibited activator concentrations (Fig. S16b). Four distinct weights ($W^*_{i,j}$) that correspond to two bits ($i = 2$ and 8) in two memories ($j = 1$ and 2) were compared with each other. The results can be explained by two factors: First, a stronger stacking energy between the base pair at the 5' end of the Tj domain on the inhibited activator and that at the 3' end of the Xit domain on the inhibited weight (base pair sequences highlighted in orange) initiates a faster forward reaction for the undesired interaction between the two molecules and makes the leak worse. Second, a weaker stacking energy between the two adjacent base pairs to the 3' side of the mismatch (base pair sequences highlighted in gray) encourages more breathing in the leak product shown in Fig. S16a, which in turn creates a larger gap for initiating reverse branch migration, slowing the backward reaction and makes the leak worse. Inhibited activator $Act^*_{8,1}$ had the weakest first energy among the four Tj and Xit combinations and the stronger second energy among the two Xi domains, leading to the least leak (bottom left plot). Conversely, $Act^*_{2,2}$ had the strongest first energy and the weaker second energy, leading to the worst leak (top right plot). $Act^*_{2,1}$ and $Act^*_{8,2}$ had energies in between. If only the first energy is considered, $Act^*_{8,2}$ would have worse leak than $Act^*_{2,1}$. However, the second energy suggested the opposite, which agreed with the experimental observation (top left and bottom right plots), possibly because the difference in the second energy was larger than the first.

Rather than shortening the toehold in the drain to address the problem of unreacted drains occluding weight activation (Supplementary Note 4.5), an alternative solution is to introduce a wobble base pair in that toehold so that occlusion becomes weaker when it binds to the open Tj toehold on the inhibited weight. This change would require the inhibited activator to adopt a wobble base pair at the same position in the Tj toehold so that the intermediate waste has a fully complementary sequence to the altered drain toehold for effectively driving the learning reaction

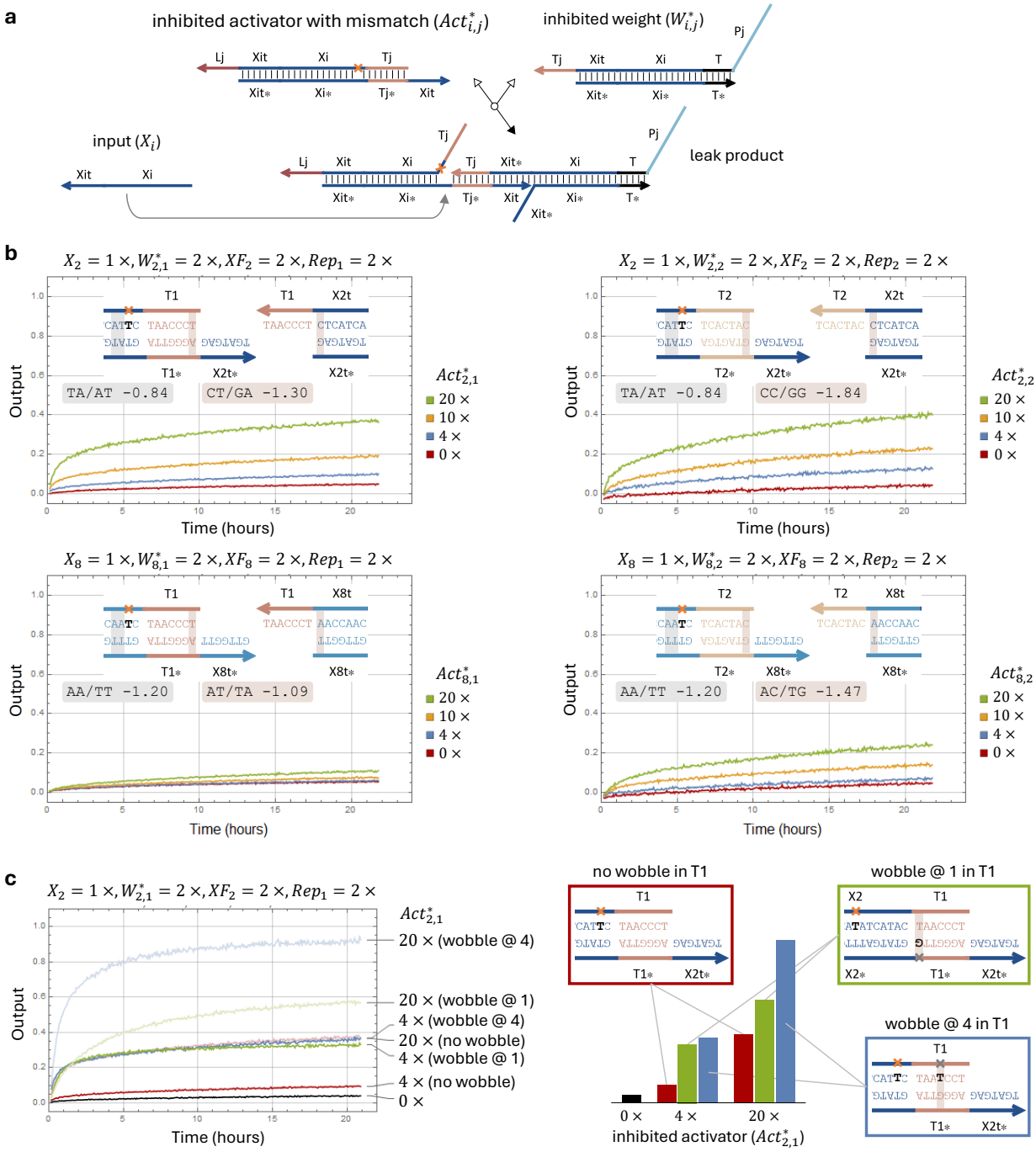


Fig. S16 | Leak at the interface of learning and testing. a, Mechanism of leak between an inhibited activator with a mismatch and an inhibited weight. The mismatch is located at the second nucleotide position from the 5' end of the Xi domain, indicated as an orange cross and a missing vertical line. **b,** Fluorescence kinetics experiments testing the leak with varying inhibited activator concentrations from 0 to 20×, where 1× = 100 nM. Four different pairs of $Act^*_{i,j}$ and W^*i,j were chosen to have different base stacking sequences, illustrating highly variable leak dependent on the identity of the Tj toehold and the Xit and Xi sequences. The unit of base stacking free energies is kcal per mol, and the values are taken from Huguet et al.[24] **c,** Experiments testing the change of leak when a wobble base pair was introduced into the Tj domain of the inhibited activator complex. Inhibited activators with no wobble base pair, a wobble at position 1, and a wobble at position 4 were compared with each other. End point fluorescence at 20 hours is shown in a bar chart with sequence-level diagrams of the relevant parts of the inhibited activators.

forward, whereas the activator strand has the original Tj$*$ sequence for effective weight activation. However, the alternative solution made the leak between the inhibited activator and weight even worse (Fig. S16c). A wobble at position 4 of the T1 toehold, counting from the 3' end, resulted in more than twice the amount of leak (blue versus red bars in the bar chart). The leak level was approaching the level of desired weight activation when the inhibited activator was at 20×. Like the unreacted drains, a large excess of unreacted inhibited activators is necessary for achieving interesting and diverse training tasks. Therefore, the observed leak was not acceptable. Naturally, we considered moving the mismatch in the Xi domain to a more distant position away from the wobble to reduce the collective effect of the two (sequence-level diagram highlighted in the green box in Fig. S16c). The wobble itself was moved to position 1 to reduce its impact in favoring the leak product. These changes resulted in some improvement, but still too much leak to merit a valid design choice.

At this point, we concluded that despite the simplicity of the weight activation and learning motifs, we were unable to achieve robust integration of the two for the purpose of demonstrating molecular pattern classification using learned weights. Minor design changes were insufficient to address the major issues identified in occlusion (Supplementary Note 4.5), reversibility (Supplementary Note 4.6), and leak (Supplementary Note 4.7). We will discuss another issue in the crosstalk of weight activation (Supplementary Note 4.8) in the next section before moving on to redesign the motifs based on what we have learned so far.

Importantly, the implementation of chemical reaction networks using these two motifs (Fig. S12) is generally not affected by these issues. Occlusion of a downstream gate caused by unreacted drains from an upstream reaction is a specific issue for adding irreversibility to the reversible scheme shown in Fig. S12. When drains are used, undesired reversibility comes from fluorophore-quencher interaction, which will not be present unless the product of that reaction needs to be monitored. Leak between an upstream and downstream gate is highly reversible, unless a mismatch is employed in the upstream gate to address the reversibility issue. The input for a downstream gate locking in the leak caused by an upstream gate with mismatch is also a very specific case when both gates share the same input. Overall, the proposed implementation scheme remains valid for future experimental investigation.

4.8 Crosstalk in weight activation

In the activatable weight motif (Fig. S10a), each activator $Act_{i,j}$ produced from learning is designed to activate a specific weight $W_{i,j}$. The beauty of the allosteric toehold mechanism[9] is how compact the extension on a gate is for achieving additional control that turns the gate on and off via the presence and absence of a short signal strand, the activator. In our case, the two toehold domains Tj$*$ and Xit$*$ composing the activator are used to encode memory identity j and input identity i, respectively. Given two memories, only two distinct Tj sequences are needed, which is straightforward to design. However, 100-bit input patterns require 100 unique Xit sequences. Using a 3-letter code[2,12] for minimizing secondary structures within each signal strand and reducing spurious interactions between signals, a 7-nt Xit domain has $3^7 = 2,187$ choices of sequences. This may seem like a large enough space for designing 100 distinct toeholds. However, consider the first 4 nucleotides at the 3' end of Xit, there are only $3^4 = 81$ choices, suggesting that some Xit domains must share the same sequences within that region. The shared sequence will lead to partial activation and undesired crosstalk between weights and mismatching activators (Fig. S17a).

Experiments with four weight and activator pairs confirmed the crosstalk issue (Fig. S17b). When the two weights $W^*_{1,1}$ and $W^*_{2,1}$ were paired with matching activators $Act_{1,1}$ and $Act_{2,1}$, respectively, they turned on within 2 hours (top left and bottom right plots). When the activators, whose sequences have 4 nucleotides in common, were swapped, the weights turned half on within 12 hours (top right and bottom left plots). The kinetics of the experimental data (dotted trajectories) agreed with the simulations (solid trajectories), suggesting that the crosstalk was well predicted based on the sequence similarity of the X1t and X2t domains. Experiments further revealed that the exact crosstalk kinetics differed slightly between $W^*_{1,1} + Act_{2,1}$ and $W^*_{2,1} + Act_{1,1}$, presumably due to the difference in the remaining 3 nucleotides of the Xit toeholds as well as the branch migration domain Xi sequences. Unsurprisingly, the weight that had a slightly faster kinetics with its matching activator also had a slightly worse crosstalk (red trajectories in Fig. S17b).

Despite the slight asymmetry in the experimental data, we decided that the number of common nucleotides on the 3' end of the Xit domains is a strong indicator for the degree of crosstalk in weight activation. Analysis of one memory in a 9-bit neural network showed a representative range of crosstalk (Fig. S17c): All matching activators had 7-nt complementarity with the inhibited weights, indicated by the yellow pixels on the diagonal. Most off-diagonal pixels are dark blue, indicating perfect orthogonality. Shades of lighter blue to green indicate 1 to 4-nt partial complementarity. While it is possible to completely avoid 2 to 4-nt partial complementarity in a 9-bit neural network, in the best case senecio each activator in a 100-bit neural network will have on average $100/3^3 - 1 = 10$ weights that it can spuriously activate with 3-nt partial complementarity.

The impact of the crosstalk is not actually as problematic as it appears for two reasons. First, the crosstalk shown in Fig. S17b was evaluated in isolation rather than in competition with the desired weight activation as it would be in an activatable memory. All possible inhibited weights composing the memory will be present, and thus each activator can simultaneously react with a matching and mismatching weight. The difference in reaction rates will naturally bias the activator toward the matching weight. The level of crosstalk will be 10 to 100-fold less than what is shown in Fig. S17b if the desired reaction is 10 to 100-fold faster. Second, the winner-take-all neural network architecture can tolerate noise reasonably well. A relatively low level of background in the memories would not affect the classification performance for most patterns. The impact of the noise is to shift the input pattern to a less ideal position in the weighted sum space, which in some cases

may result in reduced on-off separation in the output. Simulations confirmed that a 100-bit neural network with two activatable memories was expected to still perform well when crosstalk in weight activation was included in the model (Fig. S17d).

Even though we concluded that crosstalk was a relatively minor problem here, we included it in the considerations for a better design (Supplementary Note 4.9).

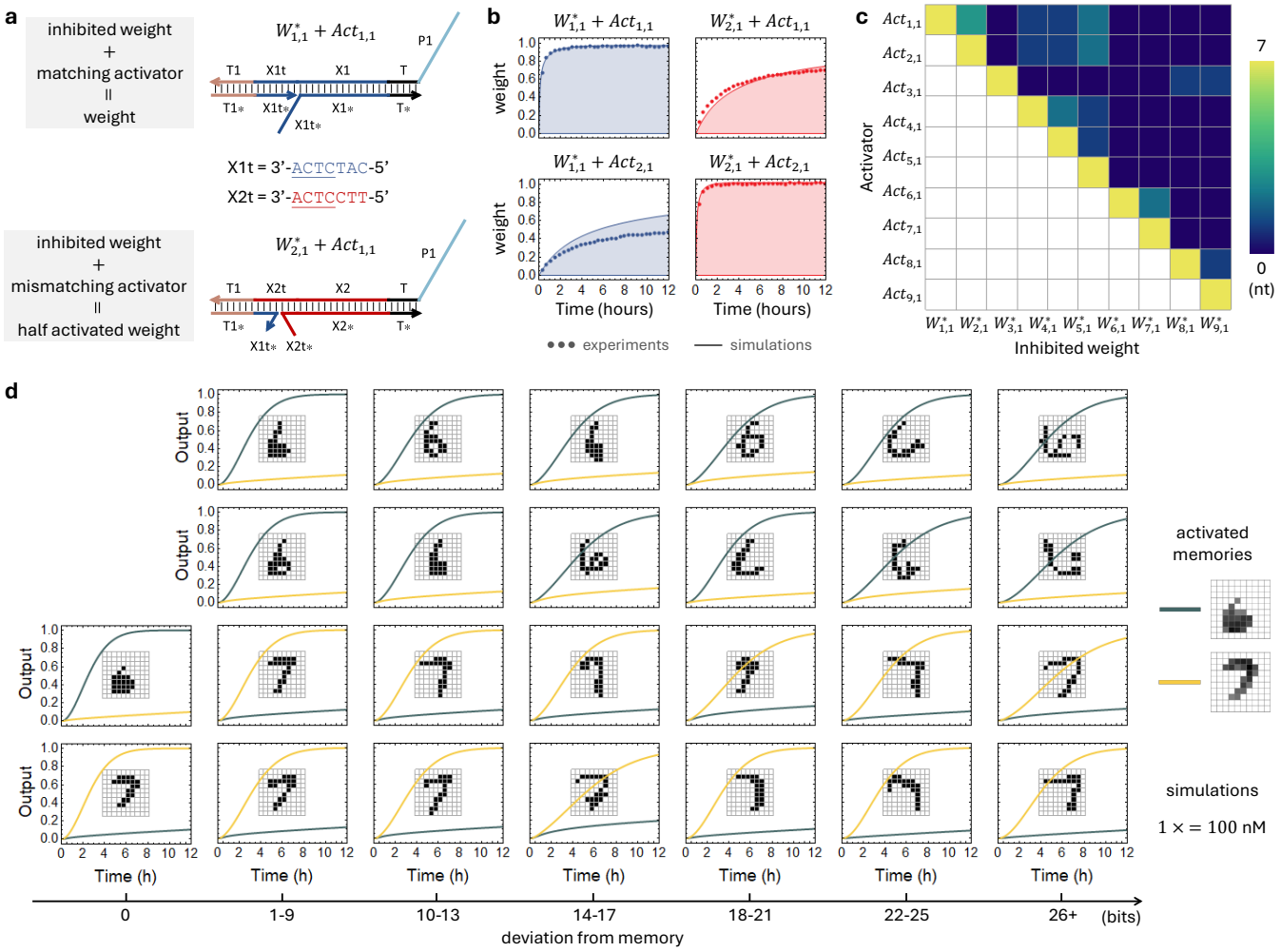


Fig. S17 | Crosstalk in weight activation. a, Domain-level diagrams illustrating two inhibited weights that can be differentially activated by the same activator strand. Sequences of the Xit domains are given to illustrate shared nucleotides at the 3' end, allowing a mismatching activator within the same memory ($Act_{i,j}$ with the same j but different i) to bind by the Tj∗ toehold and spurious branch migrate to half way open up the Xit∗ toehold on the inhibited weight. In a 100-bit neural network with strand sequences using a 3-letter alphabet, some bits will share up to 4 nucleotides of the Xit domain. **b,** Fluorescence kinetics experiments of the desired on-target weight activation compared with the undesired off-target weight activation. Experimental data are shown in dots and simulations are shown in solid lines. In each plot the species concentrations were $Act_{i,j} = 1\times$, $X_i = 1\times$, $W^*_{i,j} = 2\times$, and $Rep_j = 2\times$, where $1\times = 100$ nM. **c,** A matrix showing the possible levels of crosstalk in weight activation. Each of the activators shares all 7 nucleotides (yellow; maximum strength) with its intended target weight. However, all activators also have varying strengths of off-target weight activation with between 0 (dark blue) and 4 (green) shared nucleotides. **d,** Evaluating the impact of crosstalk by simulations of a 100-bit two-memory neural network, using the type of matrix in (c) as the basis for modeling the crosstalk. Two activated memories of MNIST digits 6 and 7 are each represented by a corresponding collection of 20 activators. The test pattern is shown on each plot. A representative set of test patterns were selected based on their positions in the weighted sum space and their deviations from the memories.

4.9 Moving to a better design

We learned an important lesson from extensive rounds of design revisions discussed above (Supplementary Notes 4.1 to 4.8): Identifying challenges one by one and coming up with solutions for each challenge may lead to limited success. For complex molecular systems, a solution for one problem could give rise to another problem somewhere else in the system. This phenomenon could further cascade, and in the worst case scenario, forming a deadlock in a cycle. Realizing the fundamental failure mode of this debugging strategy, we decided to utilize a different strategy where all challenges are considered as a whole and solutions are devised to address the entire body of challenges simultaneously.

We summarized seven problems discovered in Design 1 (Fig. S18, left).

Problem 1: The learning process requires training patterns and their labels to be irreversibly consumed and activators irreversibly produced during each training event so that no leftover inputs and labels would interfere with subsequent training and no learned memories would be partially erased. When drains used in the learning motif are not fully irreversible, errors occur. For example, the first learned memory have a reduced magnitude compared to the second, and the second learned memory have a higher background like a shadow from the first.

Problem 2: The ability to learn complex and diverse patterns depends on the total number of species involved in learning and the ratio of reacted versus unreacted species upon the completion of training. For example, learning two classes of 100-bit MNIST handwritten digits that each contains 20 bits of 1s requires 200 inhibited activators and 200 drains, out of which 80% will remain unreacted. The open toehold Tj$*$ on any unreacted drain $D_{i,j}$ from learning a j-th class of memory allows it to occlude any inhibited weight $W^*_{k,j}$ within the same memory. The occlusion significantly reduces the effectiveness of weight activation and leads to poor testing performance.

Problem 3: Unreacted inhibited activator $Act^*_{i,j}$ from learning can trigger an undesired leak reaction with inhibited weight $W^*_{i,j}$, initiated by blunt-end base stacking and subsequent branch migration within both Tj and Xit$*$ domains. This leak temporarily activates all weights and allows input strands in a test pattern to spuriously react with weights that are not in the learned memories. When a mismatch adjacent to the Tj domain is introduced in the inhibited activator to improve the irreversibility of learning, the leak becomes severe and leads to poor testing performance.

Problem 4: The activators used to turn on the weights encode both class specificity j and bit specificity i in short toeholds. This information encoding scheme poses challenges for complex learning and pattern classification tasks where the number of bits in training and test patterns is large. Crosstalk in weight activation becomes possible when some activators have shared nucleotide sequences at the 3' end of the Xit domain. The crosstalk allows input strands in a test pattern to react with partially activated weights that are not in the learned memories and leads to reduced testing performance.

Problem 5: Because toehold Xit is used to encode bit specificity, all input strands have distinct toeholds for reacting with the weight molecules. The difference in toehold sequences lead to difference in reaction rates for implementing the weight multiplication function. On the other hand, the implementation of winner-take-all function requires the weighted sum signals to arrive at the same time, otherwise whichever signal that arrives first will bypass the annihilator and produce an output. The rate difference in weight multiplication will favor the inputs that have stronger toehold sequences, making the corresponding bits more significant in a memory, resulting in biased classification decision.

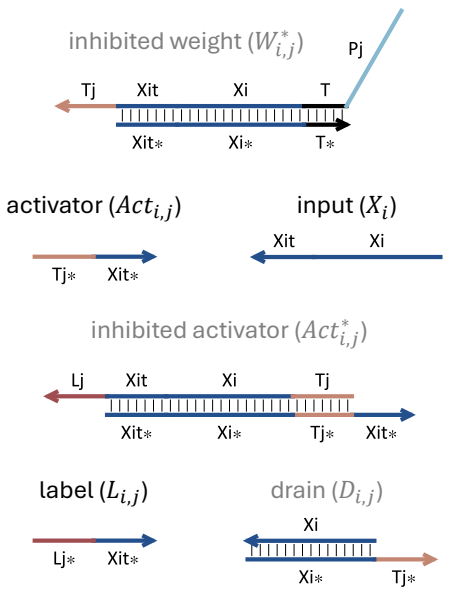
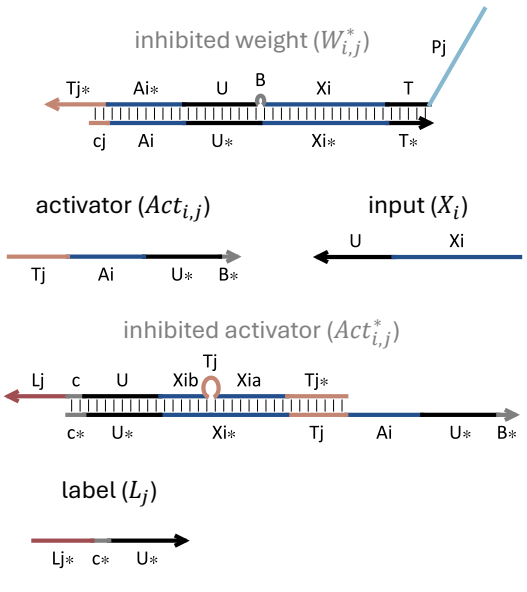
	Design 1	Design 2
		
Design properties	▪ Distinct toehold (Xit) in input ▪ Specificity of weight activation encoded in toehold (Xit) ▪ Fast reverse rate in weight activation ▪ Violation of 3-letter code in activator (Tj* + Xit*) ▪ Bit (i) and class (j) information both required in label ▪ Irreversibility of learning provided by a separate drain molecule ▪ No clamp on inhibited weight ▪ No clamp on inhibited activator	▪ Universal toehold (U) in input ▪ Specificity of weight activation encoded in branch migration domain (Ai) ▪ Slower reverse rate (requiring the formation of bulge B) in weight activation ▪ Restored 3-letter code in activator (U* and B* use As and Ts only) ▪ Only class (j) information required in label ▪ Irreversibility of learning provided by hairpin formation (Tj-Xia-Tj*) ▪ Clamp on inhibited weight (cj) ▪ Clamp on inhibited activator (c*)
Problem 1: reversibility in learning	Drain is not fully irreversible, resulting in errors in subsequent training	Enhanced irreversibility embedded within the inhibited activator itself
Problem 2: weight occlusion	Open toehold (Tj*) on drain occludes toehold (Tj) on inhibited weight	No drains needed; toehold (Tj) hidden in a bulge or a hairpin before or after learning occurs
Problem 3: leak between inhibited activator and weight	Mismatch introduced to improve drain irreversibility results in worse leak	No mismatches needed; clamp added
Problem 4: weight crosstalk	Not enough specificity in Xit	Enhanced specificity in Ai
Problem 5: weight multiplication rates	Different rates because of distinct toehold sequence (Xit) on input.	Similar rates because of universal toehold sequence (U) on input
Problem 6: activator occlusion	Violation of 3-letter code leads to occlusion between signals	Restored 3-letter code reduces occlusion
Problem 7: number of label strands	n strands for each class label (n is the number of bits in each memory)	1 strand for each class label regardless of the number of bits per memory

Fig. S18 | Summary of two molecular designs with unique properties and tradeoffs.

Problem 6: The 3-letter code is used to minimize undesired secondary structures within molecules and reduce spurious bindings between molecules. Specifically, all single strands and single-stranded domains longer than a toehold on a double- or multi-stranded complex must contain As, Ts, and Cs only. This rule applies to all initial, intermediate, and final species in the entire system (Supplementary Note 3.3). For consistency, all non-star domains in our designs contain As, Ts, and Cs only, and accordingly their complementary star domains contain As, Ts, and Gs only. The activator strand violates the 3-letter code, because it comprises two consecutive toeholds that are star domains, longer than a toehold. As a result, all activators can spuriously bind to other single-stranded non-star domains via G-C base pairs, for example occluding the intermediate products released from weight multiplication or become occluded by the single-stranded Pj domains on the weights. The former occlusion will contribute to biased classification decision, whereas the latter will contribute to reduced effectiveness of weight activation, both negatively impact the testing performance. The label strand also violates the 3-letter code, but the impact is less problematic because the molecules involved in leaning, the inhibited activators and drains, do not have single-stranded non-star domains to spuriously bind to the labels. All labels will be consumed during training and not expected to interfere with testing. Both the label and activator can occlude the input via hybridization between the Xit$*$ and Xit domains. However, this occlusion is unavoidable given the allosteric toehold mechanism and is not much of a concern when Xit is no longer than 7 nucleotides.

Problem 7: The label strand is supposed to only contain the class information j, providing a desired output response for each training pattern. However, it contains both Lj$*$ and Xit$*$ domains, the latter is required for reacting with the inhibited activator $act^*_{i,j}$ and uncovering the toehold for input binding. As a result, 100 label strands are needed to represent a desired output response for arbitrary 100-bit training patterns, and 100 inhibitor strands are needed to consume the excess label. This complexity introduces undesired cost and potential spurious interactions that reduce the performance of learning and testing.

With the seven challenges in mind, we came up with two main goals for an improved design.

Goal 1: Develop a learning motif that does not involve a drain but embed irreversibly within the inhibited activator itself – this would address the first three problems as a whole.

Goal 2: Utilize a universal toehold on the input while introducing a separate branch migration domain for the activatable weight motif – this would address the last four problems as a whole.

In Design 2 (Fig. S18, right), the inhibited weight incorporates a 9-nt Ai domain. The activator strand binds to the open Tj$*$ toehold, branch migrate through the Ai domain, and further branch migrate through the universal toehold U* and expose it for input binding. Compared to encoding the bit information in toehold Xit, the longer Ai domain provides a better specificity. The universal toehold promotes similar reaction rates for weight multiplication across all inputs in a test pattern and allows for a single label strand that encodes the class information for a training pattern. The universal toehold also provides a solution for the violation of 3-letter code in Design 1: the sequence of U can be designed to contain As and Ts only, while the other two domains on the activator strand can be altered to non-star domains Tj and Ai. In this case, the activator sequence now contains As, Ts, and Cs only, satisfying the 3-letter code. It is undesired to change the U$*$ domain to a U on the activator, because that would result in a U$*$ on the input strand, breaking the 3-letter code again. Without a universal toehold, it would be impossible to design a large number of distinct toehold sequences using As and Ts only.

The additional Ai domain in the inhibited weight comes with a cost: the exposure time of the

toehold U would be reduced as the result of a larger state space for the intermediates in branch migration before U is reached. The shorter exposure time suggests a slower strand displacement reaction for weight multiplication. To compensate for this issue, we added a 2-nt bulge B between the U and Xi domains on the inhibited weight and incorporated its complementary domain B$*$ on the activator. When the activator binds to and reaches the end of the branch migration point on the inhibited weight, the bulge will be eliminated. Reverse branch migration becomes slower because it now involves the thermodynamic penalty for the reformation of the bulge, effectively extending the exposure time of the toehold U.

To remove the drain, we introduced a bulge Tj in the inhibited activator. The waste product of the learning reaction now folds into a hairpin, covering up the Tj$*$ toehold required for the reverse reaction (Fig. 1c). Even if the hairpin is not entirely stable and can briefly open up to expose Tj$*$, reverse branch migration will be hindered by the reformation of the bulge. We will show in a later section that the desired irreversibility was achieved (Supplementary Note 5.2). Because the Tj domain is hidden in a bulge or hairpin, before or after the learning reaction occurs, it cannot effectively occlude the Tj$*$ toehold on the inhibited weight. Furthermore, because the bulge is in the middle of the Xi domain away from the Tj$*$ toehold, it does not worsen the leak between the inhibited activator and weight. Nonetheless, we added a 2-nt clamp on the inhibited weight to mitigate the leak. Finally, we suspected that the bulge could promote undesired leak between the input and the inhibited activator, and thus added a 2-nt clamp on the inhibited activator to provide a thermodynamic penalty and kinetic barrier for the leak.

Moving to Design 2 was a major decision – it meant that we must throw out three years of work on Design 1 and start over from scratch. This decision eventually led to a successful demonstration of learning in DNA-based neural networks. Beyond the science presented in this paper, we cannot over emphasize the philosophy that all challenges must be considered as a whole and that the most extraordinary courage is needed at the most desperate time.

5 Design 2

5.1 Activatable weight motif

The activatable weight motif in Design 2 (Fig. S19a) utilizes an activator strand $Act_{i,j}$ consisting of four domains: toehold Tj, branch migration domain Ai, universal toehold U∗, and bulge domain B∗. Tj binds to the open toehold Tj∗ on inhibited weight $W^*_{i,j}$ and initiates branch migration across the Ai and U∗ domains. At the end of branch migration, B∗ binds to and eliminates the bulge loop B on $W^*_{i,j}$, driving the weight activation forward and allowing for a longer exposure time of toehold U∗ on the activated weight $W_{i,j}$ for input X_i to react with. Subsequent weight multiplication occurs following the same reaction mechanism as in Design 1. Key changes of this motif in Design 2 include encoding activator bit i information in a longer, branch migration domain (Ai), allowing for a universal toehold (U) on all input strands. We expect these changes to result in improved specificity of weight activation, and more uniform reaction rates of weight multiplication that promote fair competition in the downstream winner-take-all function, at the cost of two more domains in the activator and the extended part of the weight molecule.

In initial experiments for characterizing the activatable weight motif, we designed the Xi domains to be 26 nucleotides long (Fig. S19a). The length was determined based on a consideration for the stability of the learning motif in Design 2 (Fig. S20a): a bulge loop was inserted into the middle of the Xi domain on the inhibited activator, splitting it into two halves that each must be long enough (for example, 13 nucleotides) to remain stably bound at an intermediate state of the desired or leak reaction pathways. We will have more discussions about the impact of the Xi domain length in a later section (Supplementary Note 5.3), whereas the conclusions that we will draw in this section does not depend on that length.

As a result of a longer Xi domain and two addition domains Ai∗ and B, the top strand in the inhibited weight is quite long (70 nucleotides total in Fig. S19a). We asked: is there a benefit to replace the bulge with a nick using a three-stranded inhibited weight molecule (Fig. S19b, top right diagram)? Based on the design criteria that we established (Supplementary Note 3.1), the fewer strands per molecule the better for reducing stoichiometry errors. On the other hand, the fewer nucleotides per strand the better for reducing synthesis errors. Given the trade-offs, we decided to investigate both designs and compare their performances on four distinct measures: desired weight activation, undesired leak within the weight motif, crosstalk in activation, and leak at the interface of learning and testing.

For evaluating the performance of weight activation, we included designs with a smaller 2-nt bulge and a larger 5-nt bulge (Fig. S19b). We expected the larger bulge to drive the desired reaction forward more effectively but also increase the risk of undesired leak. Fluorescence kinetics experiments showed that the fastest kinetics and highest reaction completion were achieved with the nick design, due to the irreversibility of the activation step. However, while initial leak at the beginning of the experiments were similar across three designs, gradual leak in the nick design was 2 to 3 times worse than the 2-nt bulge design when the activator was absent (blue trajectories). Leak could be initiated by the input or fuel invading at the bulge or nick location, releasing the intermediate product via toeless strand displacement. The main difference in the leak mechanism is that the reverse reaction is either unimolecular (bulge design) or bimolecular (nick design). Since unimolecular reactions are faster at a relatively low concentration ($1\times = 100$ nM), leak is not only kinetically slow but also thermodynamically unfavored for the bulge design. The design with a

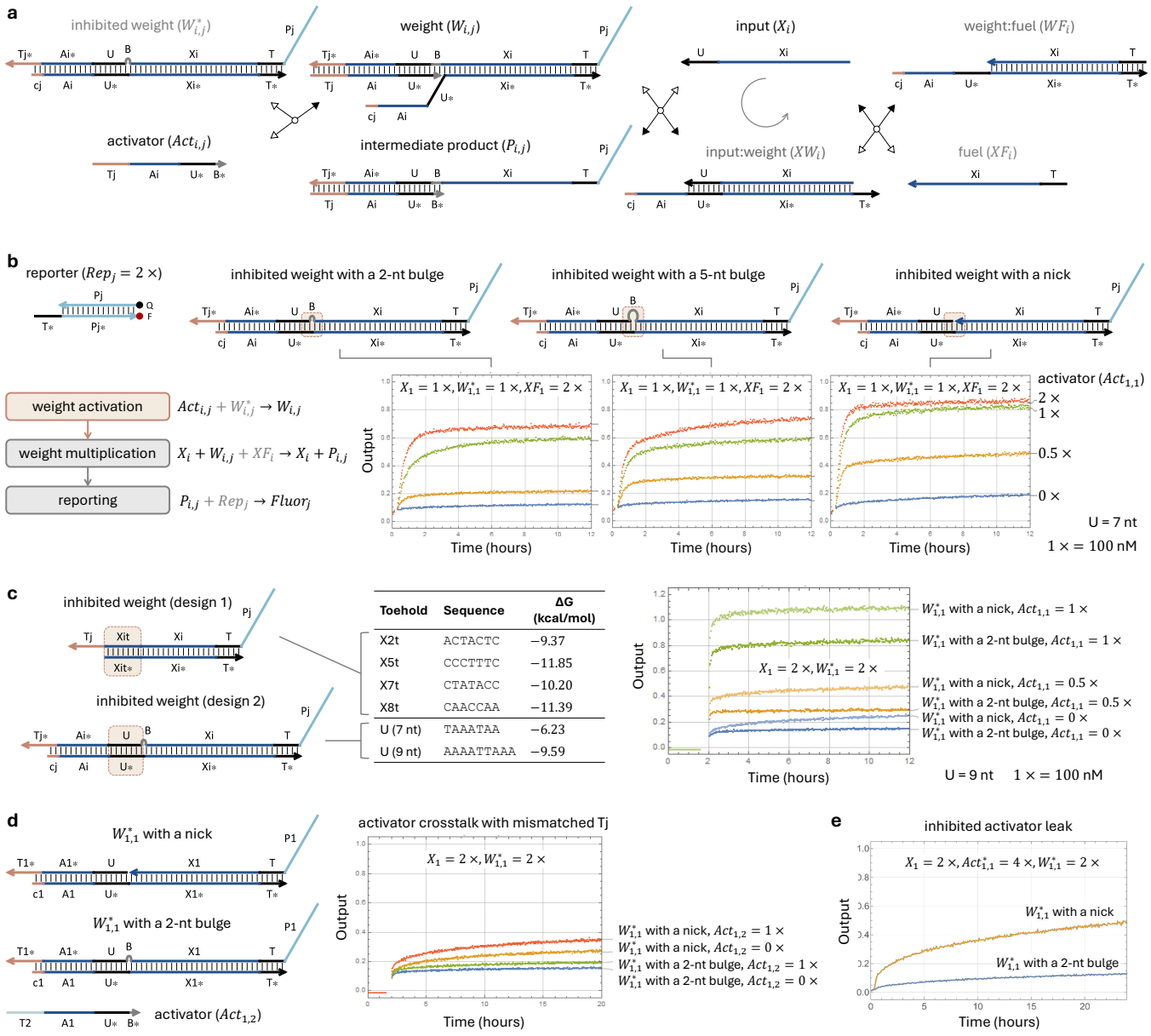


Fig. S19 | Activatable weight motif in Design 2. a, Redesigned DNA strand-displacement implementation of an activatable weight motif. The design features a branch migration domain Ai for encoding bit i information and a separate universal toehold U for inputs to react with the activated weights. The redesigned motif also includes a bulge loop B in order to bias the activation step forward to reveal the universal toehold. **b,** Fluorescence kinetics experiments of weight activation, comparing three distinct design choices. The bulge loop was either 2 or 5 nucleotides, or replaced with a nick resulting in a 3-stranded complex. All trajectories have input, inhibited weight, fuel, and reporter. In each set of experiments, various amounts of activator were added which control the amount of weight that becomes active. **c,** Weight activation experiments with a longer, 9-nt toehold U. In Design 1, each activator toehold Xit was unique to the bit sequence and varied in strength with a ΔG (change in Gibbs free energy) around -9 to -12 kcal per mol. Design 2 switched to a universal sequence U that contains only As and Ts. To achieve a binding strength comparable to the previously designed Xit domains, we updated the length of U from 7 to 9 nucleotides. **d,** Measuring the crosstalk of weight activation with an activator that has mismatched Tj domain, indicating crosstalk of the same bit between two memories. **e,** Measuring leak at the interface of learning and testing. Unreacted inhibited activators from learning are expected to be present when weight activation is carried out in testing.

5-nt bulge had slightly better kinetics and reaction completion, but worse leak compared to the 2-nt bulge. We concluded that a larger bulge was not necessary and moved on to compare other performances between the 2-nt bulge and the nick design.

Given that the universal toehold U in Design 2 is As and Ts only, it is unsurprising that weight activation was generally slower than Design 1 (Fig. S10). To match the strength of a 7-nt toehold with average sequences, we updated the design with a 9-nt sequence for U and repeated the weight activation experiments (Fig. S19c). Indeed, faster kinetics was observed for both the 2-nt bulge and nick designs, suggesting that the updated toehold strength achieved a balance between undesired activator-input occlusion and desired input-weight interaction. The nick design remained to exhibit higher reaction completion (green and orange trajectories) but worse leak (blue trajectories). For simplicity, fuel was removed in this set of experiments and the observed leak was purely due to the input invasion.

There are two types of crosstalk in weight activation with an activator $Act_{i,j}$ that has mismatched i or j – different bits within a memory or the same bit in different memories. In Design 1, we focused on evaluating the first type of crosstalk (Fig. S17). Here in Design 2, the bit i information is encoded in Ai and cross activation is only possible if branch migration proceeds through all mismatched nucleotides within Ai, at the end of which toehold U$*$ would be exposed. Considering the kinetic barrier and thermodynamic penalty, this type of crosstalk is highly unlikely. We thus focused on the second type of crosstalk for comparing the bulge and nick designs (Fig. S10d). Crosstalk between memories is possible regardless of the orthogonality of Tj toeholds, because an activator can invade at the bulge or nick location and branch migrate through the U and Ai domains. Experiments showed more crosstalk in the nick design, which was measured by the difference between the signal levels with and without the mismatched activator. Like the leak, the activator crosstalk can be reversed via a unimolecular reaction in the bulge design before input comes in, but not in the nick design, providing an explanation for the observed difference.

Finally, we evaluated the leak at the interface of leaning and testing by adding an inhibited activator with matching i and j (Fig. S10e). While the bulge design maintained a level of leak similar to that within the weight motif, the nick design became nearly 5 times worse than the bulge design. This result pushed us to conclude that the nick design was not suitable for the purpose of creating a learning DNA neural network. We will discuss the mechanism of the interface leak in a later section (Supplementary Note 5.3). The conclusion here is simply that having a fast, unimolecular reverse rate helps mitigate leak, which is a unique feature of the two-stranded inhibited weight design with a bulge. This feature is especially useful when multiple motifs are integrated together to create a complex molecular system.

5.2 Learning motif

The learning motif in Design 2 (Fig. S20a) embeds irreversibly within an inhibited activator itself, eliminating the need of a separate drain. A label strand L_j reversibly binds to the inhibited activator $Act*_{i,j}$, uncovering the toehold U$*$ for input binding. Input X_i has the same format as in the activatable weight motif, participating in both learning and testing. Here, it displaces the top strand in the inhibited activator and becomes bound to the bottom strand while revealing the Tj toehold that completes the function of an activator $Act_{i,j}$ whose concentration represents the value of a learned weight. The irreversibility is designed into two mechanisms: First, the Tj$*$ toehold in the released top strand binds to the Tj domain within the same strand and becomes inhibited in the stem part of a hairpin, unable to initiate the reverse reaction. The reversibility of the hairpin formation depends on the strength of Tj and the length of Xia, the loop part of the hairpin – a longer or stronger Tj makes the hairpin more stable and a longer Xia makes it less stable.[4] With a 7-nt Tj and a 13-nt Xia, NUPACK[25] analysis suggests that the hairpin is expected to be closed with an over 98% probability (single-nucleotide fraying at either end of the double-stranded domain is excluded). Second, even when the hairpin is open, the reverse reaction is unfavored due to the entropic cost of a bulge loop formation, which is expected to slow down branch migration by over 150-fold based on NUPACK analysis.

The bulge loop Tj splits the Xi domain in the top strand of the inhibited activator into two separate domains Xia and Xib (Fig. S20a). In the intermediate activator with a bound label, Xib needs to be long enough to remain double-stranded, avoiding input crosstalk by partial complementarity within the Xib domain. Similarly, Xia also needs to be long enough to prevent displacement of the top strand by a mismatched input, as reverse branch migration is slow due to the bulge loop. In the initial experiments shown in Fig. S20, we used a 26-nt Xi domain that splits into a 13-nt Xia and Xib. As one of our design criteria is the fewer nucleotides per stand the better, we asked how short can Xia and Xib be and showed the impact of these two domain lengths in a later section when leak was evaluated at the interface of learning and testing (Supplementary Note 5.3).

To observe the process of learning, the inhibited activator can be modified with a fluorophore and the input with a quencher (Fig. S20a). The fluorophore and quencher will be brought together in the input-bound activator, resulting in lower fluorescence. We experimentally investigated four inhibited activator designs, with a Tj bulge or with a nick, and with or without a clamp (Fig. S20b). Two separate wastes are produced in the nick design, one of which can initiate reserve branch migration but only within the Xia domain, providing a stronger irreversibility than the bulge if only branch migration is considered. However, the increased complexity of a three-stranded complex and the exposed Tj$*$ domain in a waste are both undesired. Thus we only used the nick design as a comparison for evaluating the irreversibility of the bulge design.

The clamp was designed to reduce leak between the input and the inhibited activator when the label is absent. Without the clamp, fraying at the 5' end of the bottom strand in the inhibited activator allows for input invasion via toeless strand displacement. On the other hand, the clamp increases the state space of branch migration for the label strand and reduces the time that the toehold U$*$ will be fully exposed for input binding.

Experiments suggested that the activator was successfully produced when both label and input were present for all four designs (Fig. S20b). When an excess of inhibitor Inh_j was added to bind to the label and drive the learning reaction backward, no fluorescence change, expect minor fluctuation due to instrument noise, was observed in 24 hours. This result proved that the bulge design was as

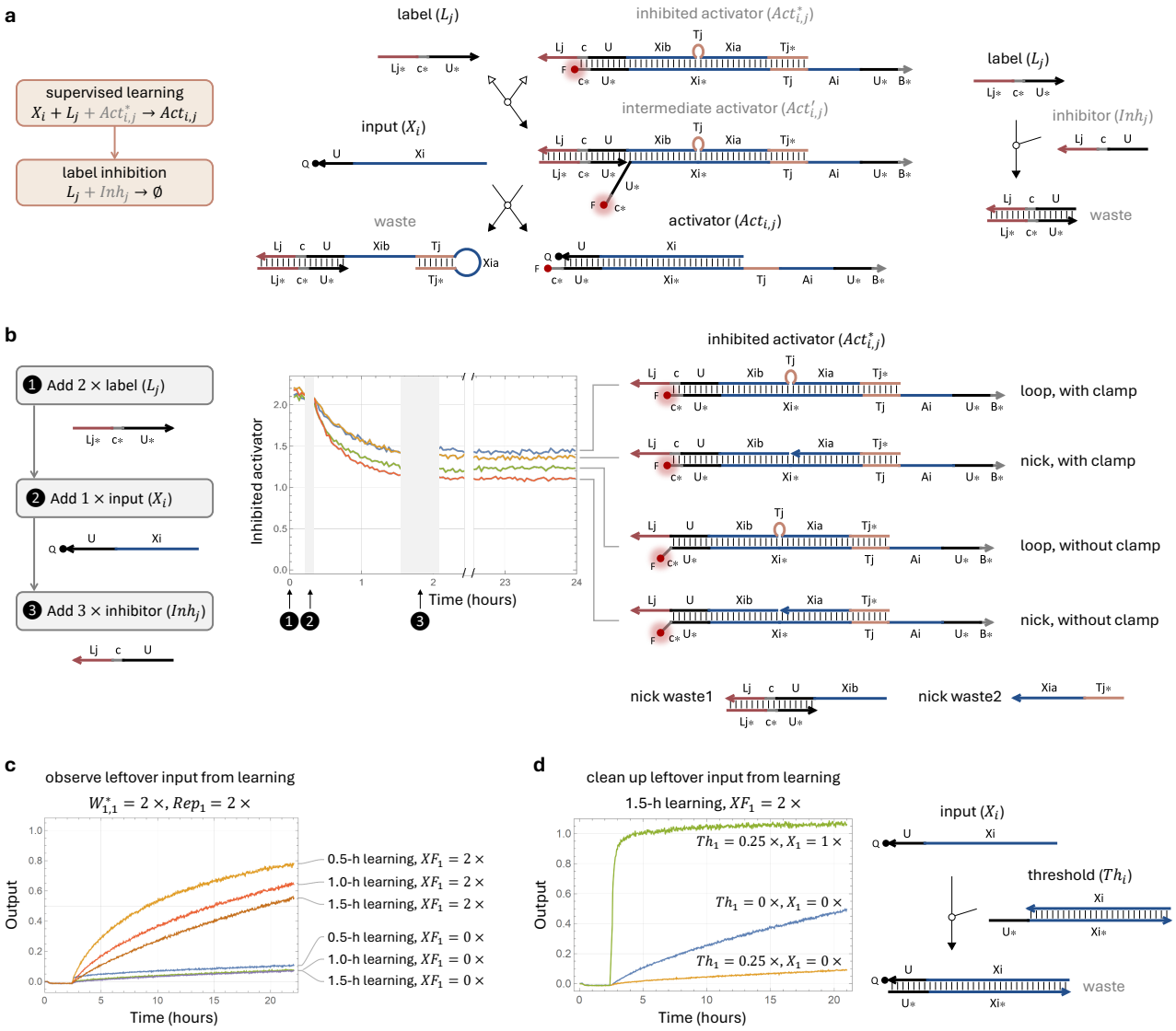


Fig. S20 | Learning motif in Design 2. a, Redesigned DNA strand-displacement implementation of a learning motif. Inhibited activators can have a fluorophore label for observing the learning process, which will be quenched by an input in the produced activator. **b,** Demonstration of a learning reaction and its irreversibility. Two types of designs were compared. The first had a bulge loop Tj which disrupts reversible branch migration as well as forms a hairpin stem. The second design used a nick in the top strand. After adding label (step 1) and input (step 2), inhibitor was added (step 3). The inhibitor was expected to bind to the label and drive the learning reaction backwards, creating an increase in fluorescence, if the final step of learning is not fully irreversible. At the end of the experiments, approximately 22 hours after the inhibitor was added, the fluorescence remained unchanged for both types of designs. We also tested the two types of designs with and without a clamp. The designs with clamps learned more slowly. **c,** Measuring input leftover after learning for 0.5, 1, or 1.5 hours. Input patterns leftover due to incomplete learning will affect subsequent rounds of training as well as the fidelity of the test patterns. After learning for a given amount of time, weight and reporter molecules were added to observe how much output signal was produced from remnant input strands. The inclusion of fuel in the weight multiplication reaction caused the input to react catalytically; without fuel very little output was produced. Learning for longer lessened the amount of leftover input. **d,** Evaluating the impact of a threshold for clean up input strands leftover from learning. After 1.5 hours of learning, the leftover input produced 0.5× output in approximately 20 hours when no threshold was present. With 0.25× of threshold, only 0.1× of output was produced. As desired, this amount of threshold did not prevent 1× of new input from catalytically releasing output strands from all 1× of the activated weight.

irreversible as the nick design, supporting the effectiveness of toehold inhibition within a hairpin. The nick and bulge designs without a clamp had a larger fluorescence change within the first 1.5 hours of the experiments than the designs with a clamp, indicating faster kinetics, which agreed with the expectation above. Unfortunately, there was a flaw in the experimental design here: we should have evaluated the input leak by adding the input first before the label. We falsely concluded that the clamp was not necessary and moved on with the bulge design without clamp to promote faster kinetics – this mistake caused us a substantial detour. We will discuss how leak within the learning motif affects the overall performance of a 100-bit two-memory neural network and bring back the clamp in a later section (Supplementary Note 5.9).

While an inhibitor strand was used to remove excess label after each round of training (Fig. S20a), input was expected to be fully consumed and not interfering with subsequent training or testing. However, this assumption is only true if the learning period is long enough to reach reaction completion, the reaction is fully irreversible, and no input is defective and left unreacted. To complement the explicit evaluation of reversibility in Fig. S20b and evaluate the other two conditions, we designed another set of experiments to observe leftover input from learning (Fig. S20c). In these experiments, learning took place when label L_1 and input X_1 were present to collectively produce activator $Act_{1,1}$ for 0.5, 1, or 1.5 hours before inhibitor Inh_1 was added to clean up excess label. After leaning, an inhibited weight molecule $W^*_{1,1}$ was introduced, along with a reporter Rep_1 that detects the product of weight multiplication. The inhibited weight is expected to be activated by $Act_{1,1}$ and react with any leftover input from learning. With and without fuel XF_1, the leftover input will produce a signal catalytically or stoichiometrically. Indeed, a small amount of leftover input was detected without fuel and amplified with fuel. The amount was reduced but not eliminated with a longer period of learning, suggesting unreacted, defective input.

Unsurprisingly, synthesis errors in inputs affect their ability to react with inhibited activators. In particular, truncations at the 5' end of an input strand could prevent it from successfully displacing the top strand in the inhibited activator when the length of the domain that must spontaneously dissociate becomes too long. This length is 7 nucleotides in Tj plus the number of nucleotide truncations on the input. On the other hand, the weight molecule has a shorter, 5-nt toehold T. An input that fails to be consumed in learning could still react with a weight molecule and release an output signal because of the roughly 100-fold faster dissociation rate at the end of branch migration.

Leftover input from an earlier training pattern adds noise to the next training or test pattern. The noise is small enough to be neglected in subsequent training because its impact is limited to stoichiometric reactions, but may cause issues in testing because of the signal amplification process embedded in weight multiplication. To clean up the undesired noise, we used a threshold molecule that converts an input to waste (Fig. S20d). After 1.5 hours of learning, the leftover input was again amplified by the fuel, reaching 0.5× in approximately 20 hours (blue trajectory, a repeat of the brown trajectory shown in Fig. S20c). By contrast, only 0.1× of signal was produced within the same time when 0.25× of threshold was introduced (orange trajectory). Importantly, this amount of threshold did not negatively impact the desired behavior when new input was added after learning – output was catalytically released to reach a fully triggered signal level of 1× within 2 hours (green trajectory). We concluded that a threshold helps clean up unreacted input from learning and is necessary for maintaining the desired testing performance.

5.3 Impact of domain length on the interface leak

As discussed in Supplementary Notes 5.1 and 5.2, the length of the Xi domain will impact the performance of learning, testing, and their interface. To explicitly investigate this, we started by evaluating the impact of this domain length on weight activation, comparing a 26-nt and a 10-nt Xi (Fig. S21a). With the same amount of activator $Act_{1,1}$, a higher reaction completion was observed with the inhibited weight $W^*_{1,1}$ that has a shorter Xi domain (green and blue trajectories). This observation can be explained by synthesis errors: Both strands in each of the complexes were ordered unpurified from Integrated DNA Technologies (IDT). We expected more synthesis errors in longer strands, especially including truncations at the 5' end of the strands. With a truncated Pj domain, the released output strand will react with the reporter reversibly. The reversibility will be further increased by fluorophore-quencher interaction (Fig. S15b), resulting in lower reaction completion. Moreover, stoichiometry errors can result in excess top strand in the inhibited weight, functioning as a threshold for the activator. Due to the limited resolution of gel electrophoresis, PAGE-purification of the annealed complex cannot remove all structures with synthesis errors, but it helps remove excess strands and substantially truncated structures – this explains the improvement in reaction completion when the complexes were purified, whereas output produced by the inhibited weight with a 26-nt Xi remained at a lower signal level compared to that with a 10-nt Xi (red and orange trajectories).

From the point of synthesis errors, the shorter strands the better. However, other problems arise when the Xi domain is too short. When a 12-nt Xi domain was used in a 9-bit two-memory neural network, it exhibited poor pattern classification performance after learning (Fig. S21b) – 7 out of 12 test patterns were classified incorrectly, either as a tie between classes "L" and "T" (both outputs went on in the 2nd and 3rd tests on the first row and the 5th test on the second row) or as the opposite class (output Y_2 went above Y_1 for the first class, "L", and vice versa, in the in 4th through 6th tests on the first row and the 6th test on the second row).

We hypothesized that the poor performance was due to leak between the unreacted inhibited activator from leaning and the inhibited weight required in testing (Fig. S21c), creating high background noise that diminishes the result of learning. When an input strand X_i in a test pattern is introduced, it will have an active weight to react with if bit i is present in the learned memory, but it can also participate in a leak pathway when the weight remains inhibited. As all inhibited activators $Act^*_{i,j}$ must be present during learning to allow for arbitrary training patterns, unused $Act^*_{i,j}$ will remain present when testing takes place. The B$*$ domain on an inhibited activator can invade at the bulge B on any inhibited weight and temporarily open the toehold U$*$ for input binding. Subsequent branch migration will reveal the toehold T for the output signal from weight activation to be detected by a reporter with the Pj domain. However, this leak mechanism is expected to be highly reversible because the backward reaction is unimolecular. If the bit i and class j information match between the inhibited activator and the inhibited weight, branch migration initiated at the bulge will also proceed through the U$*$, Ai, and Tj domains. If the Xia domain on the inhibited activator is short enough to spontaneously dissociate at the end of branch migration, Tj$*$ and Tj on the top strand of the inhibited activator will form a hairpin. The hairpin will prevent reserve branch migration and allow for the input to displace the top strand and irreversibly convert the inhibited activator to an activator. Once the activator is produced as a leak product, it will quickly trigger weight activation, corrupting the learned memories.

To verify the above hypothesis about leak at the interface of learning and testing, we compared

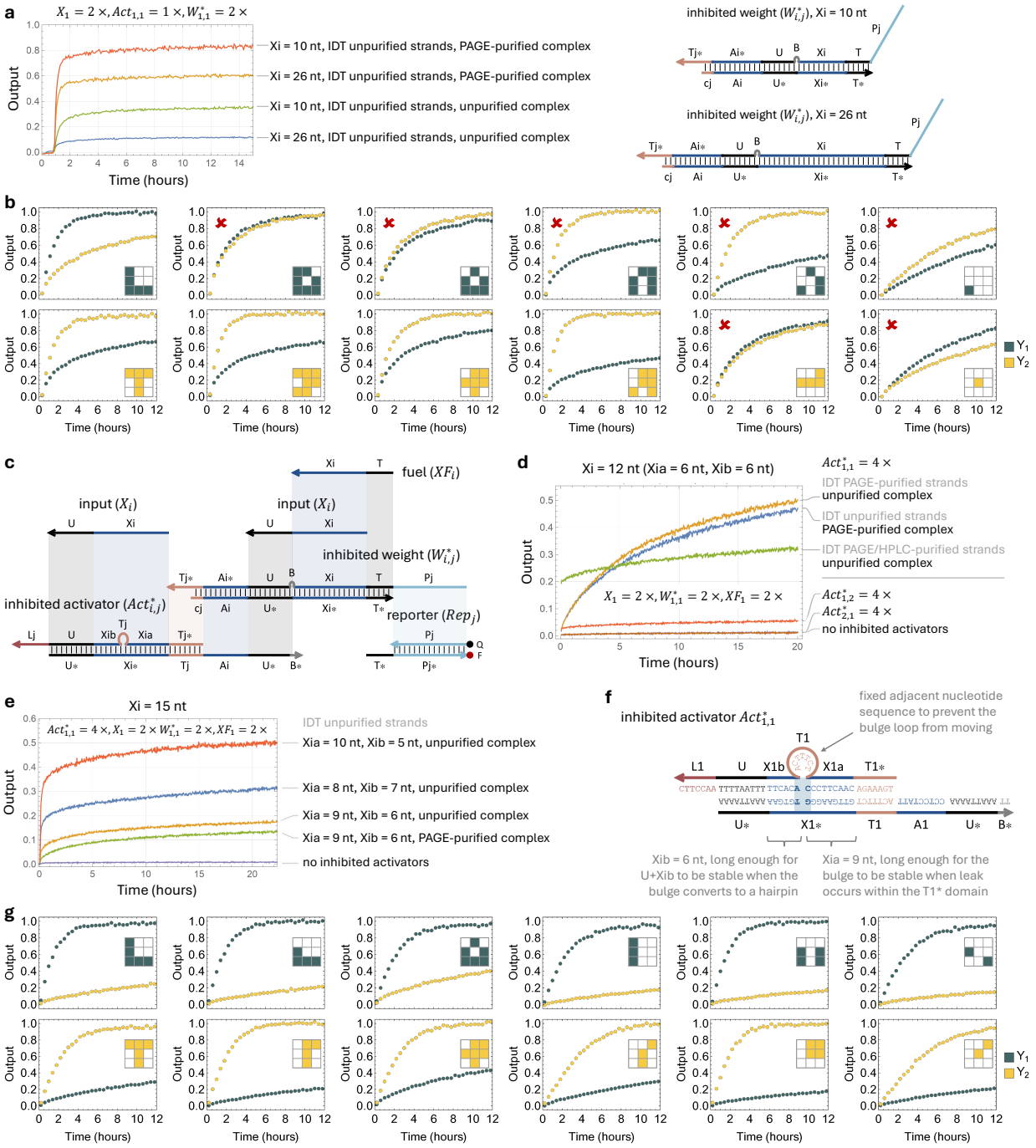


Fig. S21 | Impact of domain length on leak at the interface of learning and testing. **a,** Fluorescence kinetics experiments of weight activation using two distinct domain lengths of Xi. **b,** Experiments on pattern classification of a trained 9-bit two-memory neural network using motifs from Design 2. The first column of test patterns are the same as training patterns "L" and "T". Pattern corruption – deviation from the learned patterns – increases for each experiment plot going left to right. Red crosses indicate incorrect classification. Xi domains are 12-nt long. **c,** Domain-level diagram highlighting the shared domains of all species for evaluating the leak pathway. **d,** Experiments measuring the leak with matched and mismatched inhibited activators. **e,** Experiments evaluating the leak with a common 15-nt Xi domain but distinct combinations of domain lengths of Xia and Xib. **f,** Summary of design choices for leak reduction. **g,** Experiments on pattern classification of a trained 9-bit two-memory neural network using revised sequences based on the design criteria in (f). $1\times = 100$ nM for all experiments.

inhibited activators with matching and mismatching domains (Fig. S21d). Indeed, for spurious weight activation in $W^*_{1,1}$, inhibited activator $Act^*_{2,1}$ with mismatched bit i did not introduce any background noise compared to the case with no inhibited activators (bottom two trajectories that are exactly on top of each other). Similarly, inhibited activator $Act^*_{1,2}$ with mismatched class j resulted in a very small amount of leak below 0.05× in 20 hours. By contrast, the matching inhibited activator $Act^*_{1,1}$ produced a 10-fold higher leak, reaching 0.5× in 20 hours (orange trajectory).

We also investigated how the leak is affected by the strand and complex purity of the inhibited activator (Fig. S21d). In general, for individual strands ordered from IDT, PAGE is expected to provide higher purity than HPLC, and both of them are expected to enhance full-length products.[26] Surprisingly, in this set of experiments, inhibited activator using HPLC-purified bottom strand exhibited higher initial leak but lower gradual leak compared to PAGE-purified bottom strand (green trajectory). The initial leak could be due to excess bottom strand given that the annealed complex was not further purified. We do not have an explanation for the difference in gradual leak, except a speculation that the two strands had different qualities due to random noise in the synthesis process given that they were ordered at different times. PAGE purification of the annealed complex showed a mild improvement compared to the unpurified complex with PAGE-purified strands (blue trajectory), even though the two strands were ordered unpurified. This observation is consistent across a variety of experiments that we have performed, and is the reason that we typically order unpurified strands if they will be annealed into a complex which will then be PAGE-purified to remove excess strands and malformed structures.

Taking a closer look at the leak mechanism (Fig. S21c), we further hypothesized that a relatively short Xi domain can be tolerated if a balance between Xia and Xib domain lengths is achieved to mitigate the progression of leak. The Xia domain needs to be long enough to reduce the probability of hairpin formation when Tj$*$ on the top strand of the inhibited activator is open due to spurious branch migration. The U and Xib domains together needs to be long enough so that the top strand will not spontaneously dissociate from the bottom strand even when the hairpin has formed, allowing for the possibility of reverse branch migration when the hairpin opens up. Experimental results supported the hypothesis (Fig. S21e): For a 15-nt Xi domain, the combination of a 9-nt Xia and a 6-nt Xib resulted in the least amount of leak, below 0.15× in 10 hours (orange trajectory). A shorter Xia and longer Xib led to roughly twice more leak (blue trajectory) and a longer Xia and shorter Xib led to over three times more leak (red trajectory). PAGE purification of the inhibited activator complex helped further reduce the leak to roughly 0.1× in 10 hours (green trajectory).

We summarized the design choices in Fig. S21f. The Tj bulge loop could slide to the left or right within the Xi domain if adjacent nucleotides matched the sequences at the 5' or 3' end of the bulge. This sliding would shorten Xib or Xia, respectively. Given the importance of both domain lengths, we designed all Xi domains to have specific nucleotide sequences adjacent to the bulge that lock it in place. With these design criteria, the 9-bit two-memory neural network exhibited much improved pattern classification performance after learning (Fig. S21g) – all 12 test patterns were correctly classified with a clear on-off separation.

5.4 Scaling up to a 100-bit learning neural network

With a successful demonstration of 9-bit pattern classification after supervised learning (Fig. S21g), we moved on to scale up the system for learning 100-bit MNIST handwritten digits (Figs. S22a-c).

A hundred training patterns per class were randomly selected from the database and the correspondingly DNA strands were used as a single mixture. This training procedure presents all patterns of the same class at once, conceptually similar to the batch training technique used in machine learning. In the mixture of training patterns, the 20 most common bits (top 20 bits with the largest analog values in the averaged training patterns) were preserved whereas the less significant bits were omitted – this can be viewed as a manual pruning step done by a human teacher, which is undesired for autonomous learning within the DNA neural network itself. In preliminary experiments, this non-ideal yet simple procedure allowed us to compare the performance of a learning neural network with that of a neural network carrying out pattern classification after *in silico* training shown in our prior work,[3] where the 20 most common bits were used as weights. We will discuss an alternative learning procedure that avoids the manual pruning step in a later section (Supplementary Note 5.12).

A representative set of six test patterns per class was selected based on their positions in the weighted sum space – farther from the diagonal line (equal weighted sums, $s_1 = s_2$) is easier to classify and closer is harder. Test patterns within a 20% margin of the diagonal line ($|s_1 - s_2| \leq 0.2$) were excluded – they were determined to be theoretically classifiable but experimentally unfeasible. Test patterns outside of the 20% margin were divided into six uniformly distributed areas and one test pattern per area was randomly chosen. Additionally, each mixture of training patterns was converted into a binary pattern and used as a reference test pattern for each class. If the training patterns are successfully learned into a memory, one of the two weighted sums for the reference test pattern should reach the maximum value 1, resulting in the largest distance to the diagonal line among all test patterns.

We trained the neural network with three distinct pairs of 100-bit MNIST digits in three separate learning procedures (Figs. S22a-c). In all cases, a strong bias was observed in the second learned memory – test patterns in the second class (digits "1", "4", and "7") were classified correctly with clear on-off separation between the two outputs, whereas test patterns in the first class (digits "0", "3", and "6") were poorly or incorrectly classified with a small difference between the two outputs.

To investigate the source of the bias, we performed five sets of diagnosis experiments using two test patterns per class including the reference tests "6" and "7" (Figs. S22d-h). First, we asked: did the second set of training patterns leaked into the first learned memory? To answer this question, we performed a parallel learning procedure where the two classes of training patterns were presented to the DNA neural network in two separate test tubes, and the learned memories were combined after training. With this procedure, the impact of the order of training was removed. A mild improvement was observed but the bias largely remained (Fig. S22d), suggesting that the order of training did not play a major role in creating the bias.

Second, we asked: was the bias introduced by learning or did it already exist in pattern classification with activatable memories? Experimental data showed that when the learning process was removed and a set of activator strands were directly given to the DNA neural network, a more significant improvement was observed (Fig. S22e), suggesting that some molecules involved in learning were major contributors to the bias observed in testing. However, bias also existed in the activatable memories alone, because the second test pattern in class "6" was farther away from the diagonal

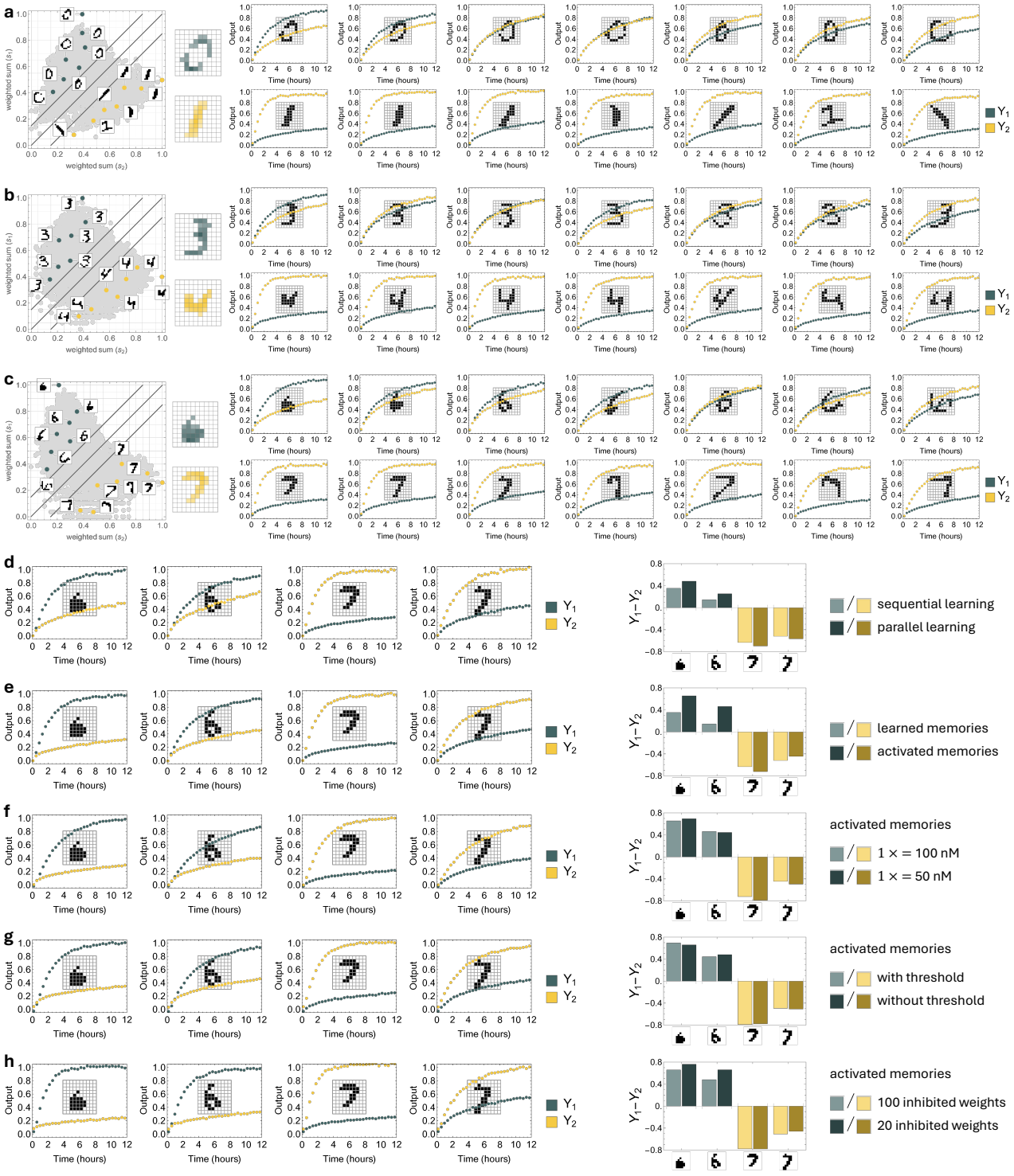


Fig. S22 | Scaling up to a 100-bit learning neural network. a-c, Fluorescence kinetics experiments of testing a 100-bit neural network after trained with MNIST handwritten digits "0" and "1" (**a**), "3" and "4" (**b**), or "6" and "7" (**c**). Two memories were learned sequentially, at 1× = 100 nM, with 0.25× threshold for cleaning up leftover inputs from learning, and with all 100 inhibited weights present for testing. **d-h,** Diagnosis experiments comparing sequential and parallel learning (**d**), learned and activated memories (**e**), higher and lower concentration (**f**), with and without threshold (**g**), or a full set and subset of inhibited weights (**h**). Bar charts show the difference between two outputs at 12 hours for each test pattern in the diagnosis experiment (darker colors) compared to that in the original experiment shown in c (d-e, lighter colors) or the earlier diagnosis experiment shown in e (f-h, lighter colors).

line in the weighted sum space than the second test pattern in class "7" and was expected to show a larger on-off separation between the two outputs, disagreeing with the experimental observation.

The next three sets of diagnosis experiments focused on the activatable memories without learning and were compared to the results shown in Fig. S22e. We previously identified that a challenge for scaling up DNA strand-displacement circuits was spurious binding that led to toehold occlusion and proposed using a lower concentration to reduce the spurious binding at the cost of slower computation (Supplementary Note S15.1 in Qian and Winfree, *Science*, 2011[2]). Here, to evaluate the impact of concentration, we repeated the experiments in Fig. S22e but at half the concentration (Fig. S22f). The results were roughly the same, indicating that the desired reactions did not slow down at a lower concentration, presumably due to reduced toehold occlusion. Even though the overall performance was similar, we decided that a lower concentration helps save materials and reduce labor for in-house PAGE purification of annealed complexes and thus is preferred for future experiments.

In an earlier section (Supplementary Note 5.2) we measured the amount of input leftover from learning and used a threshold molecule to clean up each unreacted input strand. Here, we sought to evaluate if the threshold molecules introduced any undesired crosstalk or occlusion that affected the testing performance (Fig. S22g). The results were roughly the same with and without the threshold molecules, suggesting that they did not contribute to the observed bias in pattern classification.

Finally, we simplified the activatable memories by using a subset of the inhibited weights that corresponded to just the 20 bits that were supposed to be activated in each memory (Fig. S22h). Eliminating the 80 unused inhibited weights led to desired performance of pattern classification where both test patterns in class "6" showed increased on-off separation between the two outputs, whereas the outputs for both test patterns in class "7" remained roughly unchanged, indicating that the bias was fully corrected. This result suggested that the unused inhibited weights were not fully inhibited and spurious activation in one memory was more severe than that in the other.

Together, the five sets of diagnosis experiments led us to conclude that undesired interactions occurred both in learning and in testing – they collectively impacted the performance of pattern classification after training. To investigate further, we will take a closer look at the learning motif, the activatable weight motif, and their integration in the next three sections (Supplementary Notes 5.5 through 5.7), particularly considering the roles of unused molecules including inhibited activators and inhibited weights.

5.5 Label occlusion

Only a small fraction of the inhibited activators will react with inputs from a training pattern and the rest will be unused for any specific learning event. For example, 80% of the inhibited activators are unused for learning a 100-bit MNIST handwritten digit that has 20 1s. Thus it is important to understand how the unused inhibited activators impact the system behavior. As shown in Fig. S23a, a label strand L_1 encoding the class 1 information will react with all inhibited activators for the correspondingly memory 1 ($Act^*_{i,1}, \forall i$) regardless of whether the input X_i is present. This aspect of the design suggests that the presence of unused inhibited activators is expected to cause label occlusion. To evaluate the degree of occlusion, we designed a set of experiments where a learning reaction takes place to store a 1-bit pattern X_5 into a 4-bit memory 1 consisting of four inhibited activators $Act^*_{1,1}$, $Act^*_{3,1}$, $Act^*_{5,1}$, and $Act^*_{7,1}$. In these experiments, we investigated three distinct versions of the label strand L_1 where the toehold U$*$ was truncated by 0, 1, and 2 nucleotides, respectively.

The full length label was expected to experience the strongest occlusion (longest time in a temporary state bound to $Act^*_{1,1}$, $Act^*_{3,1}$, or $Act^*_{7,1}$), whereas occlusion of the truncated labels were expected to be weaker. A partially opened U$*$ domain on $Act^*_{5,1}$ can still allow for the input strand X_5 to react with, and thus all three versions of the label were expected to participate in the desired learning reaction and become irreversibly bound to the top strand in the produced activator $Act_{5,1}$. On the other hand, the kinetics of learning may or may not be affected by the U$*$ domain length: a shorter U$*$ on the label has a weaker binding to the toehold U on the input, resulting in less input occlusion; but it also exposes a shorter toehold on the inhibited activator, leading to slower strand displacement – the impact of the two could balance out, in which case the truncation will mainly affect just the strength of occlusion. For all three versions, at any given time the fraction of label strand participating in the learning reaction depends on the ratio of inhibited activators with and without a corresponding input. If the reverse rate of label occlusion is sufficiently fast, the presence of unused inhibited activators should only slow down the learning reaction but not alter the steady-state concentration of the product. Conversely, the steady-state concentration will depend on the ratio of the inhibited activators.

Experiments showed that the latter scenario was true (Fig. S23b). The presence of unused inhibited activators (3× total concentration of $Act^*_{1,1}$, $Act^*_{3,1}$, and $Act^*_{7,1}$) significantly affected the performance of learning. The produced activator $Act_{5,1}$ concentration (inferred by the decrease in inhibited activator $Act^*_{5,1}$ concentration) was reduced by roughly 50% (middle three trajectories) compared to when the unused inhibited activators were absent (bottom orange trajectory). This result suggested that the reverse rate of label occlusion was very slow. A possible explanation is that the Xib domain has been shortened to 6 nucleotides to reduce synthesis errors and improve strand quality in both the learning motif and activatable weight motif (Supplementary Note 5.3). When the label is bound to an inhibited activator, the Xib domain could spontaneously dissociate, slowing down the reserve rate. In particular, intramolecular spurious binding between the U$*$ and Xib$*$ domains (Extended Data Fig. **??**b) could significantly impact the reserve rate. In this case, three quarters of the label strand will be strongly occluded by the unused inhibited activators and only one quarter will effectively participate in the learning reaction. With 0.5× input and 0.25× effective label, activator can be produced at half the amount of the input, agreeing with the observed 50% decrease in $Act_{5,1}$ production.

A solution to the problem of label occlusion is to increase the label concentration to be above

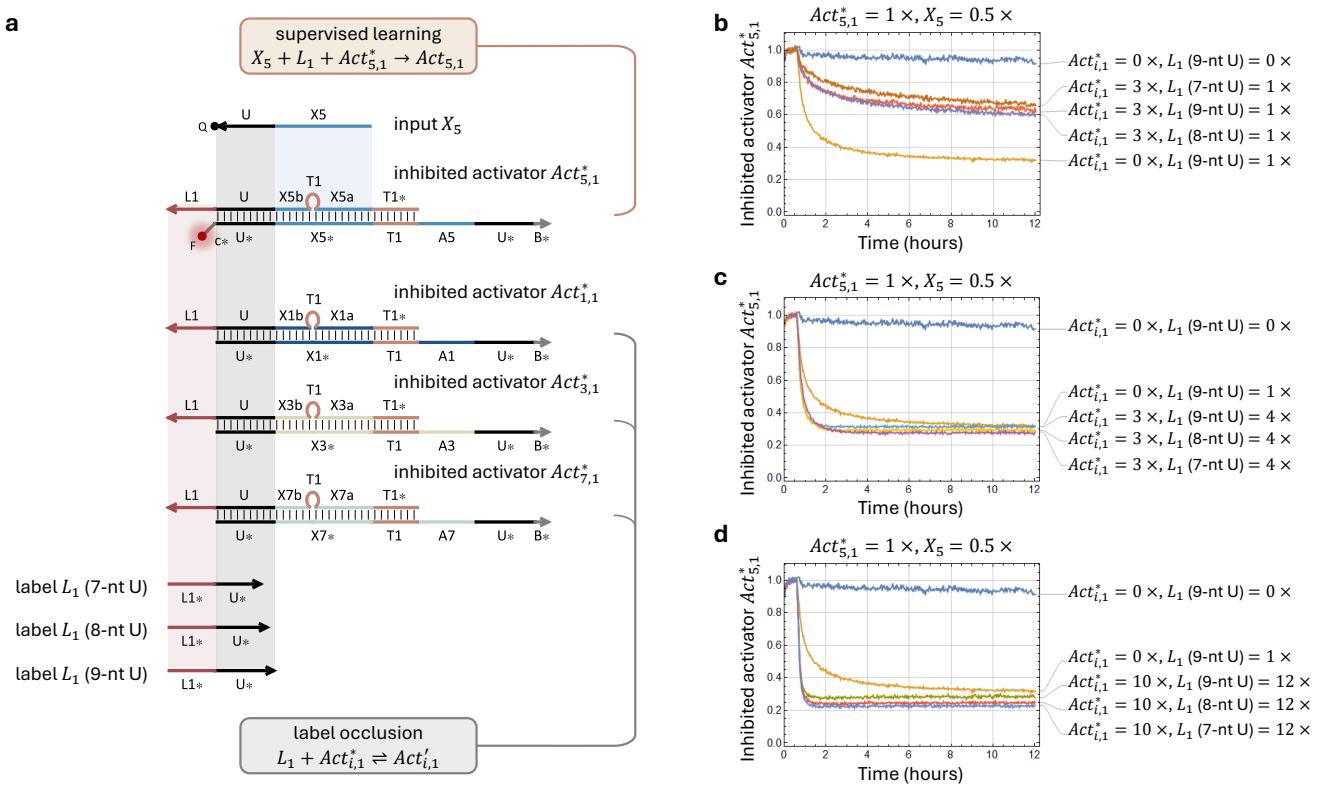


Fig. S23 | Unused inhibited activators occluding the label for learning. a, Domain-level diagrams of species involved in learning a 1-bit pattern into a 4-bit memory. Three choices for the label strand L_1 are shown with a 7, 8, or 9-nt universal toeholds U∗. Input X_5 has a quencher and inhibited activator $Act^*_{5,1}$ has a fluorophore. Decrease in fluorescence indicates reduced concentration of $Act^*_{5,1}$ and increased concentration of produced activator $Act_{5,1}$. **b-d,** Fluorescence kinetics experiments quantifying the effect of label occlusion with varying concentrations of unused inhibited activators and label at 3× and 1× (**b**), 3× and 4× (**c**), or 10× and 12× (**d**), respectively. Here, $Act^*_{i,1}$ is a mixture of $Act^*_{1,1}$, $Act^*_{3,1}$, and $Act^*_{7,1}$ at equal concentrations. Data in all three plots was normalized using a common positive and negative control: average of the first 5 data points in a sample with no label (blue trajectory) provided a baseline for no learning and was used as a reference for $Act^*_{5,1} = 1\times$; average of the last 5 data points in a sample with 3× excess label and 2.5× excess input (not shown here) resulted in a fully triggered learning reaction and was used as a reference for fully consumed $Act^*_{5,1} = 0\times$. Standard concentration $1\times = 100$ nM. Note that the produced activator was expected to be determined by the input concentration of 0.5×, but observed to be higher at roughly 0.7×. This difference was due to the inaccuracy of nominal concentrations of the fluorophore-labeled inhibited activator $Act^*_{5,1}$ and quencher-labeled input X_5 – because 1× in the plots was determined by the effective concentration of $Act^*_{5,1}$, it can be lower than the effective concentration of 1× input.

the total concentration of unused inhibited activators. With 4× label, the desired learning behavior was restored, achieving a full production level of activator $Act_{5,1}$ (Fig. S23c). Unsurprisingly, the kinetics also became faster with a higher label concentration. Scaling up from 4-bit to 100-bit memories requires a larger ratio of unused and used inhibited activators. Experiments showed that increased total concentration of unused inhibited activators was not an issue as long as the label concentration was also increased accordingly (Fig. S23d).

Besides occlusion, another type of undesired behavior that unused molecules can do is generating leak, which also becomes worse with increased complexity of the neural network. In the next section (Supplementary Note 5.6), we will explore adjusted annealing ratio as a method to reduce synthesis errors and thus leak in annealed complexes, including inhibited weights and inhibited activators.

5.6 Annealing ratio and complex purity

Annealing ratio is expected to have an impact on complex purity because of the competition between strands in hybridization. For example, if an inhibited weight molecule $W^*_{i,j}$ is annealed with an excess bottom strand, binding between a top strand and a full-length bottom strand will be preferred over truncated bottom strand. Even if the inhibited weight has formed with a truncated bottom strand at a higher temperature during annealing, an unbound full-length bottom strand can still displace the truncated strand at a lower temperature. As truncations occur near the 5' end of the strand, they may lead to a population of inhibited weight molecules that have the clamp (cj) missing or even part of the Ai$*$ domain uncovered. This population of molecules would react with an inhibited activator $Act^*_{i,j}$ and produce undesired leak of an output signal without learning, and adjusting the annealing ratio would help reduce the leak. Importantly, the annealed complex will be PAGE-purified to remove any excess strands, ensuring that no unbound bottom strands are present to function as a threshold for the input or occlude other signals in the system.

Experiments showed that after PAGE purification, an inhibited weight $W^*_{1,2}$ annealed with a 20% excess bottom strand performed similarly to one annealed with equal amounts of the top and bottom strands for desired weight activation (Fig. S24a), while the undesired leak with $Act^*_{1,2}$ was reduced from 0.28× to 0.16× in 15 hours (Fig. S24b). The same set of molecules were used in both sets of experiments, except that the activator $Act_{1,2}$ in Fig. S24a was replaced by one of the four inhibited activators in Fig. S24b. $Act^*_{1,2}$ was expected to produce the highest amount of leak, because it has both a matching toehold T2 and a matching branch migration domain A1 to spuriously activate the inhibited weight $W^*_{1,2}$ when synthesis errors are present in both molecules. $Act^*_{1,1}$ was expected to produce less leak, because it has a matching branch migration domain but a mismatching toehold, resulting in an increased barrier for spurious activation. $Act^*_{3,2}$ and $Act^*_{3,1}$ were expected to produce the lowest amount of leak, because they have a mismatching branch migration domain, inhibiting the progression of spurious activation. Experimental observations agreed with the expectations, and showed that leak was reduced in all four cases when a 20% excess bottom strand was used in annealing the inhibited weight before it was PAGE-purified.

Determining the annealing ratio for the inhibited activator was trickier. An excess top strand would reduce truncations within Tj$*$ on the 5' end of the top strand in the annealed $Act^*_{i,j}$, which helps reduce leak between the inhibited activator and inhibited weight. An excess bottom strand would reduce truncations within U$*$ on the 5' end of the bottom strand, which promotes faster learning. At first, we thought an excess top strand was a better choice because the benefit of leak reduction would be more important than the speed of learning. However, experiments showed that an unused activator annealed with a 20% excess top strand led to severe label occlusion, almost indistinguishable from the negative control with no labels (Fig. S24c). This result was surprising because we expected that PAGE purification would have removed any unbound top strands and they should not have been present to bind to and inhibit the label strand.

Upon a closer examination of the design, we realized that the excess top strand in $Act^*_{i,j}$ could bind to the inhibited activator complex via hybridization between the open U and U$*$ domains. Any partial sequence complementarity between the Lj and Ai domains as well as that between the Xib and B$*$ domains would enhance the undesired binding. At a high concentration used in PAGE purification (for example, 45 μM), the undesired binding could be favored. With a large amount of sample (for example, 150 μL per well) loaded onto a gel, separation between the desired two-stranded complex and undesired three-stranded complex could be insufficient, and the sample extracted from

the gel could have included the undesired complex. As a lower concentration (1× = 100 nM in Fig. S24c), the excess top strand would dissociate from the inhibited activator and bind to the label strand instead. This hypothesis was supported by the appearance of samples on a purification gel (Fig. S24c, bottom right gel image): $Act^*_{1,1}$ annealed with a 20% excess top strand (1.2:1) showed a smear (highlighted in the orange zone) larger than the expected size of the complex (blue zone), and no band corresponding to the excess top strand was observed on the gel.

Apart from the unexpected label occlusion, the expected leak reduction was observed for an inhibited activator annealed with excess top strand (Fig. S24d). Compared to the same inhibited activator annealed with excess bottom strand, the initial leak decreased from 0.05× to 0.02×, whereas the gradual leak remained similar, consistent with the hypothesis of reduced synthesis errors and improved complex purity. Given the tradeoff between label occlusion and leak, we decided that for the inhibited activator neither strand in excess was desired, and opted for a 1:1.05 ratio of annealing in later experiments. The slight excess of bottom strand was used to account for the inaccuracy in stoichiometry.

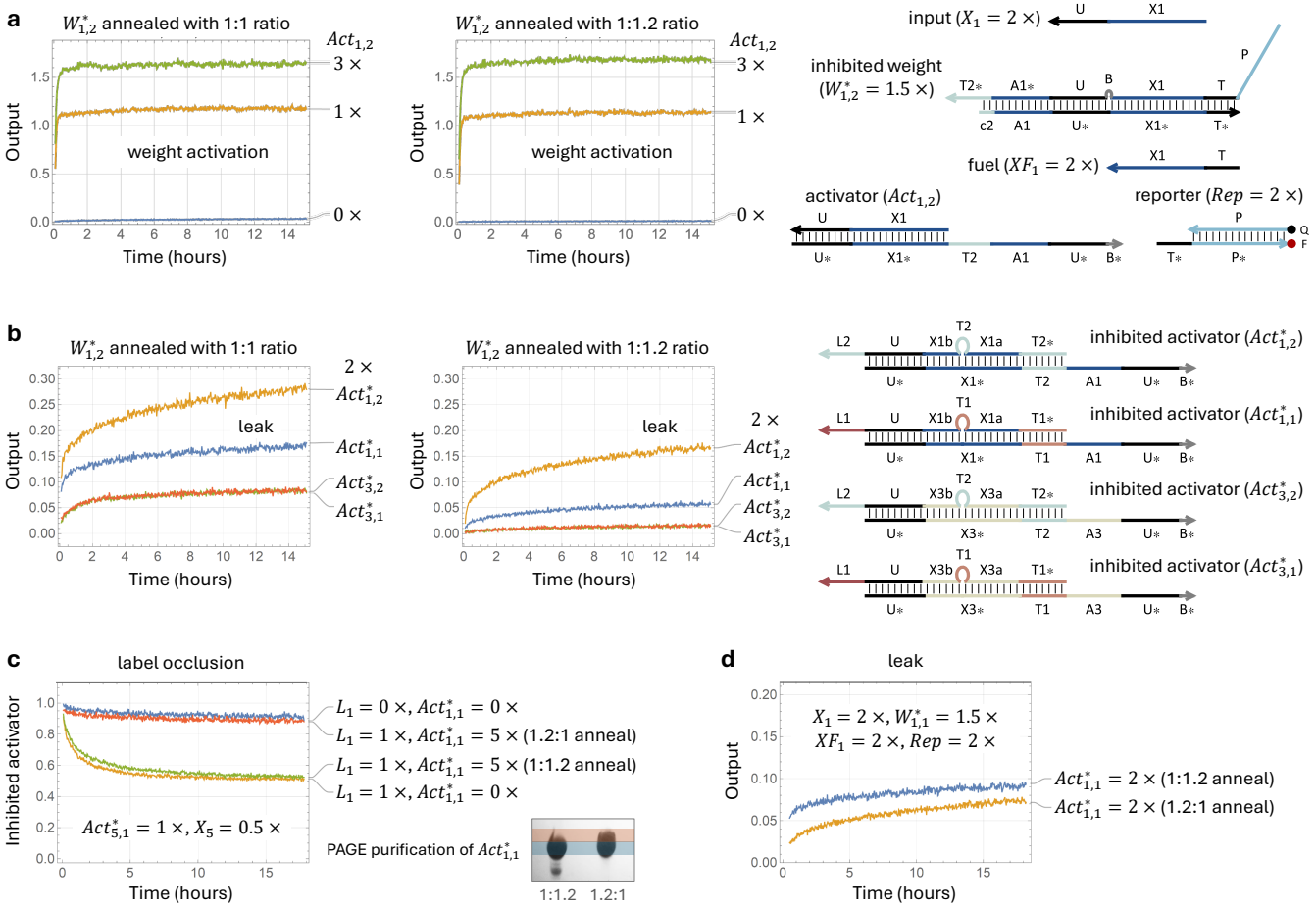


Fig. S24 | Annealing ratio and complex purity. a, Comparing the performance of an inhibited weight annealed with a top to bottom strand ratio of 1:1 or 1:1.2. **b,** Comparing spurious activation leak between inhibited activators and the inhibited weight annealed with distinct strand ratios. **c,** Comparing annealing ratios for an unused inhibited activator present in learning. An image of a purification gel is shown with a blue and orange zone highlighting the expected and unexpected annealing product, respectively. **d,** Comparing inhibited activator annealing ratios for leak with an inhibited weight. All two-stranded complexes were PAGE purified after annealed. 1× = 100 nM.

5.7 Effectiveness of fuel

Besides occluding labels, unused inhibited activators could also occlude inputs by temporarily covering up the toehold U. This occlusion would reduce the effective concentration of the inputs and possibly affect computation within the weight multiplication layer of the neural network. A fuel strand is used in weight multiplication so that if the input signal is below the activated weight, the output can still be catalytically produced to reach the desired value of weighted input. To evaluate whether input occlusion is a problem, we investigated the effectiveness of the fuel.

The fuel strand has a toehold T for initiating strand displacement that releases the input strand after it has displaced the top strand from the inhibited weight, allowing a single copy of the input to react with multiple copies of the weight molecule. In our previous work on creating DNA neural networks for molecular pattern classification but not learning,[3] a common 5-nucleotide toehold T was used in all signals including input and fuel. Here, for creating a learning DNA neural network, the weight molecules must be initially inhibited and only become selectively activated upon training. The weight activation mechanism reversely exposes a toehold U$*$ for input binding, and this toehold needs to be strong enough for weight multiplication to effectively proceed. Moreover, because of the system-level three-letter code requirement (Supplementary Note 3.3), U is composed of As and Ts only and its length was chosen to be 9 nucleotides to match the strength of an average 7-nt toehold with As, Ts, and Cs (Fig. S19c).

The imbalanced strengths of input toehold U and fuel toehold T led to a problem (Fig. S25a, left plot): the observed kinetics of output production was divided into a fast stoichiometric phase and a slow catalytic phase. Output signal quickly reached the level of the input signal (0.2, 0.4, or 0.6×) within the first couple of data points and then slowly increased by roughly 0.3× over 2 hours. This result suggested that the fuel was unable to effectively release the input and drive the output production. By contrast, a stronger 7-nt toehold T on the fuel restored the desired catalytic behavior (Fig. S25a, right plot), where output reached roughly 1× reaction completion within one hour for all three input concentrations above zero. Here, the stronger toehold was achieved by a 2-nt extension at the 3' end of the bottom strand in the inhibited weight that covers up the last two nucleotides of the P domain and a complementary extension at the 5' end of the fuel.

The speed of weight multiplication is expected to have an important impact on the pattern classification performance of the DNA neural network for the following reason. The winner-take-all function is implemented using pairwise annihilation followed by signal amplification, and the desired order of the two reactions is enabled by the difference of the two reaction rates – stronger toeholds are used on the annihilators and weaker toeholds on the amplification gates. However, a limitation of the implementation is that the annihilation reaction is trimolecular, which results in reduced rate difference when the two competing signals are both at low concentrations. Slower weight multiplication means slower production of the two competing weighted sum signals, which makes annihilation less effective. A larger fraction of the losing signal may bypass the annihilator and get amplified, leading to reduced on-off separation between output signals and thus worse performance in pattern classification.

Making weight multiplication faster by revising the toehold on the fuel would require revisions of the downstream network components. Instead, we decided to compensate the ineffectiveness of fuel by a higher input concentration. This method sets the input concentration above the activated weight and eliminates the need for catalytic behavior – it is not a good solution but an acceptable one here, given the goal of this work is to demonstrate learning. In future work, the fuel strands can

be redesigned to have a longer common toehold and the change can be propagated to downstream network components, enabling more robust pattern classification after learning, where test patterns with a wide range of input concentrations can be effectively classified, even when the inputs are occluded by unused inhibited activators.

With a higher input concentration, experiments showed consistently better on-off separation for eight distinct test patterns in a 100-bit DNA neural network with two activated memories each consisting of 20 inhibited weights (Fig. S25b). For example, in the bottom right plot, with 0.05× inputs, the largest on-off separation was achieved around 8 hours, where output Y_2 reached 0.9× and output Y_1 remained below 0.2×. With a 10-fold higher input concentration, the largest on-off separation was achieved earlier around 4 hours, where Y_2 reached 0.9× and Y_1 remained below 0.1×, resulting in roughly 0.1× increase in on-off separation. Based on this result, we decided to use the higher input concentration in later experiments.

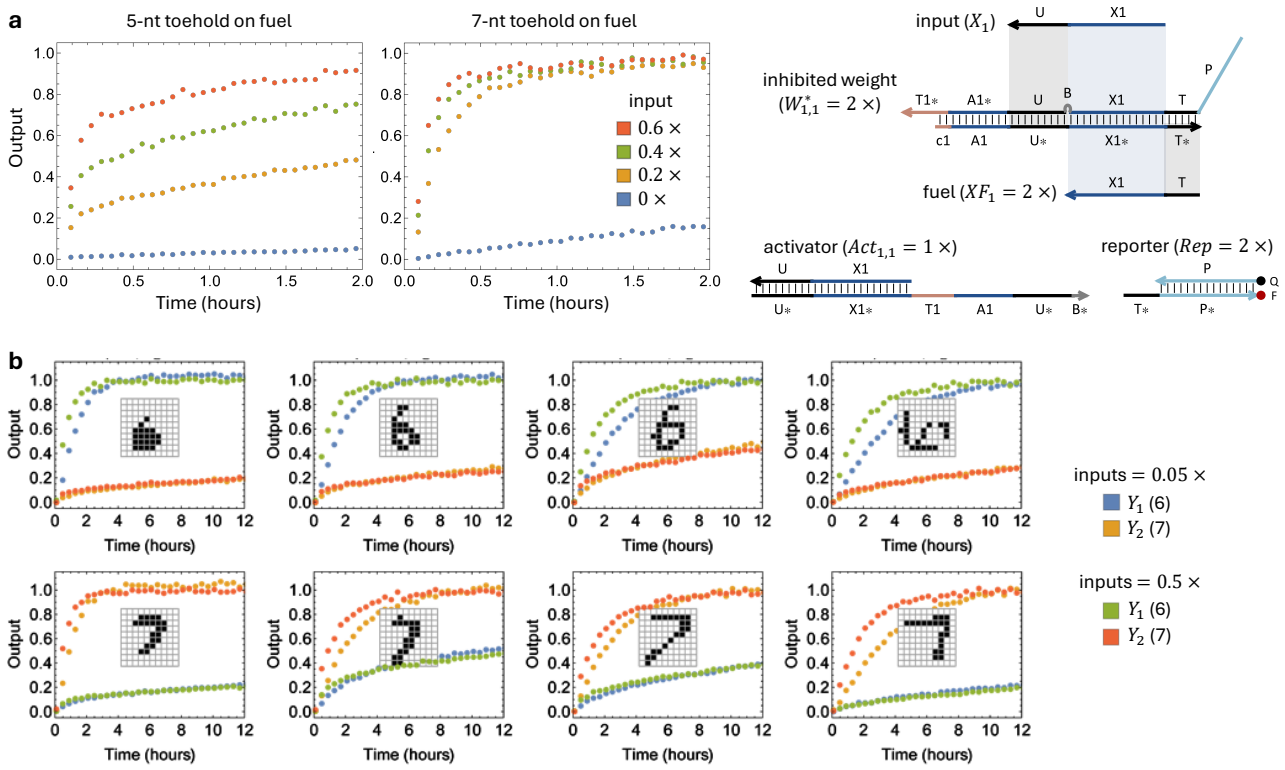


Fig. S25 | Effectiveness of fuel for catalytic weight multiplication. a, Comparison of the kinetics of output production in weight multiplication using fuel with either a 5 or 7-nt toehold. The rate at which a fuel strand displaces an input strand from an input-weight product species is dependent on the strength of the universal fuel toehold T and the universal input toehold U. When the fuel toehold T is 7 nt, the release of input happens more quickly than when the fuel toehold is 5 nt. **b,** Comparison of pattern classification performance in a 100-bit two-memory DNA neural network with a 0.05× or 0.5× input concentration. Each memory consisted of 20 inhibited weights, which were directly activated by a set of 20 activators without learning. The left four plots with 0.05× inputs are repeats of the same experiments shown in Fig. S22h. Total input concentration is 1× or 10×, given 20 1s in each test pattern. Analog weights are the same as shown in Fig. S22c, with a total concentration of 1×. One pair of output trajectories (blue and yellow) shows the network classification performance when the input concentration is similar to the weight concentration. The other pair of output trajectories (green and red) shows faster classification and better on-off separation when the input concentration is 10-fold higher. In both cases, the fuel toehold is 5 nt. In the latter case, weight multiplication does not rely on the fuel species enabling catalysis. 1× = 50 nM.

5.8 Improved performance in a 100-bit learning neural network

In the previous three sections, we looked into the scalability problem by investigating the roles of unused molecules that are supposed to remain inhibited after learning and not interfere with the desired system behavior. Because the fraction of these molecules grows larger with increasing network complexity, any spurious activation or unexpected interference would negatively impact the system scalability. We concluded that unused inhibited activators cause label occlusion, and a solution is to use excess label (Supplementary Note 5.5). We also concluded that annealing ratio affects complex purity and spurious activation. To achieve optimal performance both when used and unused, each complex may have a specific annealing ratio based on its molecular structure (Supplementary Note 5.6). Finally, we identified that the fuel driving catalytic output production in weight multiplication was ineffective, leading to reduced pattern classification performance when inputs are at a low concentration. To improve the system robustness to input occlusion by unused inhibited activators, a higher input concentration is necessary (Supplementary Note 5.7).

Applying the above findings, we repeated a representative set of experiments in Fig. S22 with

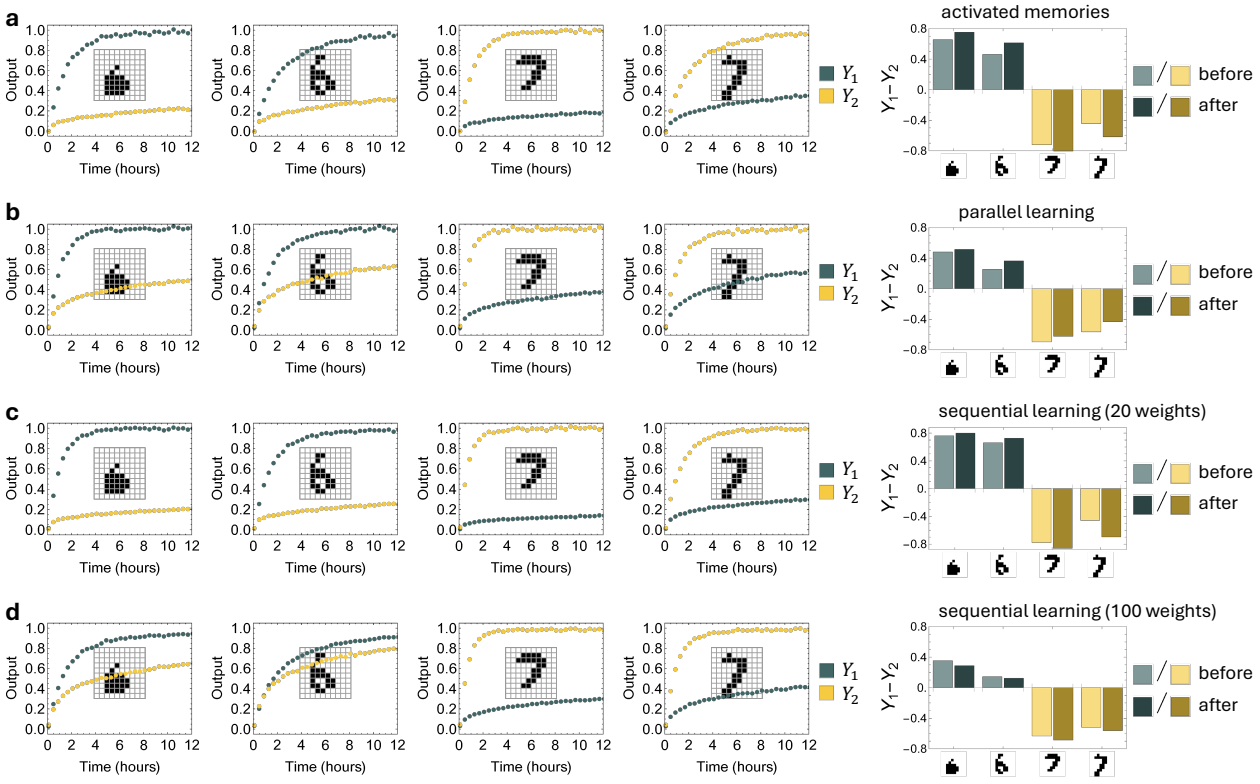


Fig. S26 | Improved performance in a 100-bit learning neural network with adjusted annealing ratio and input concentration. a-d, Fluorescence kinetics experiments comparing the effect of the adjustments for activated memories (**a**), parallel learning (**b**), and sequential learning with 20 (**c**) or 100 (**d**) inhibited weights. Bar charts show the difference between two outputs at 12 hours for each test pattern in the experiment after the adjustments (darker colors) compared to that before the adjustments (lighter colors) shown in Figs. S22e, d, h, and c, respectively. Inhibited weights were annealed with a 20% excess bottom strand before PAGE purified. Inhibited activators were annealed with a 5% excess bottom strand before PAGE purified. Label was at 5×. Input was at 0.5×. 1× = 50 nM. Note that experiments of sequential learning with 20 weights shown in c are compared to an easier case of activated memories with 20 weights shown in Fig. S22h, suggesting that the actual improvement is more than what appears in the bar chart.

adjusted annealing ratio and input concentration. Label concentration was already in excess in Fig. S22 before we verified the impact of label occlusion, and thus no additional change was implemented here. Experiments showed mildly improved performance of 100-bit pattern classification with activated memories (Fig. S26a) and with memories learned in parallel (Fig. S26b). However, for sequential learning, the performance was only improved when the unused inhibited weights were absent (Figs. S26c and d).

The observations led us to a hypothesis: the second set of training patterns leaked into the first learned memory, and spuriously activated the weights that were supposed to be unused. This hypothesis explains why test patterns in the first class, digits "6", cannot be properly classified, if and only if the two classes of patterns are learned sequentially and the unused inhibited weights are present. We will investigate this hypothesis in the next two sections (Supplementary Notes 5.9 and 5.10) by focusing on the role of input leak in learning.

5.9 Input leak in learning

Input leak in learning refers to an input strand spuriously displacing the top strand of an inhibited activator and producing an activator without the presence of a label.

$$X_i + Act^*_{i,j} \rightarrow Act_{i,j}$$

This leak is expected to be largely irreversible because once released the top strand of the inhibited activator will fold into a hairpin, where the toehold for initiating the reserve reaction will be concealed.

To understand the leak, we designed a set of experiments to answer the following questions. First, the input is modified with a quencher and inhibited activator modified with a fluorophore to monitor the kinetics of learning and to detect leak. Does the fluorophore-quencher interaction affect the leak? In an earlier section (Supplementary Note 4.6) we had discovered that the fluorophore-quencher interaction on the drain in Design 1 contributed to the undesired reversibility of learning. Here, understanding the impact of the fluorophore and quencher would allow us to separate an effect that only exists in the leak detection process versus an effect that helps explain the problem observed in a 100-bit neural network where learning was not monitored. Second, does the input sequence affect the leak? Third, does the inhibited activator sequence affect the leak? Finally, can the above questions be answered with respect to each of the two phases of the leak, the initial leak that happens instantly and the long-term leak that happens gradually? As the underlying mechanisms for these two phases are different, this understanding would help us evaluate the significance of each cause and devise solutions accordingly.

The experiments involved four inputs, two classes of inhibited activators, and four distinct fluorophore-quencher pairs (Fig. S27a). To separate instant leak from gradual leak, we utilized two versions of the input strand, with and without a quencher (Fig. S27b). When the unlabeled input is added first, it is expected to "quietly" initiate the instant leak without changing the fluorescence signal. In this process, copies of the inhibited activator that have synthesis errors are expected to react with the input much faster than copies without synthesis errors. Next, when the quencher-labeled input is added after the fraction of inhibited activator with synthesis errors have been consumed, it is expected to trigger graduate leak via toeless strand displacement, resulting in fluorescence decrease.

In each of the eight combinations of input and inhibited activator (Fig. S27c), a negative control (blue trajectory) was included as a reference for minimum fluorescence change when both input and label were absent, where the inhibited activator remained at 1×. A positive control (orange trajectory) was included as a reference for maximum fluorescence change when both input and label were in excess, where the inhibited activator was fully consumed and reached 0×. The full input leak (green trajectory), including instant and gradual leak, was observed when the input was present but the label was absent. The gradual leak alone (red trajectory) was observed when the unlabeled input was used to quietly remove instant leak before the quencher-labeled input was introduced.

The worst gradual leak occurred between $Act^*_{7,2}$ and X_7 (Fig. S27c, bottom right plot). It was not caused by the input sequence alone, because the same input exhibited much less gradual leak with another inhibited activator $Act^*_{7,1}$ (Fig. S27c, top right plot). Moreover, it was not caused by the fluorophore-quencher interaction alone, because the same fluorophore-quencher pair (ATTO590 and RQ) exhibited much less graduate leak in a different combination of input and inhibited activator (Fig. S27c, third plot in the top row). Unlike the fluorophore-quencher interaction observed earlier

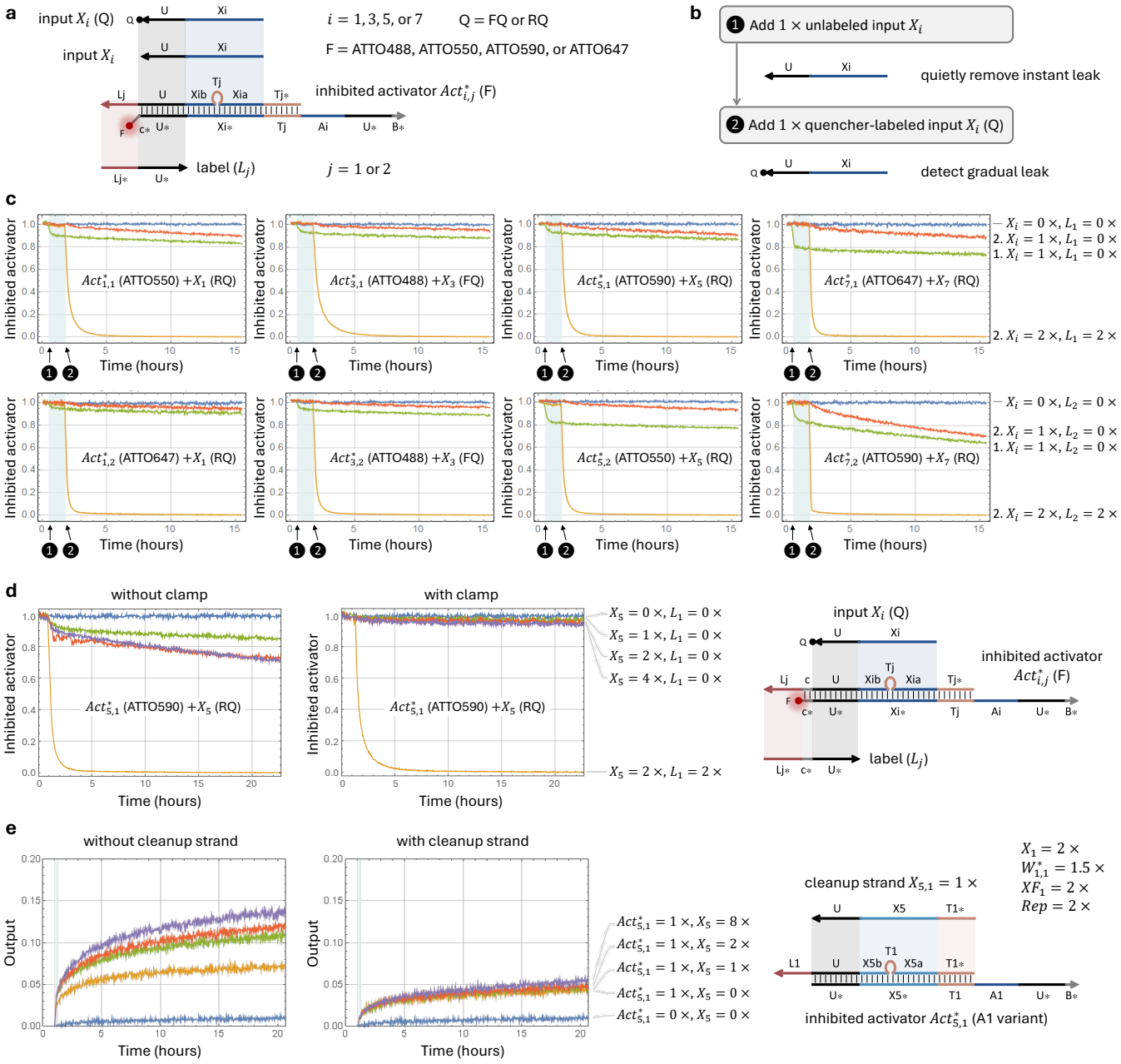


Fig. S27 | Input leak in learning. **a,** Domain-level diagrams of species involved in measuring the input leak. Four input strands with distinct Xi sequences, two classes of inhibited activators with distinct Tj sequences, and four fluorophore-quencher pairs are used to evaluate their impact on the leak. **b,** A two-step mechanism for separating the detection of gradual leak from instant leak. **c,** Fluorescence kinetics experiments measuring the input leak in learning with eight distinct combinations of input X_i and inhibited activator $Act^*_{i,j}$, where $i = 1, 3, 5,$ or 7 and $j = 1$ or 2. Except for X_3, fluorophore-quencher pairs are different for input X_i and inhibited activators $Act^*_{i,1}$ and $Act^*_{i,2}$, allowing for the separation of impact from input sequence and fluorophore-quencher interaction. X_i in the trajectory labels indicates quencher-labeled input, whereas number 1 or 2 indicates whether the input was added at the first or second step around 30 minutes or 2 hours, respectively. In the red trajectories, unlabel input was added at the first step, which quietly triggered instant leak without showing any fluorescence change. **d-e,** Experiments demonstrating leak reduction with a clamp (**d**) and a cleanup strand (**e**). For the cleanup strand, input leak in learning was measured by a downstream weight motif, where leak product $Act_{5,1}$ (A1 variant) functions as activator $Act_{1,1}$ and allows for the input X_1 to react with the activated weight $W_{1,1}$, resulting in an output that can be detected by the reporter.

(Supplementary Note 4.6), the fluorophore and quencher here are separated by a 2-nt spacer (c∗ on the bottom strand of the inhibited activator) and expected to have a weaker interaction. We thus hypothesized that the Tj toehold sequence played a role here. Even though T1 and T2 were designed to have a very similar binding energy ($\Delta G = -9.87$ kcal per mol for T1 = TCTTTCA and $\Delta G = -9.83$ kcal per mol for T1 = CTCTATT), the slightly weaker T2 toehold resulted in a faster dissociation of the top strand in $Act^*_{7,2}$, completing the toeless strand displacement leak initiated by the input.

The worst instant leak occurred between $Act^*_{7,1}$ and X_7 (Fig. S27c, top right plot) and between $Act^*_{5,2}$ and X_5 (Fig. S27c, third plot in the bottom row). There are no correlations between these two examples, because the input sequences, the Tj toehold sequences, and the fluorophore-quencher pairs are all different. This observation could be explained by that the cause of instant leak is synthesis errors, which may vary randomly from batch to batch in chemically synthesized DNA strands (for example ordered from Integrated DNA Technologies).

The main conclusions we had here were the following. First, input leak in learning was significant: up to 0.2× instant leak in less than 30 minutes and up to 0.3× gradual leak in 14 hours were observed. Second, no clear patterns were observed for the cause of the instant leak. Third, gradual leak depends on both the Xi sequence and the Tj sequence, and a stronger Tj toehold may help reduce the gradual leak.

Clamps have been commonly used to reduce gradual leak caused by toeless strand displacement.[2,27,28] We originally had designed a clamp in the learning motif, but upon earlier characterization experiments we falsely concluded that the clamp was not necessary (Supplementary Note 5.2). Experiments here revealed that the leak is sequence dependent and thus specific characterization experiments may not effectively inform system behavior for increasing system complexity. Indeed, reintroducing the clamp in the learning gate significantly reduced the leak (Fig. S27d). Even at a 4-fold higher input concentration than the experiments shown in Fig. S27c, leak remained roughly the same as when the input was absent.

We further explored using a cleanup strand to reduce instant leak (Fig. S27e). The function of the cleanup strand is similar to that of excess top strand in the inhibited activator (Supplementary Note 5.6). After the inhibited activator is annealed and before it is PAGE purified, the cleanup strand can be added, a fraction of which that are well synthesized will replace the top strand that have synthesis errors, especially truncations on the 5' end. Unlike the top strand itself, the cleanup strand does not have the Lj domain and thus is expected to have less spurious binding to the bottom strand during PAGE purification. Furthermore, the cleanup strand also lacks the Tj bulge, allowing it to more effectively displace the top strand that have synthesis errors near the bulge. With this design, any leftover cleanup strands must be removed by PAGE purification, otherwise they will function as inputs and corrupt both training and test patterns. For a quick evaluation without PAGE purification, we created an A1 variant of $Act^*_{5,1}$ – it can be used to test input X_5 leak, where the leak product functioning as $Act_{1,1}$ can be detected by a downstream weight motif that utilizes a different input X_1. Experiments showed that after the inhibited activator was treated with the cleanup strand, the instant leak for up to 8× input was reduced to roughly the same as when the input was absent.

Combining the above two approaches for reducing instant and gradual leak, we will evaluate their impact on a 100-bit learning neural network in the next section (Supplementary Note 5.10).

5.10 Further improvement in a 100-bit learning neural network

The input leak discussed in the previous section (Supplementary Note 5.9) not only allows for the second set of training patterns to leak into the first learned memory, but also allows for test patterns to leak into both leaned memories, worsening the pattern classification performance even when the two memories are trained in parallel. After reintroduction of clamps into the inhibited activators and treating them with cleanup strands before PAGE purification, the performance of both parallel and sequential learning was much improved (Fig. S28). This set of experiments was the first time that we observed a clear ($\geq 0.4\times$) on-off separation in the pair of outputs for all test patterns. Moreover, the performance of sequential learning (Fig. S28d) is now comparable to that of parallel learning (Fig. S28c), and the best and worst on-off separations for both classes of test patterns are now similar to each other, resolving the strong bias observed earlier (Fig. S22a-c).

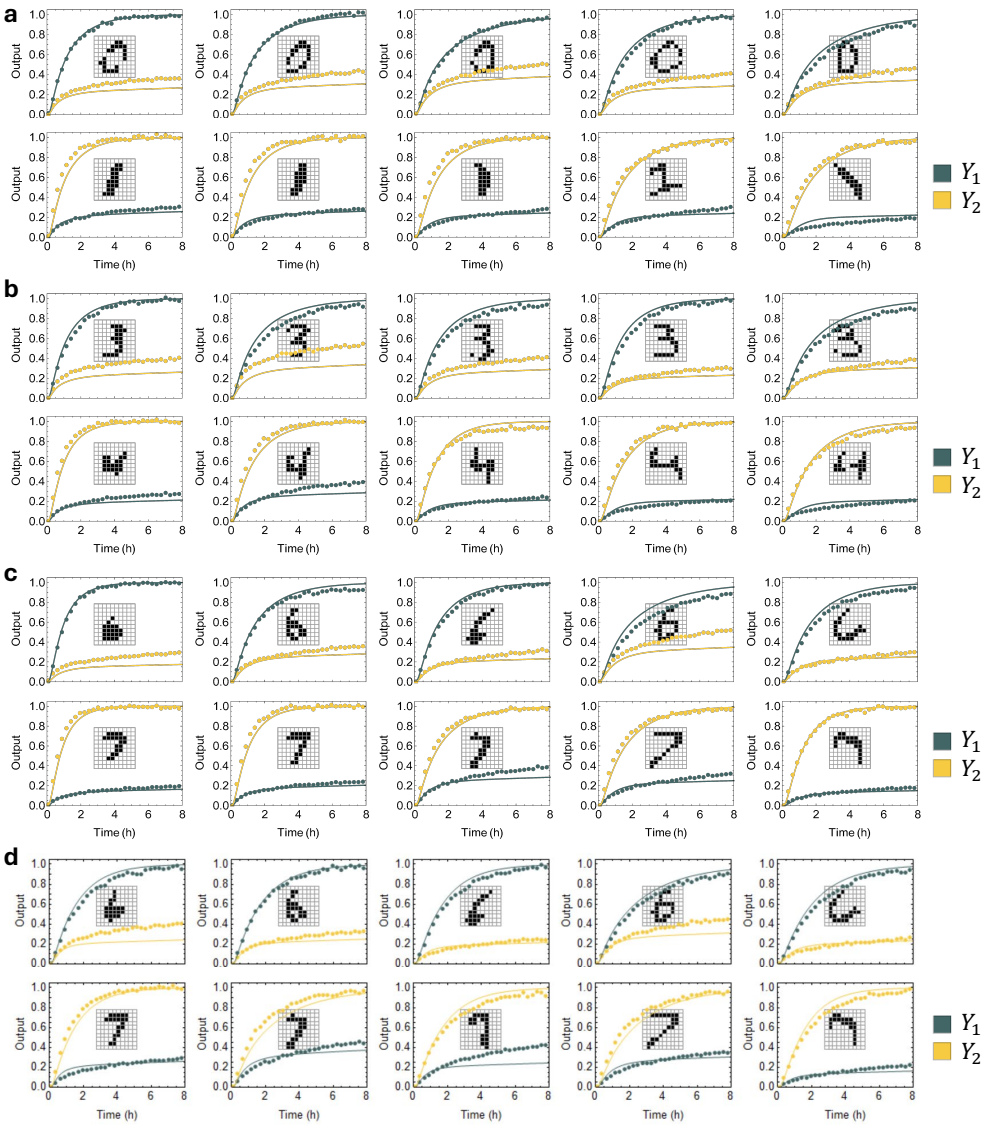


Fig. S28 | Further improvement in a 100-bit learning neural network with clamps and cleanup strands. **a-d,** Fluorescence kinetics experiments of pattern classification after trained in parallel with MNIST handwritten digits "0" and "1" (**a**), "3" and "4" (**b**), or "6" and "7" (**c**), and after trained sequentially with "6" and "7" (**d**).

The use of cleanup strands adds an incubation step after annealing and before PAGE purification of inhibited activators. After seeing the improvement of system performance with clamps and cleanup strands, we opted to evaluate the impact of cleanup strands alone in a 100-bit learning neural network (Fig. S29), as a justification of the increased complexity in the experimental procedure. In these experiments, all inhibited activators had clamps, allowing us to focus on comparing the system performance without and with the cleanup procedure in two distinct orders of learning the same classes of patterns "6" and "7". Experimental results suggested slightly improved on-off separation for the first learned memory and slightly worse separation for the second learned memory when cleanup strands were used (Figs. S29e and f). Without the cleanup procedure, a slightly stronger bias was observed when patterns "6" were learned first (Fig. S29g). With the cleanup procedure, roughly the same system performance was observed regardless of the learning order (Fig. S29h).

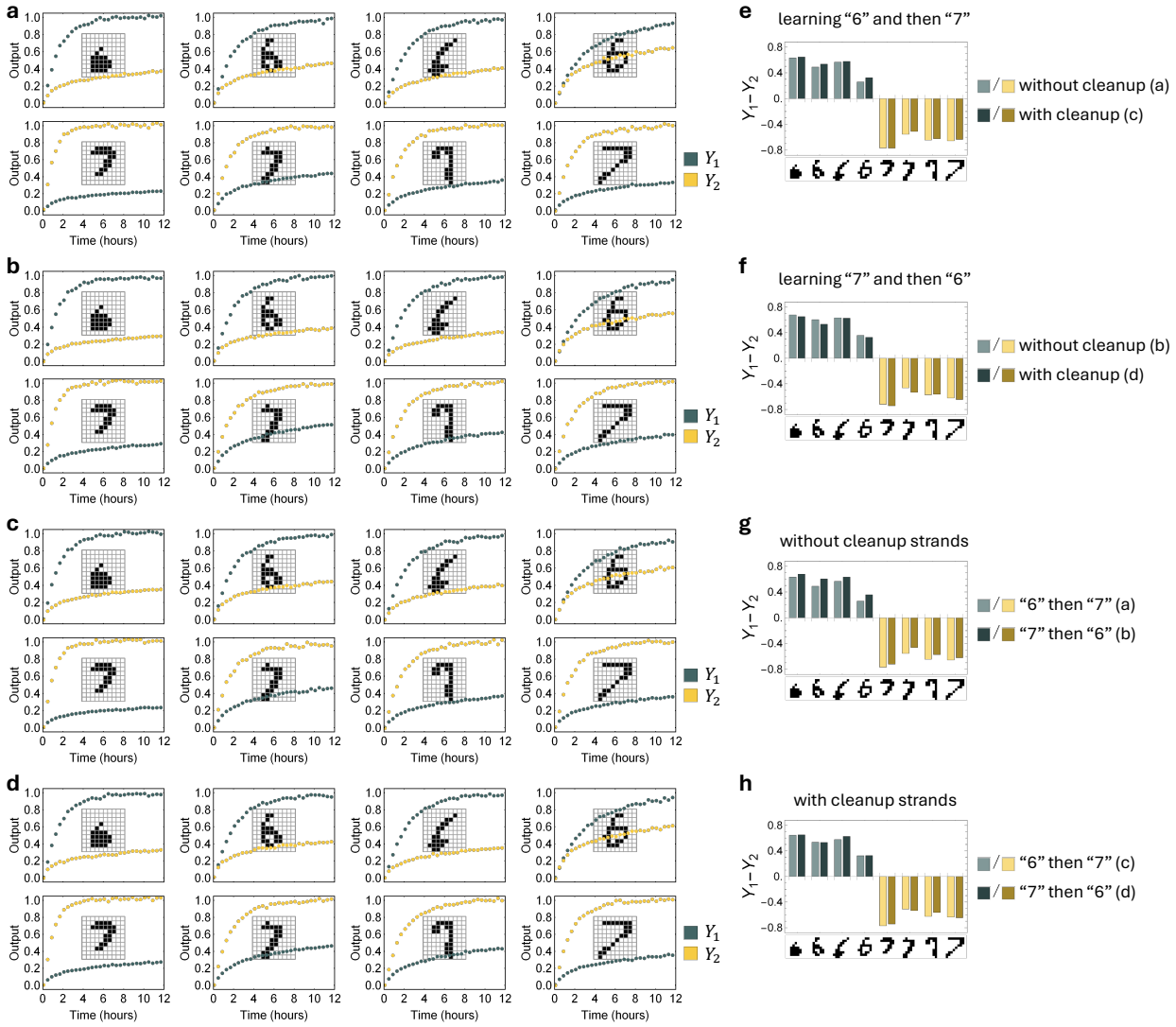


Fig. S29 | Separated evaluation of cleanup strands in a 100-bit learning neural network. a-d, Fluorescence kinetics experiments of pattern classification after learning "6" first and then "7" (**a** and **c**) or learning "7" first and then "6" (**b** and **d**), using inhibited activators without (**a** and **b**) or with (**c** and **d**) a cleanup treatment. **e-h,** Bar charts comparing the impact of cleanup strands (**e** and **f**) and learning order (**g** and **h**). Each bar shows the difference between two outputs at 12 hours for each test pattern. All experiments here used inhibited activators with clamps.

Overall, the observed impact of cleanup strands on system behavior was minor, presumably due to the intrinsic robustness of the winner-take-all architecture, where a small amount of noise in both memories can be cancelled out in the annihilation process. To keep the experimental procedure simple, we stopped using cleanup strands in later experiments.

The experiments with two learning orders highlighted a small yet remaining bias for the second learned memory. We will attempt to better understand the system bias in the next section (Supplementary Note 5.11).

5.11 System bias

There are two known sources of bias in the system. First, we observed that activating weights in memory 2 was generally faster than activating weights in memory 1 (for example, comparing $W^*_{i,2}$ to $W^*_{i,1}$ in Extended Data Fig. 4d). This rate difference is presumably due to the difference in strength of the 5-nt exposed part of toehold Tj on the inhibited weight (Fig. S30a, left). We designed the 7-nt toehold T1 and T2 to have very similar binding energies. However, when the clamp cj was incorporated to reduce spurious activation caused by inhibited activators, it covered up 2 nucleotides of Tj and created an imbalance in the remaining 5 nucleotides – the exposed part of T1∗ now has one G whereas that of T2∗ has two Gs. We estimated that the toehold dissociation rate for T1 was roughly 5 times faster than that for T2 in binding to the double-stranded activators produced from learning (Supplementary Note 2.1). Second, we measured that signal amplification for output 2 was slightly (1.4×) faster than that for output 1 (Fig. S30a, right). This rate difference is presumably due to the difference in branch migration sequences of the two amplification gates, despite that a universal toehold was used.

To evaluate whether the known biases in the two distinct layers of the neural network explain the observed system-level bias in pattern classification after leaning, we explored four combinations of reaction pathways for storing and classifying the same 100-bit pattern with distinct memory and output identities (Fig. S30b). In the first case, handwritten digit "6" was learned into memory 1 and connected with output 1, resulting in a slower weight activation and a slower signal amplification. In the second case, "6" was learned into memory 1 but connected with output 2, resulting in a slower weight activation but a faster signal amplification. In the third case, "6" was learned into memory 2 and connected with output 2, resulting in a faster weight activation and a faster signal amplification. In the last case, "6" was learned into memory 2 but connected with output 1, resulting in a faster weight activation but a slower signal amplification. Overall, we expected that test patterns in class "6" would be best classified in the third case (faster-faster) and worst classified in the first case (slower-slower). A stronger system bias would arise in the first and third cases, whereas the second and fourth cases (slower-faster and faster-slower) could provide improved system performance.

Another factor that we considered was incomplete production of activators in learning. In characterizing the learning motif, we observed that activator concentration was generally lower than that of the input (Fig. 2e). This is unsurprising because signal loss was expected in stoichiometric reactions where synthesis errors in reactant molecules would prevent fully effective release of product molecules. To account for the incomplete production of activators, we explored a 1.25-fold higher concentration of molecules yielded from learning, which would correspond to 1.25× learned weights if full production is assumed. We also included lower concentrations of learned weights to evaluate their impact on the system bias.

Experimental results agreed with our expectations, while providing quantitative conclusions on how the two sources of bias affect the system behavior (Fig. S30c). The best on-off separation (approximately 0.55× in 8 hours) was observed when "6" was learned into memory 2 and connected with output 2 at the highest concentration (1.25×) of learned weights (top and bottom trajectories in the bottom right plot). Conversely, the worst on-off separation (approximately 0.15× in 8 hours) was observed when "6" was learned into memory 1 and connected with output 1 at the lowest concentration (0.5×) of learned weights (middle two trajectories in the top left plot). Swapping the wires between the weighted sum and winner-take-all layers resulted in mild improvement of the system bias, reducing the on-off separation in the favored memory (middle two trajectories in

all plots in the bottom row) while simultaneously increasing the on-off separation in the unfavored memory (top and bottom trajectories in all plots in the top row).

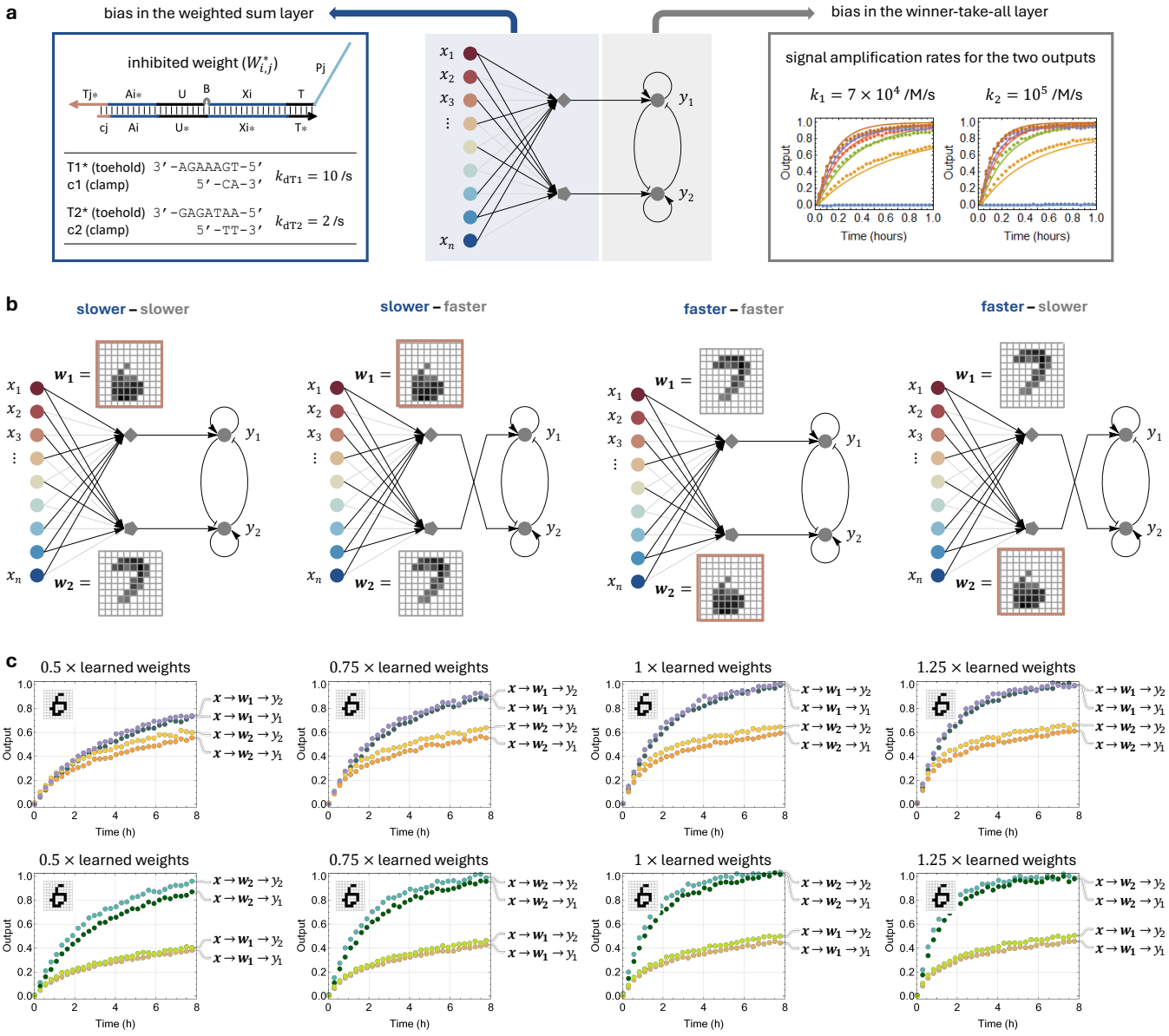


Fig. S30 | System bias in a learning DNA neural network. a, Two types of biases in the weighted sum layer (highlighted in blue) and winner-take-all layer (highlighted in gray) of the neural network. **b,** Evaluating the impact of each type of bias using four distinct combinations of reaction pathways. Weight matrix highlighted in orange, memory of MNIST handwritten digit "6", is involved in all four specified combinations of expected relative speed in the two layers. "slower" and "faster" in blue represent the expected bias in the weight sum layer, whereas "slower" and "faster" in gray represent the expected bias in the winner-take-all layer. **c,** Fluorescence kinetics experiments for the four combinations of reaction pathways with varying total concentration of learned weights. A single test pattern "6" was used to evaluate the bias in all combinations and across all concentrations of learned weights. Plots in the top row show experiments with "6" learned into memory 1 and "7" learned into memory 2 in parallel (first two cases in b). Plots in the bottom row have the two memories swapped (last two cases in b). Trajectory label $\boldsymbol{x} \to \boldsymbol{w}_j \to y_k$ indicates the pathway for which the input pattern $\boldsymbol{x}$ is compared with memory $\boldsymbol{w}_j$ for producing output y_k. If the two wires between the weighted sum layer and winner-take-all layer are swapped, $j \neq k$, otherwise $j = k$, where $j, k \in \{1, 2\}$.

Experimental results also suggested that if the learned weights was only half the expected amount, the system bias would become significantly worse (left two plots in Fig. S30c). This performance decline can be explained by the decreased difference and thus more challenging competition between the two weighted sum species. Fortunately, even though the effective concentration of the learned weights may be lower than the ideal situation when full production of the activators is assumed, the system performance remained similar to that with a higher weight concentration (comparing the plots in the third column to that in the last column in Fig. S30c). Like the possible explanation for the observed impact of the cleanup strands at the system level (Supplementary Note 5.10), this phenomenon may be due to the intrinsic robustness of the winner-take-all architecture – when the difference between the two weighted sum species is large enough, small changes in the species concentrations would not affect the overall pattern classification performance.

To summarize, the rate difference in memory activation played a more significant role than the rate difference in output amplification, giving rise to a system-level bias favoring the class of patterns learned into memory 2. Paring up slower memory activation with faster output amplification showed mild improvement on the classification performance, but we decided that the improvement was not significant enough to justify the change and thus the approach of swapped wires was generally not used in later experiments.

5.12 Good teacher

In all learning experiments discussed so far, we used a mixture of 100 training patterns while keeping only the input strands that corresponded to the most common 20 bits with the largest sums. This "trimming" mechanism is conceptually similar to a sorting plus thresholding function where the signals are sorted and the threshold value is set between the k-th and $(k+1)$-th signal values, or a k-winner-take-all function where the zero or one output values are multiplied by the input values. Neither of these functions have been implemented in DNA strand displacement circuits. Instead of developing a molecular implementation for the trimming mechanism, we turned to investigate a simpler alternative solution by adding constraints to the training dataset.

First, we compared the expected pattern classification performance with and without the trimming mechanism in learning (Fig. S31a). When the most common 20 bits from the average of 100 training patterns are used as weights (shown in gray boxes), test patterns are spread out in the weighted sum space (gray points). Roughly 84% of the tests are on or outside of the 20% margin to the diagonal line, indicating that the difference between the two weighted sums for comparing a test pattern to the two memories is sufficiently large for the pattern to be correctly classified via experiments in a test tube. By contrast, when all bits from the average of 100 training patterns are used as weights (shown in dark green and yellow boxes), test patterns become more clustered and closer to the diagonal line in the weighted sum space (dark green and yellow points). The expected pattern classification performance (percentage of tests with a margin ≥ 0.2) is significantly decreased to roughly 44%.

We then asked: how does the pattern classification performance change with the size of the training dataset when trimming is not allowed? When only a single training pattern is used, the test performance varies widely from below 5% to above 80% (Fig. S31b, training dataset size = 1). This is unsurprising because the best examples of a "6" or "7" look similar to a trimmed version of the averaged training patterns, whereas the worst examples look nothing like a "6" or "7". As the training dataset size increases, the medium performance improves mildly at first and then stays roughly the same (white line within each box in the box-and-whisker plot) while the distribution becomes tighter (above 25% to below 65% for training dataset size = 100). Naturally, the tighter distribution is because the probability of examples from a randomly chosen dataset are either all good or all bad is smaller when the size of the dataset is larger. From the above analysis we concluded that too many training patterns would not allow for good pattern classification performance, whereas too few training patterns would only allow for good performance if the patterns are heavily handpicked. To strike a balance, we chose the size of 10 patterns in each training dataset.

We ranked the training patterns based on their positions in the weighted sum space using the average of all training patterns as weights. A larger distance to the diagonal line is better and a smaller distance is worse. When there are a hundred examples of "6" and "7" in the training dataset, the average of them look the best if all examples are from the top 10% of the training pool (Fig. S31b, bottom two rows of averaged examples). When examples are drawn from a lower ranked decile group, the averaged patterns become more smeared. Moreover, the divergence appears to accelerate toward the bottom few decile groups. Inspired by this observation, we asked: how does the pattern classification performance change if a fraction of patterns can be removed from the training pool? Using the ranking of the training patterns, we analyzed the test performance for training datasets of 10 patterns randomly drawn from a training pool where the bottom decile group is removed for each of the subsequent trial (Fig. S31c). The medium performance increases gradually as the

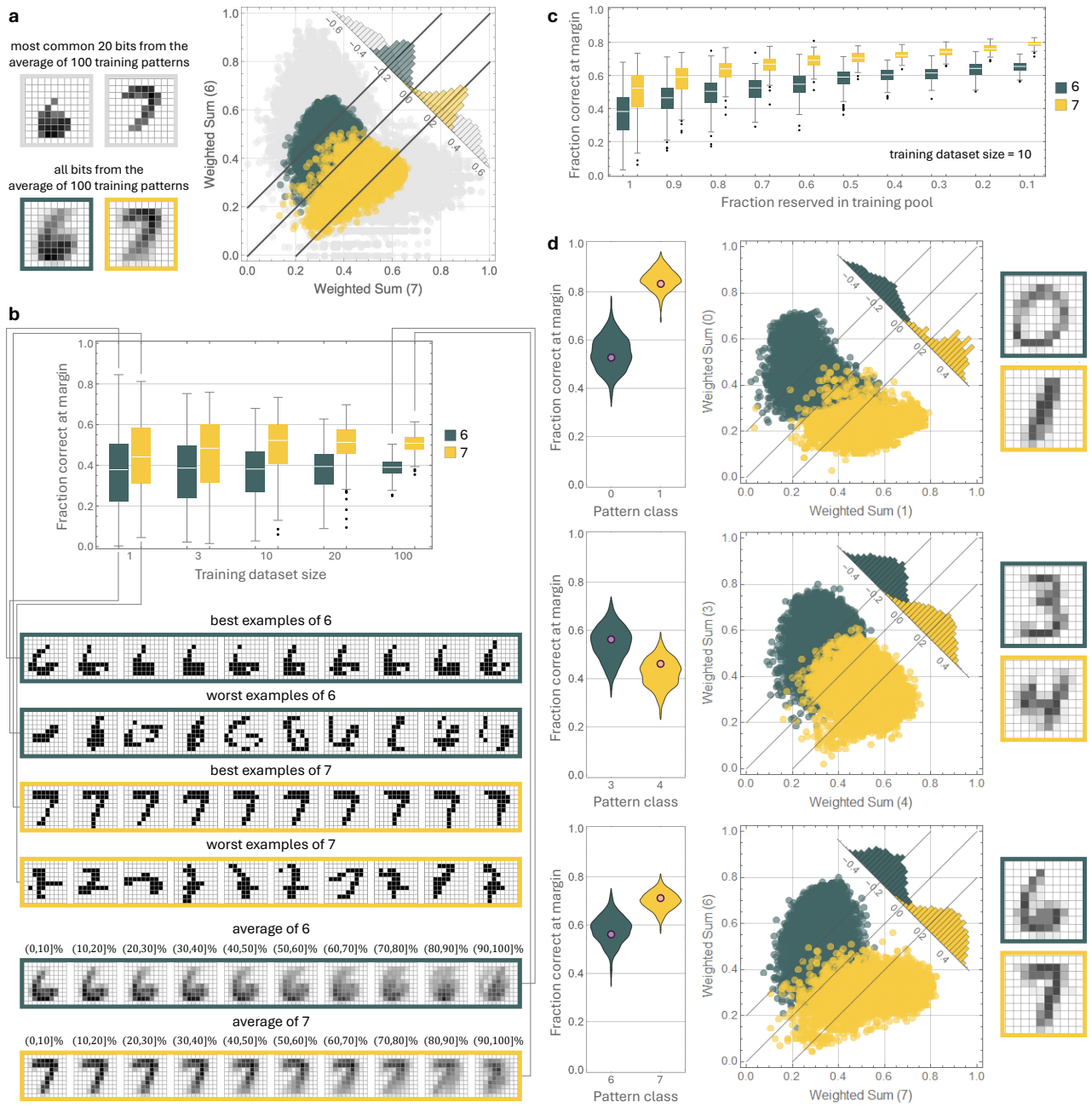


Fig. S31 | Good teacher approach for selecting a training dataset. a, Analysis of all test patterns in their weighted sum space, using the average of 100 training patterns as weights, keeping the most common 20 bits (weights highlighted in a gray box) or all bits (weights highlighted in a dark green or yellow box). Diagonal lines highlight an area with a 20% margin to equal weighted sums, test patterns outside of which are deemed experimentally feasible for correct classification. Distributions of the margin are shown near the top right corner. **b-c,** Box-and-whisker plot showing test performance for 200 random samples of a training dataset with a varying size (**b**) or drawn from a varying fraction of patterns in the training pool (**c**). White line within the box represents the medium. Top and bottom edges of the box represent the 75% and 25% quartile, respectively. Upper and lower fences of the whiskers represent maximum and minimum values excluding outliers. Individual points outside the range of the whiskers represent outliers. **d,** Violin plot showing test performance for 1000 random samples of a training dataset with 10 patterns drawn from the top 50% of the training pool for handwritten digits "0" and "1", "3" and "4", or "6" and "7". A representative sample in each class is highlighted in purple, and the weighted sum analysis is shown using weights resulted from these samples of training datasets.

fraction of patterns reserved in the training pool decreases, while the most significant improvement occurs at the beginning when the worst 10% of patterns are removed. Based on this result, we chose to use the top 50% patterns in the training pool, which on average provides above 60% pattern classification performance for the two classes of handwritten digits "6" and "7".

Combining the two constraints together, we now have a relatively small size of training dataset randomly selected from a pool where the worse-than-average examples are removed. We named it a "good teacher" approach and applied it to generate six training datasets (Fig. S31d) that will be used in later learning experiments.

5.13 Sample evaporation

From the first attempt of learning and testing in a 100-bit DNA neural network (Fig. S22) to the final experiments shown in the main paper (Figs. 4 to 6), the time span was over two years. Within this period of time, we were not always successful reproducing the same learning results. One variable was the quality of PAGE purified complexes. As discussed in Supplementary Note 5.6, loading a larger volume of sample per lane helps improve the yield but could also result in reduced resolution and worse separation of desired and undesired complexes. Similarly, cutting out a larger area of sample helps improve the yield but could also lead to lower purity of the complex. Increasing the gel running time could help improve separation but also requires more repeated times to refresh the running buffer and clean the magnesium buildup on the exposed wires of the gel apparatus. Carefully balancing the above conditions is necessary for obtaining high yield and high quality of purified complexes. Moreover, DNA degradation occurs much faster in magnesium than in TE buffer, and thus the quality of annealed and purified complexes stored in magnesium may decrease significantly in just a few months. When we had problems reproducing the experimental results, we typically first consider re-annealing and re-purifying the complexes including inhibited activators and inhibited weights. However, at one time the problem persisted, triggering an investigation on the quality of stock strands.

We measured the concentration of each DNA strand on a deep well stock plate provided by Integrated DNA Technologies, IDT (Fig. S32a, left plot). Only 32% of the stock strands was within 5% of the expected concentration of 100 μM. 31% was below 95 and above 90 μM, 34% was below 90 and above 80 μM, and 3% was below 80 μM. Upon discussions with IDT, we concluded that their DNA quantification procedure was not nearly as strict as ours, allowing for substantial errors. These errors may not be problematic for less quantitative studies using synthetic DNA, but they play in an important role in constructing DNA circuits where the concentrations of molecules directly impact the function of computation. Thus, we opted to use our own concentration measurements for most of the experiments reported in this work.

Roughly 8 months later, we repeated the measurements and found that all stock strands had their concentrations increased by 1.91 ± 1.01 μM (Fig. S32a, middle and right plots). In the worst case, a 8.8 μM increase was observed in the strand stored in well H2. We hypothesized that the increase in concentration was due to sample evaporation, which depends on how tight the seal is at specific well positions.

Given the scale of the experiments, we used an acoustic liquid handler to prepare samples such as mixtures of inputs for training and test patterns. For these automated procedures, strands were transferred from a deep well stock plate into an Echo source plate and stored there for later experiments. The concentration measurements of samples stored in an Echo source plate revealed striking changes in 8 months (Fig. S32b). Both increase and decrease in concentration were observed across the 91 strands. Compared to the IDT stock plate, the change in Echo plate had both a larger average and a wider distribution (6.24 ± 13.97 μM). For 65% of the strands, the change was an increase below 10 μM. However, in the worst case, a 48.2 μM increase and a 36.5 μM decrease were observed. Interestingly, these two extreme changes occurred at two well positions adjacent to each other (F7 and G7). In fact, this pattern of neighboring wells with increased (colored in red) and decreased (colored in blue) concentrations was common across the plate.

Utilizing the above clue, we came up with the following hypothesis. When samples evaporate, condensation forms on the sealing tape above individual wells. The condensation can travel be-

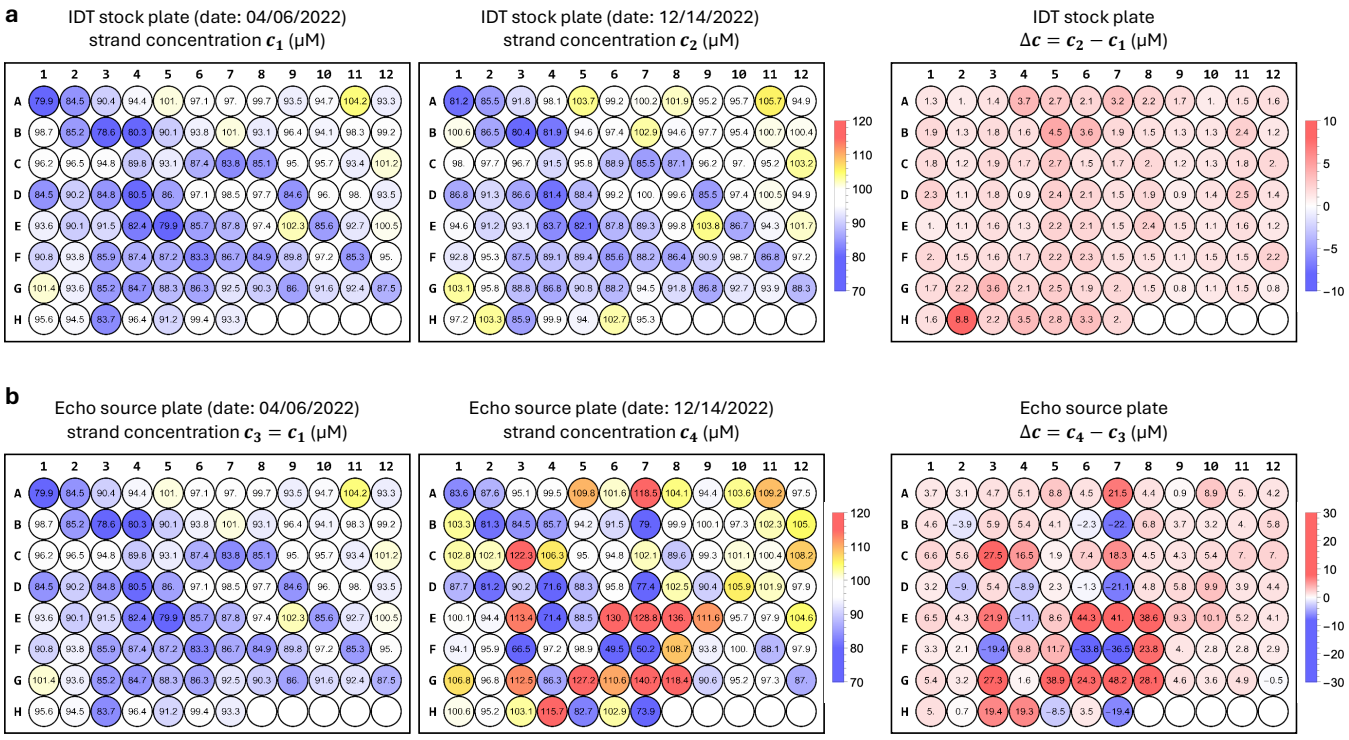


Fig. S32 | Sample evaporation in plates. a-b, Measured changes of strand concentrations on a deep well stock plate (**a**) or an Echo source plate (**b**). Strand concentrations were first measured shortly after they were synthesized, normalized to 100 μM in IDTE buffer with full yield, and delivered in a 96-well deep well plate provided by Integrated DNA Technologies (IDT), and then measured again in roughly eight months. A 60 μL sample of each strand was transferred from the IDT stock plate to a 96-well region on a 384-well Labcyte Echo source plate. The first set of measurements was done on the same day of the sample transfer and thus the concentrations for the IDT stock plate and Echo source plate are assumed to be the same. Concentration measurements were performed using UV absorbance on a Take3 microvolume plate on a BioTek Synergy H1 plate reader, which allows for high-throughput measurements of up to 16 samples in parallel. 14 out of the 91 measurements were repeated on a NanoDrop spectrophotometer. The difference was determined to be sufficiently small (0.13 ± 1.28 μM). A positive or negative number indicates increased or decreased strand concentration comparing the measurements on the two dates, respectively.

tween wells via the surface of the sealing tape, especially when gentle mixing was applied to create homogeneous samples. This hypothesis was supported by a 5 to 8 μL volume displayed for an well that was supposed to be empty (H9) in three repeats of Echo plate survey. We found a droplet in that well, and measured its concentration on a NanoDrop. The concentration was approximately zero, indicating no DNA but only buffer. This evidence agreed with the possibility of condensation leaking into neighboring wells, causing increased concentration in one well and decreased concentration in the other. We thus concluded that long-term sample storage in Echo source plate is problematic and should be avoided. For critical experiments, we opted to use Echo source plate with freshly transferred samples. For less critical experiments, we improved the sealing method by using an additional layer of Parafilm wrapped around the edges, and used pipet mixing instead of shaking or vortexing when gentle mixing is needed.

Learning from the above results, we established routine concentration survey as a method to evaluate the quality of strands in IDT stock plates and Echo source plates, guiding our decision to make fresh sample transfers or order new strands when severe changes in concentrations are

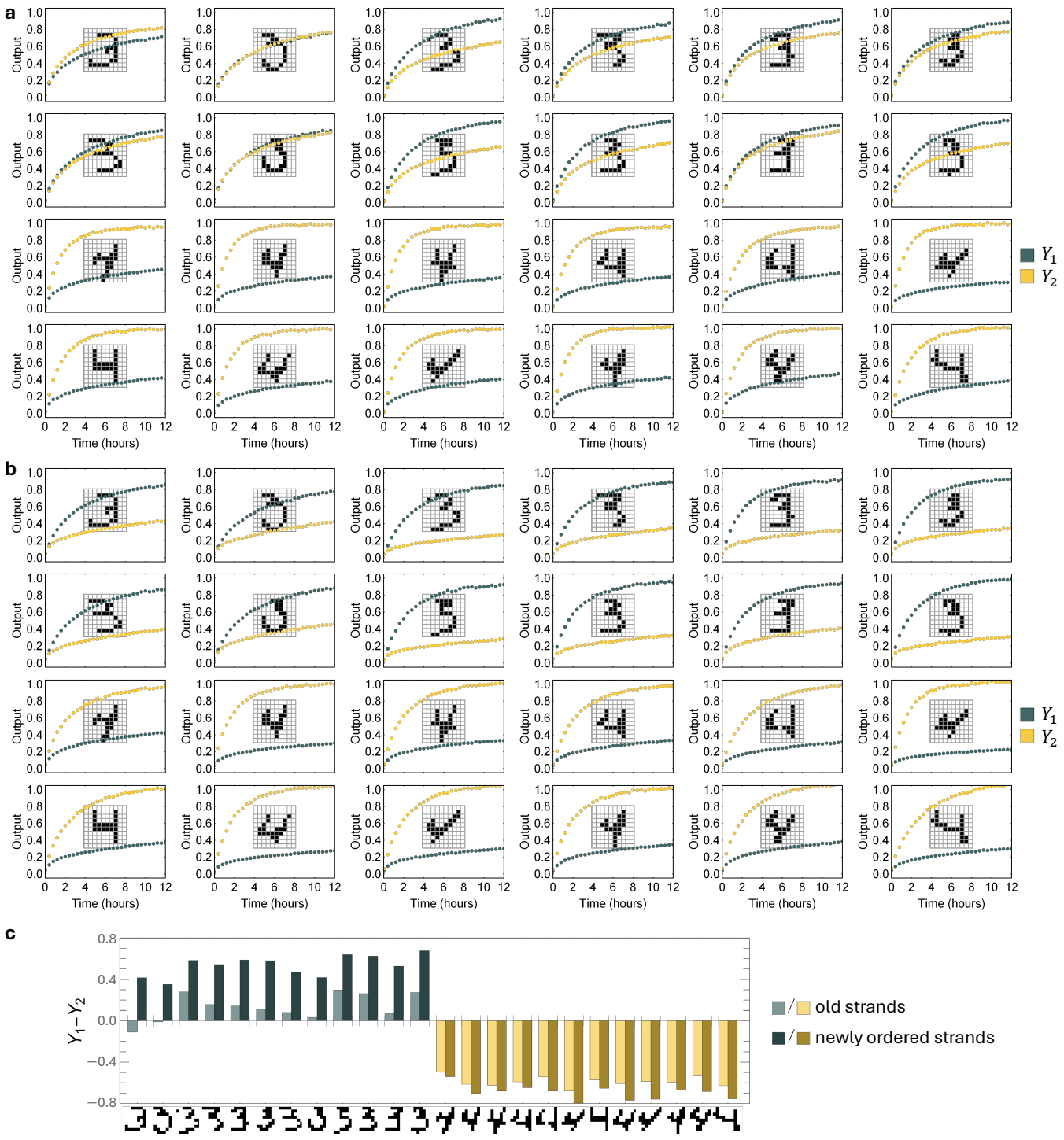


Fig. S33 | Pattern classification in a 100-bit learning neural network. a-b, Fluorescence kinetics experiments with 12 test patterns from each class of handwritten digits "3" and "4" using old strands (**a**) or newly ordered strands (**b**) for all 200 inhibited activators involved in learning. Two training datasets were chosen utilizing the good teacher approach. The average of 10 patterns in each dataset is shown in the middle plot in Fig. S31d. 12 representative test patterns in each class were chosen randomly from each of 12 evenly divided regions in the weighted sum space. **c,** Bar chart showing the difference between the two outputs at 12 hours for each of the 24 test patterns, comparing the old strands (lighter colors) with the newly ordered strands (darker colors).

observed. These changes indicate not only evaporation but also possible contamination in samples.

In one case, our attempt of pattern classification in a 100-bit neural network failed after training with a dataset using the good teacher approach (Supplementary Note 5.12). All test patterns in class "3" resulted in poor on-off separation between the pair of outputs (Fig. S33a). This result was unexpected because the same set of molecules were used in the earlier successful demonstration of pattern classification (Fig. S28). The good teacher approach was supposed to eliminate the undesired trimming step in creating the training samples but not affect the experimental results. Upon measuring the concentrations of DNA strands involved in the 100-bit experiments, we found mysterious concentration increase in some plates and decrease in the others for a batch of IDT stock plates ordered at the same time. After re-ordering all strands in those plates, the performance of the 100-bit learning neural network was restored (Fig. S33b). All test patterns from both classes resulted in clear on-off separations within 12 hours (Fig. S33c).

We have discussed earlier that for ambitious research projets that take a long time, it is important that we not only learn from each failed experiment, but also fit all pieces of the knowledge together into one complete picture (Supplementary Note 4.9). Here, we further emphasize the importance of systematic approaches for differentiating sample problems from design problems. Otherwise our energy and attention would be dissipated instead of being concentrated, to paraphrase Sir Arthur Conan Doyle.[29]

6 DNA sequences

Table S1: DNA sequences of input strands.

Name	Sequence
X[i1]	CAACTTCCCACACTT TTTAATTTT
X[i2]	CATCTAACCAACTTA TTTAATTTT
X[i3]	CACCTAAACAATACT TTTAATTTT
X[i4]	CATTACATCACAATC TTTAATTTT
X[i5]	CAACCTTACATTATC TTTAATTTT
X[i6]	CAATCCATCATCTTA TTTAATTTT
X[i7]	CATCACTACATCCAC TTTAATTTT
X[i8]	CATCCCAACATACCT TTTAATTTT
X[i9]	CAATCTCCCAACCCA TTTAATTTT
X[i10]	CATTTCCACAACTTT TTTAATTTT
X[i11]	CATCTTTTCACCACT TTTAATTTT
X[i12]	CAATTACTCAAACTC TTTAATTTT
X[i13]	CACTATCACAACCTC TTTAATTTT
X[i14]	CACAACAACACACCC TTTAATTTT
X[i15]	CAATCATACATATCC TTTAATTTT
X[i16]	CACCCTTTCATACTA TTTAATTTT
X[i17]	CATCCACTCAATCCC TTTAATTTT
X[i18]	CATACTCACATAATC TTTAATTTT
X[i19]	CATACACCCACTTCT TTTAATTTT
X[i20]	CACCTTCTCATATCT TTTAATTTT
X[i21]	CATTATTTCAACCCT TTTAATTTT
X[i22]	CATTAATACACTTCC TTTAATTTT
X[i23]	CATTTACACACACAT TTTAATTTT
X[i24]	CATAATTCCATCTTC TTTAATTTT
X[i25]	CAACACTTCACTCCT TTTAATTTT
X[i26]	CATCTATCCACTATC TTTAATTTT
X[i27]	CACTCCTACAAATTA TTTAATTTT
X[i28]	CACTACCTCATACCC TTTAATTTT
X[i29]	CATACCTACACTCTA TTTAATTTT
X[i30]	CACTCACACACCTTC TTTAATTTT
X[i31]	CAACTACACATTCTA TTTAATTTT
X[i32]	CATATTCACAAACCA TTTAATTTT
X[i33]	CACACATTCAAAACT TTTAATTTT
X[i34]	CACTTTTCCACTTTA TTTAATTTT
X[i35]	CAAATACCCACCCAC TTTAATTTT
X[i36]	CACCACATCATTATT TTTAATTTT
X[i37]	CATCCCTTCAATATA TTTAATTTT
X[i38]	CAATTCATCAACAAC TTTAATTTT
X[i39]	CATAAACACATCCCT TTTAATTTT
X[i40]	CACATATACACAAAC TTTAATTTT
X[i41]	CATCCTATCACTTTC TTTAATTTT
X[i42]	CAAAACCACAATCAC TTTAATTTT
X[i43]	CATCATAACAACACC TTTAATTTT
X[i44]	CACAAATTCATTCAC TTTAATTTT
X[i45]	CATTATCCCATAACT TTTAATTTT
X[i46]	CAATATCTCACTCAT TTTAATTTT
X[i47]	CACCAATCCATTTCA TTTAATTTT
X[i48]	CATTTTAACATTCCC TTTAATTTT
X[i49]	CACCTCCTCAACACA TTTAATTTT

Name	Sequence
X[i50]	CATTCTTACACCAAC TTTAATTTT
X[i51]	CACATTCCCATTAAC TTTAATTTT
X[i52]	CAACCATCCAAACTA TTTAATTTT
X[i53]	CAAACAACCATTTAC TTTAATTTT
X[i54]	CAATCCTCCAATACC TTTAATTTT
X[i55]	CATCACACCACCTAT TTTAATTTT
X[i56]	CAACCAAACATCACT TTTAATTTT
X[i57]	CATAACCTCAAATCT TTTAATTTT
X[i58]	CAATATTCCACATCA TTTAATTTT
X[i59]	CACTATACCAATAAC TTTAATTTT
X[i60]	CAAAATCCCAATCTA TTTAATTTT
X[i61]	CACTAAACCATACAT TTTAATTTT
X[i62]	CAACAACCCAAATCC TTTAATTTT
X[i63]	CAATAAAACACCCTA TTTAATTTT
X[i64]	CAAATCACCAAAACA TTTAATTTT
X[i65]	CATCTTATCAAACAC TTTAATTTT
X[i66]	CATACCACCATCCTC TTTAATTTT
X[i67]	CACAAACTCACCTCA TTTAATTTT
X[i68]	CAACCCACCAATTTT TTTAATTTT
X[i69]	CATTCACCCATATTA TTTAATTTT
X[i70]	CACCCAATCATAAAC TTTAATTTT
X[i71]	CACACCAACAACCAT TTTAATTTT
X[i72]	CACTCTAACACTTAT TTTAATTTT
X[i73]	CAACAATTCACCTAC TTTAATTTT
X[i74]	CACCCTACCACTCCC TTTAATTTT
X[i75]	CACACTTCCAATTAT TTTAATTTT
X[i76]	CACAATATCATCACC TTTAATTTT
X[i77]	CAAACTCTCATCATT TTTAATTTT
X[i78]	CAATTTCACACCACC TTTAATTTT
X[i79]	CAACTTACCATTCAT TTTAATTTT
X[i80]	CATCCATACAACTAT TTTAATTTT
X[i81]	CAACTCTACACATTC TTTAATTTT
X[i82]	CAAATCCTCATTTTC TTTAATTTT
X[i83]	CACCATTACAAACAT TTTAATTTT
X[i84]	CATATATTCAACTCC TTTAATTTT
X[i85]	CAATCACTCATTACA TTTAATTTT
X[i86]	CACCAACACAAAATA TTTAATTTT
X[i87]	CAAACCCACATATAT TTTAATTTT
X[i88]	CATCTCTCCATAAAT TTTAATTTT
X[i89]	CATATCAACATCTCA TTTAATTTT
X[i90]	CACATCCACACTACA TTTAATTTT
X[i91]	CATCAAACCACAACA TTTAATTTT
X[i92]	CAAACTTTCACACAC TTTAATTTT
X[i93]	CACCTACCCATCTAT TTTAATTTT
X[i94]	CACCTTTACAATTCC TTTAATTTT
X[i95]	CATTCCTTCACATAT TTTAATTTT
X[i96]	CACAACTCCACCATT TTTAATTTT
X[i97]	CAATACTCCATCAAT TTTAATTTT
X[i98]	CACTTATCCAACATT TTTAATTTT
X[i99]	CACATAATCACCCTT TTTAATTTT
X[i100]	CATACCCTCACTAAC TTTAATTTT

Table S2: DNA sequences of top strands in the learning gates for memory 1.

Name	Sequence
inhAct[m1-i1]-t	TGAAAGA CAACTTCCC TCTTTCA ACACTT TTTAATTTT CC AACCTTC
inhAct[m1-i2]-t	TGAAAGA CATCTAACC TCTTTCA AACTTA TTTAATTTT CC AACCTTC
inhAct[m1-i3]-t	TGAAAGA CACCTAAAC TCTTTCA AATACT TTTAATTTT CC AACCTTC
inhAct[m1-i4]-t	TGAAAGA CATTACATC TCTTTCA ACAATC TTTAATTTT CC AACCTTC
inhAct[m1-i5]-t	TGAAAGA CAACCTTAC TCTTTCA ATTATC TTTAATTTT CC AACCTTC
inhAct[m1-i6]-t	TGAAAGA CAATCCATC TCTTTCA ATCTTA TTTAATTTT CC AACCTTC
inhAct[m1-i7]-t	TGAAAGA CATCACTAC TCTTTCA ATCCAC TTTAATTTT CC AACCTTC
inhAct[m1-i8]-t	TGAAAGA CATCCCAAC TCTTTCA ATACCT TTTAATTTT CC AACCTTC
inhAct[m1-i9]-t	TGAAAGA CAATCTCCC TCTTTCA AACCCA TTTAATTTT CC AACCTTC
inhAct[m1-i10]-t	TGAAAGA CATTTCCAC TCTTTCA AACTTT TTTAATTTT CC AACCTTC
inhAct[m1-i11]-t	TGAAAGA CATCTTTTC TCTTTCA ACCACT TTTAATTTT CC AACCTTC
inhAct[m1-i12]-t	TGAAAGA CAATTACTC TCTTTCA AAACTC TTTAATTTT CC AACCTTC
inhAct[m1-i13]-t	TGAAAGA CACTATCAC TCTTTCA AACCTC TTTAATTTT CC AACCTTC
inhAct[m1-i14]-t	TGAAAGA CACAACAAC TCTTTCA ACACCC TTTAATTTT CC AACCTTC
inhAct[m1-i15]-t	TGAAAGA CAATCATAC TCTTTCA ATATCC TTTAATTTT CC AACCTTC
inhAct[m1-i16]-t	TGAAAGA CACCCTTTC TCTTTCA ATACTA TTTAATTTT CC AACCTTC
inhAct[m1-i17]-t	TGAAAGA CATCCACTC TCTTTCA AATCCC TTTAATTTT CC AACCTTC
inhAct[m1-i18]-t	TGAAAGA CATACTCAC TCTTTCA ATAATC TTTAATTTT CC AACCTTC
inhAct[m1-i19]-t	TGAAAGA CATACACCC TCTTTCA ACTTCT TTTAATTTT CC AACCTTC
inhAct[m1-i20]-t	TGAAAGA CACCTTCTC TCTTTCA ATATCT TTTAATTTT CC AACCTTC
inhAct[m1-i21]-t	TGAAAGA CATTATTTC TCTTTCA AACCCT TTTAATTTT CC AACCTTC
inhAct[m1-i22]-t	TGAAAGA CATTAATAC TCTTTCA ACTTCC TTTAATTTT CC AACCTTC
inhAct[m1-i23]-t	TGAAAGA CATTTACAC TCTTTCA ACACAT TTTAATTTT CC AACCTTC
inhAct[m1-i24]-t	TGAAAGA CATAATTCC TCTTTCA ATCTTC TTTAATTTT CC AACCTTC
inhAct[m1-i25]-t	TGAAAGA CAACACTTC TCTTTCA ACTCCT TTTAATTTT CC AACCTTC
inhAct[m1-i26]-t	TGAAAGA CATCTATCC TCTTTCA ACTATC TTTAATTTT CC AACCTTC
inhAct[m1-i27]-t	TGAAAGA CACTCCTAC TCTTTCA AAATTA TTTAATTTT CC AACCTTC
inhAct[m1-i28]-t	TGAAAGA CACTACCTC TCTTTCA ATACCC TTTAATTTT CC AACCTTC
inhAct[m1-i29]-t	TGAAAGA CATACCTAC TCTTTCA ACTCTA TTTAATTTT CC AACCTTC
inhAct[m1-i30]-t	TGAAAGA CACTCACAC TCTTTCA ACCTTC TTTAATTTT CC AACCTTC
inhAct[m1-i31]-t	TGAAAGA CAACTACAC TCTTTCA ATTCTA TTTAATTTT CC AACCTTC
inhAct[m1-i32]-t	TGAAAGA CATATTCAC TCTTTCA AAACCA TTTAATTTT CC AACCTTC
inhAct[m1-i33]-t	TGAAAGA CACACATTC TCTTTCA AAAACT TTTAATTTT CC AACCTTC
inhAct[m1-i34]-t	TGAAAGA CACTTTTCC TCTTTCA ACTTTA TTTAATTTT CC AACCTTC
inhAct[m1-i35]-t	TGAAAGA CAAATACCC TCTTTCA ACCCAC TTTAATTTT CC AACCTTC
inhAct[m1-i36]-t	TGAAAGA CACCACATC TCTTTCA ATTATT TTTAATTTT CC AACCTTC
inhAct[m1-i37]-t	TGAAAGA CATCCCTTC TCTTTCA AATATA TTTAATTTT CC AACCTTC
inhAct[m1-i38]-t	TGAAAGA CAATTCATC TCTTTCA AACAAC TTTAATTTT CC AACCTTC
inhAct[m1-i39]-t	TGAAAGA CATAAACAC TCTTTCA ATCCCT TTTAATTTT CC AACCTTC
inhAct[m1-i40]-t	TGAAAGA CACATATAC TCTTTCA ACAAAC TTTAATTTT CC AACCTTC
inhAct[m1-i41]-t	TGAAAGA CATCCTATC TCTTTCA ACTTTC TTTAATTTT CC AACCTTC
inhAct[m1-i42]-t	TGAAAGA CAAAACCAC TCTTTCA AATCAC TTTAATTTT CC AACCTTC
inhAct[m1-i43]-t	TGAAAGA CATCATAAC TCTTTCA AACACC TTTAATTTT CC AACCTTC
inhAct[m1-i44]-t	TGAAAGA CACAAATTC TCTTTCA ATTCAC TTTAATTTT CC AACCTTC
inhAct[m1-i45]-t	TGAAAGA CATTATCCC TCTTTCA ATAACT TTTAATTTT CC AACCTTC
inhAct[m1-i46]-t	TGAAAGA CAATATCTC TCTTTCA ACTCAT TTTAATTTT CC AACCTTC
inhAct[m1-i47]-t	TGAAAGA CACCAATCC TCTTTCA ATTTCA TTTAATTTT CC AACCTTC
inhAct[m1-i48]-t	TGAAAGA CATTTTAAC TCTTTCA ATTCCC TTTAATTTT CC AACCTTC
inhAct[m1-i49]-t	TGAAAGA CACCTCCTC TCTTTCA AACACA TTTAATTTT CC AACCTTC
inhAct[m1-i50]-t	TGAAAGA CATTCTTAC TCTTTCA ACCAAC TTTAATTTT CC AACCTTC
inhAct[m1-i51]-t	TGAAAGA CACATTCCC TCTTTCA ATTAAC TTTAATTTT CC AACCTTC

Name	Sequence
inhAct[m1-i52]-t	TGAAAGA CAACCATCC TCTTTCA AAACTA TTTAATTTT CC AACCTTC
inhAct[m1-i53]-t	TGAAAGA CAAACAACC TCTTTCA ATTTAC TTTAATTTT CC AACCTTC
inhAct[m1-i54]-t	TGAAAGA CAATCCTCC TCTTTCA AATACC TTTAATTTT CC AACCTTC
inhAct[m1-i55]-t	TGAAAGA CATCACACC TCTTTCA ACCTAT TTTAATTTT CC AACCTTC
inhAct[m1-i56]-t	TGAAAGA CAACCAAAC TCTTTCA ATCACT TTTAATTTT CC AACCTTC
inhAct[m1-i57]-t	TGAAAGA CATAACCTC TCTTTCA AAATCT TTTAATTTT CC AACCTTC
inhAct[m1-i58]-t	TGAAAGA CAATATTCC TCTTTCA ACATCA TTTAATTTT CC AACCTTC
inhAct[m1-i59]-t	TGAAAGA CACTATACC TCTTTCA AATAAC TTTAATTTT CC AACCTTC
inhAct[m1-i60]-t	TGAAAGA CAAAATCCC TCTTTCA AATCTA TTTAATTTT CC AACCTTC
inhAct[m1-i61]-t	TGAAAGA CACTAAACC TCTTTCA ATACAT TTTAATTTT CC AACCTTC
inhAct[m1-i62]-t	TGAAAGA CAACAACCC TCTTTCA AAATCC TTTAATTTT CC AACCTTC
inhAct[m1-i63]-t	TGAAAGA CAATAAAAC TCTTTCA ACCCTA TTTAATTTT CC AACCTTC
inhAct[m1-i64]-t	TGAAAGA CAAATCACC TCTTTCA AAAACA TTTAATTTT CC AACCTTC
inhAct[m1-i65]-t	TGAAAGA CATCTTATC TCTTTCA AAACAC TTTAATTTT CC AACCTTC
inhAct[m1-i66]-t	TGAAAGA CATACCACC TCTTTCA ATCCTC TTTAATTTT CC AACCTTC
inhAct[m1-i67]-t	TGAAAGA CACAAACTC TCTTTCA ACCTCA TTTAATTTT CC AACCTTC
inhAct[m1-i68]-t	TGAAAGA CAACCCACC TCTTTCA AATTTT TTTAATTTT CC AACCTTC
inhAct[m1-i69]-t	TGAAAGA CATTCACCC TCTTTCA ATATTA TTTAATTTT CC AACCTTC
inhAct[m1-i70]-t	TGAAAGA CACCCAATC TCTTTCA ATAAAC TTTAATTTT CC AACCTTC
inhAct[m1-i71]-t	TGAAAGA CACACCAAC TCTTTCA AACCAT TTTAATTTT CC AACCTTC
inhAct[m1-i72]-t	TGAAAGA CACTCTAAC TCTTTCA ACTTAT TTTAATTTT CC AACCTTC
inhAct[m1-i73]-t	TGAAAGA CAACAATTC TCTTTCA ACCTAC TTTAATTTT CC AACCTTC
inhAct[m1-i74]-t	TGAAAGA CACCCTACC TCTTTCA ACTCCC TTTAATTTT CC AACCTTC
inhAct[m1-i75]-t	TGAAAGA CACACTTCC TCTTTCA AATTAT TTTAATTTT CC AACCTTC
inhAct[m1-i76]-t	TGAAAGA CACAATATC TCTTTCA ATCACC TTTAATTTT CC AACCTTC
inhAct[m1-i77]-t	TGAAAGA CAAACTCTC TCTTTCA ATCATT TTTAATTTT CC AACCTTC
inhAct[m1-i78]-t	TGAAAGA CAATTTCAC TCTTTCA ACCACC TTTAATTTT CC AACCTTC
inhAct[m1-i79]-t	TGAAAGA CAACTTACC TCTTTCA ATTCAT TTTAATTTT CC AACCTTC
inhAct[m1-i80]-t	TGAAAGA CATCCATAC TCTTTCA AACTAT TTTAATTTT CC AACCTTC
inhAct[m1-i81]-t	TGAAAGA CAACTCTAC TCTTTCA ACATTC TTTAATTTT CC AACCTTC
inhAct[m1-i82]-t	TGAAAGA CAAATCCTC TCTTTCA ATTTTC TTTAATTTT CC AACCTTC
inhAct[m1-i83]-t	TGAAAGA CACCATTAC TCTTTCA AAACAT TTTAATTTT CC AACCTTC
inhAct[m1-i84]-t	TGAAAGA CATATATTC TCTTTCA AACTCC TTTAATTTT CC AACCTTC
inhAct[m1-i85]-t	TGAAAGA CAATCACTC TCTTTCA ATTACA TTTAATTTT CC AACCTTC
inhAct[m1-i86]-t	TGAAAGA CACCAACAC TCTTTCA AAAATA TTTAATTTT CC AACCTTC
inhAct[m1-i87]-t	TGAAAGA CAAACCCAC TCTTTCA ATATAT TTTAATTTT CC AACCTTC
inhAct[m1-i88]-t	TGAAAGA CATCTCTCC TCTTTCA ATAAAT TTTAATTTT CC AACCTTC
inhAct[m1-i89]-t	TGAAAGA CATATCAAC TCTTTCA ATCTCA TTTAATTTT CC AACCTTC
inhAct[m1-i90]-t	TGAAAGA CACATCCAC TCTTTCA ACTACA TTTAATTTT CC AACCTTC
inhAct[m1-i91]-t	TGAAAGA CATCAAACC TCTTTCA ACAACA TTTAATTTT CC AACCTTC
inhAct[m1-i92]-t	TGAAAGA CAAACTTTC TCTTTCA ACACAC TTTAATTTT CC AACCTTC
inhAct[m1-i93]-t	TGAAAGA CACCTACCC TCTTTCA ATCTAT TTTAATTTT CC AACCTTC
inhAct[m1-i94]-t	TGAAAGA CACCTTTAC TCTTTCA AATTCC TTTAATTTT CC AACCTTC
inhAct[m1-i95]-t	TGAAAGA CATTCCTTC TCTTTCA ACATAT TTTAATTTT CC AACCTTC
inhAct[m1-i96]-t	TGAAAGA CACAACTCC TCTTTCA ACCATT TTTAATTTT CC AACCTTC
inhAct[m1-i97]-t	TGAAAGA CAATACTCC TCTTTCA ATCAAT TTTAATTTT CC AACCTTC
inhAct[m1-i98]-t	TGAAAGA CACTTATCC TCTTTCA AACATT TTTAATTTT CC AACCTTC
inhAct[m1-i99]-t	TGAAAGA CACATAATC TCTTTCA ACCCTT TTTAATTTT CC AACCTTC
inhAct[m1-i100]-t	TGAAAGA CATACCCTC TCTTTCA ACTAAC TTTAATTTT CC AACCTTC

Table S3: DNA sequences of bottom strands in the learning gates for memory 1.

Name	Sequence
inhAct[m1-i1]-b	GG AAAATTAAA AAGTGTGGGAAGTTG TCTTTCA TTATCCTCC AAAATTAAA TT
inhAct[m1-i2]-b	GG AAAATTAAA TAAGTTGGTTAGATG TCTTTCA TCCAAATTC AAAATTAAA TT
inhAct[m1-i3]-b	GG AAAATTAAA AGTATTGTTTAGGTG TCTTTCA ACACCATCC AAAATTAAA TT
inhAct[m1-i4]-b	GG AAAATTAAA GATTGTGATGTAATG TCTTTCA CTCCACTCC AAAATTAAA TT
inhAct[m1-i5]-b	GG AAAATTAAA GATAATGTAAGGTTG TCTTTCA CCTTATTAC AAAATTAAA TT
inhAct[m1-i6]-b	GG AAAATTAAA TAAGATGATGGATTG TCTTTCA CCTCAAAAC AAAATTAAA TT
inhAct[m1-i7]-b	GG AAAATTAAA GTGGATGTAGTGATG TCTTTCA ATCTCTCTC AAAATTAAA TT
inhAct[m1-i8]-b	GG AAAATTAAA AGGTATGTTGGGATG TCTTTCA CATTTCTTC AAAATTAAA TT
inhAct[m1-i9]-b	GG AAAATTAAA TGGGTTGGGAGATTG TCTTTCA AACCCTACC AAAATTAAA TT
inhAct[m1-i10]-b	GG AAAATTAAA AAAGTTGTGGAAATG TCTTTCA TCATATCTC AAAATTAAA TT
inhAct[m1-i11]-b	GG AAAATTAAA AGTGGTGAAAAGATG TCTTTCA CAATCATAC AAAATTAAA TT
inhAct[m1-i12]-b	GG AAAATTAAA GAGTTTGAGTAATTG TCTTTCA CCATTTCCC AAAATTAAA TT
inhAct[m1-i13]-b	GG AAAATTAAA GAGGTTGTGATAGTG TCTTTCA AACATAACC AAAATTAAA TT
inhAct[m1-i14]-b	GG AAAATTAAA GGGTGTGTTGTTGTG TCTTTCA CTTTAACAC AAAATTAAA TT
inhAct[m1-i15]-b	GG AAAATTAAA GGATATGTATGATTG TCTTTCA CACCTATAC AAAATTAAA TT
inhAct[m1-i16]-b	GG AAAATTAAA TAGTATGAAAGGGTG TCTTTCA CCCATCAAC AAAATTAAA TT
inhAct[m1-i17]-b	GG AAAATTAAA GGGATTGAGTGGATG TCTTTCA ATTCTCATC AAAATTAAA TT
inhAct[m1-i18]-b	GG AAAATTAAA GATTATGTGAGTATG TCTTTCA TAATAACCC AAAATTAAA TT
inhAct[m1-i19]-b	GG AAAATTAAA AGAAGTGGGTGTATG TCTTTCA TACTTATCC AAAATTAAA TT
inhAct[m1-i20]-b	GG AAAATTAAA AGATATGAGAAGGTG TCTTTCA ATTCAACTC AAAATTAAA TT
inhAct[m1-i21]-b	GG AAAATTAAA AGGGTTGAAATAATG TCTTTCA CTCACTTCC AAAATTAAA TT
inhAct[m1-i22]-b	GG AAAATTAAA GGAAGTGTATTAATG TCTTTCA TATCACATC AAAATTAAA TT
inhAct[m1-i23]-b	GG AAAATTAAA ATGTGTGTGTAAATG TCTTTCA TCTCATTCC AAAATTAAA TT
inhAct[m1-i24]-b	GG AAAATTAAA GAAGATGGAATTATG TCTTTCA ATCACAAAC AAAATTAAA TT
inhAct[m1-i25]-b	GG AAAATTAAA AGGAGTGAAGTGTTG TCTTTCA ATTTCCAAC AAAATTAAA TT
inhAct[m1-i26]-b	GG AAAATTAAA GATAGTGGATAGATG TCTTTCA CATAAAACC AAAATTAAA TT
inhAct[m1-i27]-b	GG AAAATTAAA TAATTTGTAGGAGTG TCTTTCA TTCCCTTTC AAAATTAAA TT
inhAct[m1-i28]-b	GG AAAATTAAA GGGTATGAGGTAGTG TCTTTCA ACTATACAC AAAATTAAA TT
inhAct[m1-i29]-b	GG AAAATTAAA TAGAGTGTAGGTATG TCTTTCA TCCTTTTAC AAAATTAAA TT
inhAct[m1-i30]-b	GG AAAATTAAA GAAGGTGTGTGAGTG TCTTTCA TATACCACC AAAATTAAA TT
inhAct[m1-i31]-b	GG AAAATTAAA TAGAATGTGTAGTTG TCTTTCA TTACCTAAC AAAATTAAA TT
inhAct[m1-i32]-b	GG AAAATTAAA TGGTTTGTGAATATG TCTTTCA TCAATATCC AAAATTAAA TT
inhAct[m1-i33]-b	GG AAAATTAAA AGTTTTGAATGTGTG TCTTTCA CTCCTCCAC AAAATTAAA TT
inhAct[m1-i34]-b	GG AAAATTAAA TAAAGTGGAAAAGTG TCTTTCA ACCCACAAC AAAATTAAA TT
inhAct[m1-i35]-b	GG AAAATTAAA GTGGGTGGGTATTTG TCTTTCA AACAACTTC AAAATTAAA TT
inhAct[m1-i36]-b	GG AAAATTAAA AATAATGATGTGGTG TCTTTCA AAATACCTC AAAATTAAA TT
inhAct[m1-i37]-b	GG AAAATTAAA TATATTGAAGGGATG TCTTTCA TCCCAATAC AAAATTAAA TT
inhAct[m1-i38]-b	GG AAAATTAAA GTTGTTGATGAATTG TCTTTCA CATCCTCTC AAAATTAAA TT
inhAct[m1-i39]-b	GG AAAATTAAA AGGGATGTGTTTATG TCTTTCA TCTACTTAC AAAATTAAA TT
inhAct[m1-i40]-b	GG AAAATTAAA GTTTGTGTATATGTG TCTTTCA AAAACCTAC AAAATTAAA TT
inhAct[m1-i41]-b	GG AAAATTAAA GAAAGTGATAGGATG TCTTTCA CAACCACCC AAAATTAAA TT
inhAct[m1-i42]-b	GG AAAATTAAA GTGATTGTGGTTTTG TCTTTCA TATTCCCTC AAAATTAAA TT
inhAct[m1-i43]-b	GG AAAATTAAA GGTGTTGTTATGATG TCTTTCA ACCTTAAAC AAAATTAAA TT
inhAct[m1-i44]-b	GG AAAATTAAA GTGAATGAATTTGTG TCTTTCA CCTATTTTC AAAATTAAA TT
inhAct[m1-i45]-b	GG AAAATTAAA AGTTATGGGATAATG TCTTTCA AACCAAATC AAAATTAAA TT
inhAct[m1-i46]-b	GG AAAATTAAA ATGAGTGAGATATTG TCTTTCA AATCTACCC AAAATTAAA TT
inhAct[m1-i47]-b	GG AAAATTAAA TGAAATGGATTGGTG TCTTTCA TACCCAAAC AAAATTAAA TT
inhAct[m1-i48]-b	GG AAAATTAAA GGGAATGTTAAAATG TCTTTCA CCTTCAATC AAAATTAAA TT
inhAct[m1-i49]-b	GG AAAATTAAA TGTGTTGAGGAGGTG TCTTTCA TTAATTCCC AAAATTAAA TT
inhAct[m1-i50]-b	GG AAAATTAAA GTTGGTGTAAGAATG TCTTTCA ACAAAACTC AAAATTAAA TT
inhAct[m1-i51]-b	GG AAAATTAAA GTTAATGGGAATGTG TCTTTCA ATTAATCCC AAAATTAAA TT

Name	Sequence
inhAct[m1-i52]-b	GG AAAATTAAA TAGTTTGGATGGTTG TCTTTCA CCAACTCAC AAAATTAAA TT
inhAct[m1-i53]-b	GG AAAATTAAA GTAAATGGTTGTTTG TCTTTCA TAAACACTC AAAATTAAA TT
inhAct[m1-i54]-b	GG AAAATTAAA GGTATTGGAGGATTG TCTTTCA CCCTCCTTC AAAATTAAA TT
inhAct[m1-i55]-b	GG AAAATTAAA ATAGGTGGTGTGATG TCTTTCA CTATATACC AAAATTAAA TT
inhAct[m1-i56]-b	GG AAAATTAAA AGTGATGTTTGGTTG TCTTTCA CTACAATTC AAAATTAAA TT
inhAct[m1-i57]-b	GG AAAATTAAA AGATTTGAGGTTATG TCTTTCA CATATTCAC AAAATTAAA TT
inhAct[m1-i58]-b	GG AAAATTAAA TGATGTGGAATATTG TCTTTCA AAACTTTCC AAAATTAAA TT
inhAct[m1-i59]-b	GG AAAATTAAA GTTATTGGTATAGTG TCTTTCA CTCTTCACC AAAATTAAA TT
inhAct[m1-i60]-b	GG AAAATTAAA TAGATTGGGATTTTG TCTTTCA ATATTACCC AAAATTAAA TT
inhAct[m1-i61]-b	GG AAAATTAAA ATGTATGGTTTAGTG TCTTTCA CATTACACC AAAATTAAA TT
inhAct[m1-i62]-b	GG AAAATTAAA GGATTTGGGTTGTTG TCTTTCA ACTTTTACC AAAATTAAA TT
inhAct[m1-i63]-b	GG AAAATTAAA TAGGGTGTTTTATTG TCTTTCA ACCATATTC AAAATTAAA TT
inhAct[m1-i64]-b	GG AAAATTAAA TGTTTTGGTGATTTG TCTTTCA CCTCCCAAC AAAATTAAA TT
inhAct[m1-i65]-b	GG AAAATTAAA GTGTTTGATAAGATG TCTTTCA TCCACTACC AAAATTAAA TT
inhAct[m1-i66]-b	GG AAAATTAAA GAGGATGGTGGTATG TCTTTCA TCCCTTATC AAAATTAAA TT
inhAct[m1-i67]-b	GG AAAATTAAA TGAGGTGAGTTTGTG TCTTTCA TTTCCATTC AAAATTAAA TT
inhAct[m1-i68]-b	GG AAAATTAAA AAAATTGGTGGGTTG TCTTTCA ACACACTTC AAAATTAAA TT
inhAct[m1-i69]-b	GG AAAATTAAA TAATATGGGTGAATG TCTTTCA CCACTAATC AAAATTAAA TT
inhAct[m1-i70]-b	GG AAAATTAAA GTTTATGATTGGGTG TCTTTCA AATTCACAC AAAATTAAA TT
inhAct[m1-i71]-b	GG AAAATTAAA ATGGTTGTTGGTGTG TCTTTCA TTATTCCTC AAAATTAAA TT
inhAct[m1-i72]-b	GG AAAATTAAA ATAAGTGTTAGAGTG TCTTTCA ACTCTCTAC AAAATTAAA TT
inhAct[m1-i73]-b	GG AAAATTAAA GTAGGTGAATTGTTG TCTTTCA TTCAAACAC AAAATTAAA TT
inhAct[m1-i74]-b	GG AAAATTAAA GGGAGTGGTAGGGTG TCTTTCA CTCATTATC AAAATTAAA TT
inhAct[m1-i75]-b	GG AAAATTAAA ATAATTGGAAGTGTG TCTTTCA TAACACCAC AAAATTAAA TT
inhAct[m1-i76]-b	GG AAAATTAAA GGTGATGATATTGTG TCTTTCA ACATCTATC AAAATTAAA TT
inhAct[m1-i77]-b	GG AAAATTAAA AATGATGAGAGTTTG TCTTTCA CAACATATC AAAATTAAA TT
inhAct[m1-i78]-b	GG AAAATTAAA GGTGGTGTGAAATTG TCTTTCA AACTCTTAC AAAATTAAA TT
inhAct[m1-i79]-b	GG AAAATTAAA ATGAATGGTAAGTTG TCTTTCA ATCCATCAC AAAATTAAA TT
inhAct[m1-i80]-b	GG AAAATTAAA ATAGTTGTATGGATG TCTTTCA CTTCTTAAC AAAATTAAA TT
inhAct[m1-i81]-b	GG AAAATTAAA GAATGTGTAGAGTTG TCTTTCA ATAACCATC AAAATTAAA TT
inhAct[m1-i82]-b	GG AAAATTAAA GAAAATGAGGATTTG TCTTTCA CACTATTTC AAAATTAAA TT
inhAct[m1-i83]-b	GG AAAATTAAA ATGTTTGTAATGGTG TCTTTCA ATACTTCTC AAAATTAAA TT
inhAct[m1-i84]-b	GG AAAATTAAA GGAGTTGAATATATG TCTTTCA TCACCCTAC AAAATTAAA TT
inhAct[m1-i85]-b	GG AAAATTAAA TGTAATGAGTGATTG TCTTTCA TACACCCAC AAAATTAAA TT
inhAct[m1-i86]-b	GG AAAATTAAA TATTTTGTGTTGGTG TCTTTCA CCAATCTCC AAAATTAAA TT
inhAct[m1-i87]-b	GG AAAATTAAA ATATATGTGGGTTTG TCTTTCA CACAATAAC AAAATTAAA TT
inhAct[m1-i88]-b	GG AAAATTAAA ATTTATGGAGAGATG TCTTTCA CAAAATTCC AAAATTAAA TT
inhAct[m1-i89]-b	GG AAAATTAAA TGAGATGTTGATATG TCTTTCA AATACTCCC AAAATTAAA TT
inhAct[m1-i90]-b	GG AAAATTAAA TGTAGTGTGGATGTG TCTTTCA TCTAACCTC AAAATTAAA TT
inhAct[m1-i91]-b	GG AAAATTAAA TGTTGTGGTTTGATG TCTTTCA TCACTCACC AAAATTAAA TT
inhAct[m1-i92]-b	GG AAAATTAAA GTGTGTGAAAGTTTG TCTTTCA TTCTCAACC AAAATTAAA TT
inhAct[m1-i93]-b	GG AAAATTAAA ATAGATGGGTAGGTG TCTTTCA CTTACATAC AAAATTAAA TT
inhAct[m1-i94]-b	GG AAAATTAAA GGAATTGTAAAGGTG TCTTTCA ATCTACTAC AAAATTAAA TT
inhAct[m1-i95]-b	GG AAAATTAAA ATATGTGAAGGAATG TCTTTCA CTAAACATC AAAATTAAA TT
inhAct[m1-i96]-b	GG AAAATTAAA AATGGTGGAGTTGTG TCTTTCA ACTTAATCC AAAATTAAA TT
inhAct[m1-i97]-b	GG AAAATTAAA ATTGATGGAGTATTG TCTTTCA TTTTCTCAC AAAATTAAA TT
inhAct[m1-i98]-b	GG AAAATTAAA AATGTTGGATAAGTG TCTTTCA TCTTTACTC AAAATTAAA TT
inhAct[m1-i99]-b	GG AAAATTAAA AAGGGTGATTATGTG TCTTTCA TTTATCCAC AAAATTAAA TT
inhAct[m1-i100]-b	GG AAAATTAAA GTTAGTGAGGGTATG TCTTTCA TCCTAAATC AAAATTAAA TT

Table S4: DNA sequences of top strands in the learning gates for memory 2.

Name	Sequence
inhAct[m2-i1]-t	AATAGAG CAACTTCCC CTCTATT ACACTT TTTAATTTT AC TCACTAC
inhAct[m2-i2]-t	AATAGAG CATCTAACC CTCTATT AACTTA TTTAATTTT AC TCACTAC
inhAct[m2-i3]-t	AATAGAG CACCTAAAC CTCTATT AATACT TTTAATTTT AC TCACTAC
inhAct[m2-i4]-t	AATAGAG CATTACATC CTCTATT ACAATC TTTAATTTT AC TCACTAC
inhAct[m2-i5]-t	AATAGAG CAACCTTAC CTCTATT ATTATC TTTAATTTT AC TCACTAC
inhAct[m2-i6]-t	AATAGAG CAATCCATC CTCTATT ATCTTA TTTAATTTT AC TCACTAC
inhAct[m2-i7]-t	AATAGAG CATCACTAC CTCTATT ATCCAC TTTAATTTT AC TCACTAC
inhAct[m2-i8]-t	AATAGAG CATCCCAAC CTCTATT ATACCT TTTAATTTT AC TCACTAC
inhAct[m2-i9]-t	AATAGAG CAATCTCCC CTCTATT AACCCA TTTAATTTT AC TCACTAC
inhAct[m2-i10]-t	AATAGAG CATTTCCAC CTCTATT AACTTT TTTAATTTT AC TCACTAC
inhAct[m2-i11]-t	AATAGAG CATCTTTTC CTCTATT ACCACT TTTAATTTT AC TCACTAC
inhAct[m2-i12]-t	AATAGAG CAATTACTC CTCTATT AAACTC TTTAATTTT AC TCACTAC
inhAct[m2-i13]-t	AATAGAG CACTATCAC CTCTATT AACCTC TTTAATTTT AC TCACTAC
inhAct[m2-i14]-t	AATAGAG CACAACAAC CTCTATT ACACCC TTTAATTTT AC TCACTAC
inhAct[m2-i15]-t	AATAGAG CAATCATAC CTCTATT ATATCC TTTAATTTT AC TCACTAC
inhAct[m2-i16]-t	AATAGAG CACCCTTTC CTCTATT ATACTA TTTAATTTT AC TCACTAC
inhAct[m2-i17]-t	AATAGAG CATCCACTC CTCTATT AATCCC TTTAATTTT AC TCACTAC
inhAct[m2-i18]-t	AATAGAG CATACTCAC CTCTATT ATAATC TTTAATTTT AC TCACTAC
inhAct[m2-i19]-t	AATAGAG CATACACCC CTCTATT ACTTCT TTTAATTTT AC TCACTAC
inhAct[m2-i20]-t	AATAGAG CACCTTCTC CTCTATT ATATCT TTTAATTTT AC TCACTAC
inhAct[m2-i21]-t	AATAGAG CATTATTTC CTCTATT AACCCT TTTAATTTT AC TCACTAC
inhAct[m2-i22]-t	AATAGAG CATTAATAC CTCTATT ACTTCC TTTAATTTT AC TCACTAC
inhAct[m2-i23]-t	AATAGAG CATTTACAC CTCTATT ACACAT TTTAATTTT AC TCACTAC
inhAct[m2-i24]-t	AATAGAG CATAATTCC CTCTATT ATCTTC TTTAATTTT AC TCACTAC
inhAct[m2-i25]-t	AATAGAG CAACACTTC CTCTATT ACTCCT TTTAATTTT AC TCACTAC
inhAct[m2-i26]-t	AATAGAG CATCTATCC CTCTATT ACTATC TTTAATTTT AC TCACTAC
inhAct[m2-i27]-t	AATAGAG CACTCCTAC CTCTATT AAATTA TTTAATTTT AC TCACTAC
inhAct[m2-i28]-t	AATAGAG CACTACCTC CTCTATT ATACCC TTTAATTTT AC TCACTAC
inhAct[m2-i29]-t	AATAGAG CATACCTAC CTCTATT ACTCTA TTTAATTTT AC TCACTAC
inhAct[m2-i30]-t	AATAGAG CACTCACAC CTCTATT ACCTTC TTTAATTTT AC TCACTAC
inhAct[m2-i31]-t	AATAGAG CAACTACAC CTCTATT ATTCTA TTTAATTTT AC TCACTAC
inhAct[m2-i32]-t	AATAGAG CATATTCAC CTCTATT AAACCA TTTAATTTT AC TCACTAC
inhAct[m2-i33]-t	AATAGAG CACACATTC CTCTATT AAAACT TTTAATTTT AC TCACTAC
inhAct[m2-i34]-t	AATAGAG CACTTTTCC CTCTATT ACTTTA TTTAATTTT AC TCACTAC
inhAct[m2-i35]-t	AATAGAG CAAATACCC CTCTATT ACCCAC TTTAATTTT AC TCACTAC
inhAct[m2-i36]-t	AATAGAG CACCACATC CTCTATT ATTATT TTTAATTTT AC TCACTAC
inhAct[m2-i37]-t	AATAGAG CATCCCTTC CTCTATT AATATA TTTAATTTT AC TCACTAC
inhAct[m2-i38]-t	AATAGAG CAATTCATC CTCTATT AACAAC TTTAATTTT AC TCACTAC
inhAct[m2-i39]-t	AATAGAG CATAAACAC CTCTATT ATCCCT TTTAATTTT AC TCACTAC
inhAct[m2-i40]-t	AATAGAG CACATATAC CTCTATT ACAAAC TTTAATTTT AC TCACTAC
inhAct[m2-i41]-t	AATAGAG CATCCTATC CTCTATT ACTTTC TTTAATTTT AC TCACTAC
inhAct[m2-i42]-t	AATAGAG CAAAACCAC CTCTATT AATCAC TTTAATTTT AC TCACTAC
inhAct[m2-i43]-t	AATAGAG CATCATAAC CTCTATT AACACC TTTAATTTT AC TCACTAC
inhAct[m2-i44]-t	AATAGAG CACAAATTC CTCTATT ATTCAC TTTAATTTT AC TCACTAC
inhAct[m2-i45]-t	AATAGAG CATTATCCC CTCTATT ATAACT TTTAATTTT AC TCACTAC
inhAct[m2-i46]-t	AATAGAG CAATATCTC CTCTATT ACTCAT TTTAATTTT AC TCACTAC
inhAct[m2-i47]-t	AATAGAG CACCAATCC CTCTATT ATTTCA TTTAATTTT AC TCACTAC
inhAct[m2-i48]-t	AATAGAG CATTTTAAC CTCTATT ATTCCC TTTAATTTT AC TCACTAC
inhAct[m2-i49]-t	AATAGAG CACCTCCTC CTCTATT AACACA TTTAATTTT AC TCACTAC
inhAct[m2-i50]-t	AATAGAG CATTCTTAC CTCTATT ACCAAC TTTAATTTT AC TCACTAC
inhAct[m2-i51]-t	AATAGAG CACATTCCC CTCTATT ATTAAC TTTAATTTT AC TCACTAC

Name	Sequence
inhAct[m2-i52]-t	AATAGAG CAACCATCC CTCTATT AAACTA TTTAATTTT AC TCACTAC
inhAct[m2-i53]-t	AATAGAG CAAACAACC CTCTATT ATTTAC TTTAATTTT AC TCACTAC
inhAct[m2-i54]-t	AATAGAG CAATCCTCC CTCTATT AATACC TTTAATTTT AC TCACTAC
inhAct[m2-i55]-t	AATAGAG CATCACACC CTCTATT ACCTAT TTTAATTTT AC TCACTAC
inhAct[m2-i56]-t	AATAGAG CAACCAAAC CTCTATT ATCACT TTTAATTTT AC TCACTAC
inhAct[m2-i57]-t	AATAGAG CATAACCTC CTCTATT AAATCT TTTAATTTT AC TCACTAC
inhAct[m2-i58]-t	AATAGAG CAATATTCC CTCTATT ACATCA TTTAATTTT AC TCACTAC
inhAct[m2-i59]-t	AATAGAG CACTATACC CTCTATT AATAAC TTTAATTTT AC TCACTAC
inhAct[m2-i60]-t	AATAGAG CAAAATCCC CTCTATT AATCTA TTTAATTTT AC TCACTAC
inhAct[m2-i61]-t	AATAGAG CACTAAACC CTCTATT ATACAT TTTAATTTT AC TCACTAC
inhAct[m2-i62]-t	AATAGAG CAACAACCC CTCTATT AAATCC TTTAATTTT AC TCACTAC
inhAct[m2-i63]-t	AATAGAG CAATAAAAC CTCTATT ACCCTA TTTAATTTT AC TCACTAC
inhAct[m2-i64]-t	AATAGAG CAAATCACC CTCTATT AAAACA TTTAATTTT AC TCACTAC
inhAct[m2-i65]-t	AATAGAG CATCTTATC CTCTATT AAACAC TTTAATTTT AC TCACTAC
inhAct[m2-i66]-t	AATAGAG CATACCACC CTCTATT ATCCTC TTTAATTTT AC TCACTAC
inhAct[m2-i67]-t	AATAGAG CACAAACTC CTCTATT ACCTCA TTTAATTTT AC TCACTAC
inhAct[m2-i68]-t	AATAGAG CAACCCACC CTCTATT AATTTT TTTAATTTT AC TCACTAC
inhAct[m2-i69]-t	AATAGAG CATTCACCC CTCTATT ATATTA TTTAATTTT AC TCACTAC
inhAct[m2-i70]-t	AATAGAG CACCCAATC CTCTATT ATAAAC TTTAATTTT AC TCACTAC
inhAct[m2-i71]-t	AATAGAG CACACCAAC CTCTATT AACCAT TTTAATTTT AC TCACTAC
inhAct[m2-i72]-t	AATAGAG CACTCTAAC CTCTATT ACTTAT TTTAATTTT AC TCACTAC
inhAct[m2-i73]-t	AATAGAG CAACAATTC CTCTATT ACCTAC TTTAATTTT AC TCACTAC
inhAct[m2-i74]-t	AATAGAG CACCCTACC CTCTATT ACTCCC TTTAATTTT AC TCACTAC
inhAct[m2-i75]-t	AATAGAG CACACTTCC CTCTATT AATTAT TTTAATTTT AC TCACTAC
inhAct[m2-i76]-t	AATAGAG CACAATATC CTCTATT ATCACC TTTAATTTT AC TCACTAC
inhAct[m2-i77]-t	AATAGAG CAAACTCTC CTCTATT ATCATT TTTAATTTT AC TCACTAC
inhAct[m2-i78]-t	AATAGAG CAATTTCAC CTCTATT ACCACC TTTAATTTT AC TCACTAC
inhAct[m2-i79]-t	AATAGAG CAACTTACC CTCTATT ATTCAT TTTAATTTT AC TCACTAC
inhAct[m2-i80]-t	AATAGAG CATCCATAC CTCTATT AACTAT TTTAATTTT AC TCACTAC
inhAct[m2-i81]-t	AATAGAG CAACTCTAC CTCTATT ACATTC TTTAATTTT AC TCACTAC
inhAct[m2-i82]-t	AATAGAG CAAATCCTC CTCTATT ATTTTC TTTAATTTT AC TCACTAC
inhAct[m2-i83]-t	AATAGAG CACCATTAC CTCTATT AAACAT TTTAATTTT AC TCACTAC
inhAct[m2-i84]-t	AATAGAG CATATATTC CTCTATT AACTCC TTTAATTTT AC TCACTAC
inhAct[m2-i85]-t	AATAGAG CAATCACTC CTCTATT ATTACA TTTAATTTT AC TCACTAC
inhAct[m2-i86]-t	AATAGAG CACCAACAC CTCTATT AAAATA TTTAATTTT AC TCACTAC
inhAct[m2-i87]-t	AATAGAG CAAACCCAC CTCTATT ATATAT TTTAATTTT AC TCACTAC
inhAct[m2-i88]-t	AATAGAG CATCTCTCC CTCTATT ATAAAT TTTAATTTT AC TCACTAC
inhAct[m2-i89]-t	AATAGAG CATATCAAC CTCTATT ATCTCA TTTAATTTT AC TCACTAC
inhAct[m2-i90]-t	AATAGAG CACATCCAC CTCTATT ACTACA TTTAATTTT AC TCACTAC
inhAct[m2-i91]-t	AATAGAG CATCAAACC CTCTATT ACAACA TTTAATTTT AC TCACTAC
inhAct[m2-i92]-t	AATAGAG CAAACTTTC CTCTATT ACACAC TTTAATTTT AC TCACTAC
inhAct[m2-i93]-t	AATAGAG CACCTACCC CTCTATT ATCTAT TTTAATTTT AC TCACTAC
inhAct[m2-i94]-t	AATAGAG CACCTTTAC CTCTATT AATTCC TTTAATTTT AC TCACTAC
inhAct[m2-i95]-t	AATAGAG CATTCCTTC CTCTATT ACATAT TTTAATTTT AC TCACTAC
inhAct[m2-i96]-t	AATAGAG CACAACTCC CTCTATT ACCATT TTTAATTTT AC TCACTAC
inhAct[m2-i97]-t	AATAGAG CAATACTCC CTCTATT ATCAAT TTTAATTTT AC TCACTAC
inhAct[m2-i98]-t	AATAGAG CACTTATCC CTCTATT AACATT TTTAATTTT AC TCACTAC
inhAct[m2-i99]-t	AATAGAG CACATAATC CTCTATT ACCCTT TTTAATTTT AC TCACTAC
inhAct[m2-i100]-t	AATAGAG CATACCCTC CTCTATT ACTAAC TTTAATTTT AC TCACTAC

Table S5: DNA sequences of bottom strands in the learning gates for memory 2.

Name	Sequence
inhAct[m1-i1]-b	GG AAAATTAAA AAGTGTGGGAAGTTG TCTTTCA TTATCCTCC AAAATTAAA TT
inhAct[m1-i2]-b	GG AAAATTAAA TAAGTTGGTTAGATG TCTTTCA TCCAAATTC AAAATTAAA TT
inhAct[m1-i3]-b	GG AAAATTAAA AGTATTGTTTAGGTG TCTTTCA ACACCATCC AAAATTAAA TT
inhAct[m1-i4]-b	GG AAAATTAAA GATTGTGATGTAATG TCTTTCA CTCCACTCC AAAATTAAA TT
inhAct[m1-i5]-b	GG AAAATTAAA GATAATGTAAGGTTG TCTTTCA CCTTATTAC AAAATTAAA TT
inhAct[m1-i6]-b	GG AAAATTAAA TAAGATGATGGATTG TCTTTCA CCTCAAAAC AAAATTAAA TT
inhAct[m1-i7]-b	GG AAAATTAAA GTGGATGTAGTGATG TCTTTCA ATCTCTCTC AAAATTAAA TT
inhAct[m1-i8]-b	GG AAAATTAAA AGGTATGTTGGGATG TCTTTCA CATTTCTTC AAAATTAAA TT
inhAct[m1-i9]-b	GG AAAATTAAA TGGGTTGGGAGATTG TCTTTCA AACCCTACC AAAATTAAA TT
inhAct[m1-i10]-b	GG AAAATTAAA AAAGTTGTGGAAATG TCTTTCA TCATATCTC AAAATTAAA TT
inhAct[m1-i11]-b	GG AAAATTAAA AGTGGTGAAAAGATG TCTTTCA CAATCATAC AAAATTAAA TT
inhAct[m1-i12]-b	GG AAAATTAAA GAGTTTGAGTAATTG TCTTTCA CCATTTCCC AAAATTAAA TT
inhAct[m1-i13]-b	GG AAAATTAAA GAGGTTGTGATAGTG TCTTTCA AACATAACC AAAATTAAA TT
inhAct[m1-i14]-b	GG AAAATTAAA GGGTGTGTTGTTGTG TCTTTCA CTTTAACAC AAAATTAAA TT
inhAct[m1-i15]-b	GG AAAATTAAA GGATATGTATGATTG TCTTTCA CACCTATAC AAAATTAAA TT
inhAct[m1-i16]-b	GG AAAATTAAA TAGTATGAAAGGGTG TCTTTCA CCCATCAAC AAAATTAAA TT
inhAct[m1-i17]-b	GG AAAATTAAA GGGATTGAGTGGATG TCTTTCA ATTCTCATC AAAATTAAA TT
inhAct[m1-i18]-b	GG AAAATTAAA GATTATGTGAGTATG TCTTTCA TAATAACCC AAAATTAAA TT
inhAct[m1-i19]-b	GG AAAATTAAA AGAAGTGGGTGTATG TCTTTCA TACTTATCC AAAATTAAA TT
inhAct[m1-i20]-b	GG AAAATTAAA AGATATGAGAAGGTG TCTTTCA ATTCAACTC AAAATTAAA TT
inhAct[m1-i21]-b	GG AAAATTAAA AGGGTTGAAATAATG TCTTTCA CTCACTTCC AAAATTAAA TT
inhAct[m1-i22]-b	GG AAAATTAAA GGAAGTGTATTAATG TCTTTCA TATCACATC AAAATTAAA TT
inhAct[m1-i23]-b	GG AAAATTAAA ATGTGTGTGTAAATG TCTTTCA TCTCATTCC AAAATTAAA TT
inhAct[m1-i24]-b	GG AAAATTAAA GAAGATGGAATTATG TCTTTCA ATCACAAAC AAAATTAAA TT
inhAct[m1-i25]-b	GG AAAATTAAA AGGAGTGAAGTGTTG TCTTTCA ATTTCCAAC AAAATTAAA TT
inhAct[m1-i26]-b	GG AAAATTAAA GATAGTGGATAGATG TCTTTCA CATAAAACC AAAATTAAA TT
inhAct[m1-i27]-b	GG AAAATTAAA TAATTTGTAGGAGTG TCTTTCA TTCCCTTTC AAAATTAAA TT
inhAct[m1-i28]-b	GG AAAATTAAA GGGTATGAGGTAGTG TCTTTCA ACTATACAC AAAATTAAA TT
inhAct[m1-i29]-b	GG AAAATTAAA TAGAGTGTAGGTATG TCTTTCA TCCTTTTAC AAAATTAAA TT
inhAct[m1-i30]-b	GG AAAATTAAA GAAGGTGTGTGAGTG TCTTTCA TATACCACC AAAATTAAA TT
inhAct[m1-i31]-b	GG AAAATTAAA TAGAATGTGTAGTTG TCTTTCA TTACCTAAC AAAATTAAA TT
inhAct[m1-i32]-b	GG AAAATTAAA TGGTTTGTGAATATG TCTTTCA TCAATATCC AAAATTAAA TT
inhAct[m1-i33]-b	GG AAAATTAAA AGTTTTGAATGTGTG TCTTTCA CTCCTCCAC AAAATTAAA TT
inhAct[m1-i34]-b	GG AAAATTAAA TAAAGTGGAAAAGTG TCTTTCA ACCCACAAC AAAATTAAA TT
inhAct[m1-i35]-b	GG AAAATTAAA GTGGGTGGGTATTTG TCTTTCA AACAACTTC AAAATTAAA TT
inhAct[m1-i36]-b	GG AAAATTAAA AATAATGATGTGGTG TCTTTCA AAATACCTC AAAATTAAA TT
inhAct[m1-i37]-b	GG AAAATTAAA TATATTGAAGGGATG TCTTTCA TCCCAATAC AAAATTAAA TT
inhAct[m1-i38]-b	GG AAAATTAAA GTTGTTGATGAATTG TCTTTCA CATCCTCTC AAAATTAAA TT
inhAct[m1-i39]-b	GG AAAATTAAA AGGGATGTGTTTATG TCTTTCA TCTACTTAC AAAATTAAA TT
inhAct[m1-i40]-b	GG AAAATTAAA GTTTGTGTATATGTG TCTTTCA AAAACCTAC AAAATTAAA TT
inhAct[m1-i41]-b	GG AAAATTAAA GAAAGTGATAGGATG TCTTTCA CAACCACCC AAAATTAAA TT
inhAct[m1-i42]-b	GG AAAATTAAA GTGATTGTGGTTTTG TCTTTCA TATTCCCTC AAAATTAAA TT
inhAct[m1-i43]-b	GG AAAATTAAA GGTGTTGTTATGATG TCTTTCA ACCTTAAAC AAAATTAAA TT
inhAct[m1-i44]-b	GG AAAATTAAA GTGAATGAATTTGTG TCTTTCA CCTATTTTC AAAATTAAA TT
inhAct[m1-i45]-b	GG AAAATTAAA AGTTATGGGATAATG TCTTTCA AACCAAATC AAAATTAAA TT
inhAct[m1-i46]-b	GG AAAATTAAA ATGAGTGAGATATTG TCTTTCA AATCTACCC AAAATTAAA TT
inhAct[m1-i47]-b	GG AAAATTAAA TGAAATGGATTGGTG TCTTTCA TACCCAAAC AAAATTAAA TT
inhAct[m1-i48]-b	GG AAAATTAAA GGGAATGTTAAAATG TCTTTCA CCTTCAATC AAAATTAAA TT
inhAct[m1-i49]-b	GG AAAATTAAA TGTGTTGAGGAGGTG TCTTTCA TTAATTCCC AAAATTAAA TT
inhAct[m1-i50]-b	GG AAAATTAAA GTTGGTGTAAGAATG TCTTTCA ACAAAACTC AAAATTAAA TT
inhAct[m1-i51]-b	GG AAAATTAAA GTTAATGGGAATGTG TCTTTCA ATTAATCCC AAAATTAAA TT

Name	Sequence
inhAct[m1-i52]-b	GG AAAATTAAA TAGTTTGGATGGTTG TCTTTCA CCAACTCAC AAAATTAAA TT
inhAct[m1-i53]-b	GG AAAATTAAA GTAAATGGTTGTTTG TCTTTCA TAAACACTC AAAATTAAA TT
inhAct[m1-i54]-b	GG AAAATTAAA GGTATTGGAGGATTG TCTTTCA CCCTCCTTC AAAATTAAA TT
inhAct[m1-i55]-b	GG AAAATTAAA ATAGGTGGTGTGATG TCTTTCA CTATATACC AAAATTAAA TT
inhAct[m1-i56]-b	GG AAAATTAAA AGTGATGTTTGGTTG TCTTTCA CTACAATTC AAAATTAAA TT
inhAct[m1-i57]-b	GG AAAATTAAA AGATTTGAGGTTATG TCTTTCA CATATTCAC AAAATTAAA TT
inhAct[m1-i58]-b	GG AAAATTAAA TGATGTGGAATATTG TCTTTCA AAACTTTCC AAAATTAAA TT
inhAct[m1-i59]-b	GG AAAATTAAA GTTATTGGTATAGTG TCTTTCA CTCTTCACC AAAATTAAA TT
inhAct[m1-i60]-b	GG AAAATTAAA TAGATTGGGATTTTG TCTTTCA ATATTACCC AAAATTAAA TT
inhAct[m1-i61]-b	GG AAAATTAAA ATGTATGGTTTAGTG TCTTTCA CATTACACC AAAATTAAA TT
inhAct[m1-i62]-b	GG AAAATTAAA GGATTTGGGTTGTTG TCTTTCA ACTTTTACC AAAATTAAA TT
inhAct[m1-i63]-b	GG AAAATTAAA TAGGGTGTTTTATTG TCTTTCA ACCATATTC AAAATTAAA TT
inhAct[m1-i64]-b	GG AAAATTAAA TGTTTTGGTGATTTG TCTTTCA CCTCCCAAC AAAATTAAA TT
inhAct[m1-i65]-b	GG AAAATTAAA GTGTTTGATAAGATG TCTTTCA TCCACTACC AAAATTAAA TT
inhAct[m1-i66]-b	GG AAAATTAAA GAGGATGGTGGTATG TCTTTCA TCCCTTATC AAAATTAAA TT
inhAct[m1-i67]-b	GG AAAATTAAA TGAGGTGAGTTTGTG TCTTTCA TTTCCATTC AAAATTAAA TT
inhAct[m1-i68]-b	GG AAAATTAAA AAAATTGGTGGGTTG TCTTTCA ACACACTTC AAAATTAAA TT
inhAct[m1-i69]-b	GG AAAATTAAA TAATATGGGTGAATG TCTTTCA CCACTAATC AAAATTAAA TT
inhAct[m1-i70]-b	GG AAAATTAAA GTTTATGATTGGGTG TCTTTCA AATTCACAC AAAATTAAA TT
inhAct[m1-i71]-b	GG AAAATTAAA ATGGTTGTTGGTGTG TCTTTCA TTATTCCTC AAAATTAAA TT
inhAct[m1-i72]-b	GG AAAATTAAA ATAAGTGTTAGAGTG TCTTTCA ACTCTCTAC AAAATTAAA TT
inhAct[m1-i73]-b	GG AAAATTAAA GTAGGTGAATTGTTG TCTTTCA TTCAAACAC AAAATTAAA TT
inhAct[m1-i74]-b	GG AAAATTAAA GGGAGTGGTAGGGTG TCTTTCA CTCATTATC AAAATTAAA TT
inhAct[m1-i75]-b	GG AAAATTAAA ATAATTGGAAGTGTG TCTTTCA TAACACCAC AAAATTAAA TT
inhAct[m1-i76]-b	GG AAAATTAAA GGTGATGATATTGTG TCTTTCA ACATCTATC AAAATTAAA TT
inhAct[m1-i77]-b	GG AAAATTAAA AATGATGAGAGTTTG TCTTTCA CAACATATC AAAATTAAA TT
inhAct[m1-i78]-b	GG AAAATTAAA GGTGGTGTGAAATTG TCTTTCA AACTCTTAC AAAATTAAA TT
inhAct[m1-i79]-b	GG AAAATTAAA ATGAATGGTAAGTTG TCTTTCA ATCCATCAC AAAATTAAA TT
inhAct[m1-i80]-b	GG AAAATTAAA ATAGTTGTATGGATG TCTTTCA CTTCTTAAC AAAATTAAA TT
inhAct[m1-i81]-b	GG AAAATTAAA GAATGTGTAGAGTTG TCTTTCA ATAACCATC AAAATTAAA TT
inhAct[m1-i82]-b	GG AAAATTAAA GAAAATGAGGATTTG TCTTTCA CACTATTTC AAAATTAAA TT
inhAct[m1-i83]-b	GG AAAATTAAA ATGTTTGTAATGGTG TCTTTCA ATACTTCTC AAAATTAAA TT
inhAct[m1-i84]-b	GG AAAATTAAA GGAGTTGAATATATG TCTTTCA TCACCCTAC AAAATTAAA TT
inhAct[m1-i85]-b	GG AAAATTAAA TGTAATGAGTGATTG TCTTTCA TACACCCAC AAAATTAAA TT
inhAct[m1-i86]-b	GG AAAATTAAA TATTTTGTGTTGGTG TCTTTCA CCAATCTCC AAAATTAAA TT
inhAct[m1-i87]-b	GG AAAATTAAA ATATATGTGGGTTTG TCTTTCA CACAATAAC AAAATTAAA TT
inhAct[m1-i88]-b	GG AAAATTAAA ATTTATGGAGAGATG TCTTTCA CAAAATTCC AAAATTAAA TT
inhAct[m1-i89]-b	GG AAAATTAAA TGAGATGTTGATATG TCTTTCA AATACTCCC AAAATTAAA TT
inhAct[m1-i90]-b	GG AAAATTAAA TGTAGTGTGGATGTG TCTTTCA TCTAACCTC AAAATTAAA TT
inhAct[m1-i91]-b	GG AAAATTAAA TGTTGTGGTTTGATG TCTTTCA TCACTCACC AAAATTAAA TT
inhAct[m1-i92]-b	GG AAAATTAAA GTGTGTGAAAGTTTG TCTTTCA TTCTCAACC AAAATTAAA TT
inhAct[m1-i93]-b	GG AAAATTAAA ATAGATGGGTAGGTG TCTTTCA CTTACATAC AAAATTAAA TT
inhAct[m1-i94]-b	GG AAAATTAAA GGAATTGTAAAGGTG TCTTTCA ATCTACTAC AAAATTAAA TT
inhAct[m1-i95]-b	GG AAAATTAAA ATATGTGAAGGAATG TCTTTCA CTAAACATC AAAATTAAA TT
inhAct[m1-i96]-b	GG AAAATTAAA AATGGTGGAGTTGTG TCTTTCA ACTTAATCC AAAATTAAA TT
inhAct[m1-i97]-b	GG AAAATTAAA ATTGATGGAGTATTG TCTTTCA TTTTCTCAC AAAATTAAA TT
inhAct[m1-i98]-b	GG AAAATTAAA AATGTTGGATAAGTG TCTTTCA TCTTTACTC AAAATTAAA TT
inhAct[m1-i99]-b	GG AAAATTAAA AAGGGTGATTATGTG TCTTTCA TTTATCCAC AAAATTAAA TT
inhAct[m1-i100]-b	GG AAAATTAAA GTTAGTGAGGGTATG TCTTTCA TCCTAAATC AAAATTAAA TT

Table S6: DNA sequences of top strands in the weight gates for memory 1.

Name	Sequence
inhW[m1-i1]-t	CACACTATAATTCCA TCT CAACTTCCCACACTT AA TTTAATTTT GGAGGATAA TGAAAGA
inhW[m1-i2]-t	CACACTATAATTCCA TCT CATCTAACCAACTTA AA TTTAATTTT GAATTTGGA TGAAAGA
inhW[m1-i3]-t	CACACTATAATTCCA TCT CACCTAAACAATACT AA TTTAATTTT GGATGGTGT TGAAAGA
inhW[m1-i4]-t	CACACTATAATTCCA TCT CATTACATCACAATC AA TTTAATTTT GGAGTGGAG TGAAAGA
inhW[m1-i5]-t	CACACTATAATTCCA TCT CAACCTTACATTATC AA TTTAATTTT GTAATAAGG TGAAAGA
inhW[m1-i6]-t	CACACTATAATTCCA TCT CAATCCATCATCTTA AA TTTAATTTT GTTTTGAGG TGAAAGA
inhW[m1-i7]-t	CACACTATAATTCCA TCT CATCACTACATCCAC AA TTTAATTTT GAGAGAGAT TGAAAGA
inhW[m1-i8]-t	CACACTATAATTCCA TCT CATCCCAACATACCT AA TTTAATTTT GAAGAAATG TGAAAGA
inhW[m1-i9]-t	CACACTATAATTCCA TCT CAATCTCCCAACCCA AA TTTAATTTT GGTAGGGTT TGAAAGA
inhW[m1-i10]-t	CACACTATAATTCCA TCT CATTTCCACAACTTT AA TTTAATTTT GAGATATGA TGAAAGA
inhW[m1-i11]-t	CACACTATAATTCCA TCT CATCTTTTCACCACT AA TTTAATTTT GTATGATTG TGAAAGA
inhW[m1-i12]-t	CACACTATAATTCCA TCT CAATTACTCAAACTC AA TTTAATTTT GGGAAATGG TGAAAGA
inhW[m1-i13]-t	CACACTATAATTCCA TCT CACTATCACAACCTC AA TTTAATTTT GGTTATGTT TGAAAGA
inhW[m1-i14]-t	CACACTATAATTCCA TCT CACAACAACACACCC AA TTTAATTTT GTGTTAAAG TGAAAGA
inhW[m1-i15]-t	CACACTATAATTCCA TCT CAATCATACATATCC AA TTTAATTTT GTATAGGTG TGAAAGA
inhW[m1-i16]-t	CACACTATAATTCCA TCT CACCCTTTCATACTA AA TTTAATTTT GTTGATGGG TGAAAGA
inhW[m1-i17]-t	CACACTATAATTCCA TCT CATCCACTCAATCCC AA TTTAATTTT GATGAGAAT TGAAAGA
inhW[m1-i18]-t	CACACTATAATTCCA TCT CATACTCACATAATC AA TTTAATTTT GGGTTATTA TGAAAGA
inhW[m1-i19]-t	CACACTATAATTCCA TCT CATACACCCACTTCT AA TTTAATTTT GGATAAGTA TGAAAGA
inhW[m1-i20]-t	CACACTATAATTCCA TCT CACCTTCTCATATCT AA TTTAATTTT GAGTTGAAT TGAAAGA
inhW[m1-i21]-t	CACACTATAATTCCA TCT CATTATTTCAACCCT AA TTTAATTTT GGAAGTGAG TGAAAGA
inhW[m1-i22]-t	CACACTATAATTCCA TCT CATTAATACACTTCC AA TTTAATTTT GATGTGATA TGAAAGA
inhW[m1-i23]-t	CACACTATAATTCCA TCT CATTTACACACACAT AA TTTAATTTT GGAATGAGA TGAAAGA
inhW[m1-i24]-t	CACACTATAATTCCA TCT CATAATTCCATCTTC AA TTTAATTTT GTTTGTGAT TGAAAGA
inhW[m1-i25]-t	CACACTATAATTCCA TCT CAACACTTCACTCCT AA TTTAATTTT GTTGGAAAT TGAAAGA
inhW[m1-i26]-t	CACACTATAATTCCA TCT CATCTATCCACTATC AA TTTAATTTT GGTTTTATG TGAAAGA
inhW[m1-i27]-t	CACACTATAATTCCA TCT CACTCCTACAAATTA AA TTTAATTTT GAAAGGGAA TGAAAGA
inhW[m1-i28]-t	CACACTATAATTCCA TCT CACTACCTCATACCC AA TTTAATTTT GTGTATAGT TGAAAGA
inhW[m1-i29]-t	CACACTATAATTCCA TCT CATACCTACACTCTA AA TTTAATTTT GTAAAAGGA TGAAAGA
inhW[m1-i30]-t	CACACTATAATTCCA TCT CACTCACACACCTTC AA TTTAATTTT GGTGGTATA TGAAAGA
inhW[m1-i31]-t	CACACTATAATTCCA TCT CAACTACACATTCTA AA TTTAATTTT GTTAGGTAA TGAAAGA
inhW[m1-i32]-t	CACACTATAATTCCA TCT CATATTCACAAACCA AA TTTAATTTT GGATATTGA TGAAAGA
inhW[m1-i33]-t	CACACTATAATTCCA TCT CACACATTCAAAACT AA TTTAATTTT GTGGAGGAG TGAAAGA
inhW[m1-i34]-t	CACACTATAATTCCA TCT CACTTTTCCACTTTA AA TTTAATTTT GTTGTGGGT TGAAAGA
inhW[m1-i35]-t	CACACTATAATTCCA TCT CAAATACCCACCCAC AA TTTAATTTT GAAGTTGTT TGAAAGA
inhW[m1-i36]-t	CACACTATAATTCCA TCT CACCACATCATTATT AA TTTAATTTT GAGGTATTT TGAAAGA
inhW[m1-i37]-t	CACACTATAATTCCA TCT CATCCCTTCAATATA AA TTTAATTTT GTATTGGGA TGAAAGA
inhW[m1-i38]-t	CACACTATAATTCCA TCT CAATTCATCAACAAC AA TTTAATTTT GAGAGGATG TGAAAGA
inhW[m1-i39]-t	CACACTATAATTCCA TCT CATAAACACATCCCT AA TTTAATTTT GTAAGTAGA TGAAAGA
inhW[m1-i40]-t	CACACTATAATTCCA TCT CACATATACACAAAC AA TTTAATTTT GTAGGTTTT TGAAAGA
inhW[m1-i41]-t	CACACTATAATTCCA TCT CATCCTATCACTTTC AA TTTAATTTT GGGTGGTTG TGAAAGA
inhW[m1-i42]-t	CACACTATAATTCCA TCT CAAAACCACAATCAC AA TTTAATTTT GAGGGAATA TGAAAGA
inhW[m1-i43]-t	CACACTATAATTCCA TCT CATCATAACAACACC AA TTTAATTTT GTTTAAGGT TGAAAGA
inhW[m1-i44]-t	CACACTATAATTCCA TCT CACAAATTCATTCAC AA TTTAATTTT GAAAATAGG TGAAAGA
inhW[m1-i45]-t	CACACTATAATTCCA TCT CATTATCCCATAACT AA TTTAATTTT GATTTGGTT TGAAAGA
inhW[m1-i46]-t	CACACTATAATTCCA TCT CAATATCTCACTCAT AA TTTAATTTT GGGTAGATT TGAAAGA
inhW[m1-i47]-t	CACACTATAATTCCA TCT CACCAATCCATTTCA AA TTTAATTTT GTTTGGGTA TGAAAGA
inhW[m1-i48]-t	CACACTATAATTCCA TCT CATTTTAACATTCCC AA TTTAATTTT GATTGAAGG TGAAAGA
inhW[m1-i49]-t	CACACTATAATTCCA TCT CACCTCCTCAACACA AA TTTAATTTT GGGAATTAA TGAAAGA
inhW[m1-i50]-t	CACACTATAATTCCA TCT CATTCTTACACCAAC AA TTTAATTTT GAGTTTTGT TGAAAGA
inhW[m1-i51]-t	CACACTATAATTCCA TCT CACATTCCCATTAAC AA TTTAATTTT GGGATTAAT TGAAAGA

Name	Sequence
inhW[m1-i52]-t	CACACTATAATTCCA TCT CAACCATCCAAACTA AA TTTAATTTT GTGAGTTGG TGAAAGA
inhW[m1-i53]-t	CACACTATAATTCCA TCT CAAACAACCATTTAC AA TTTAATTTT GAGTGTTTA TGAAAGA
inhW[m1-i54]-t	CACACTATAATTCCA TCT CAATCCTCCAATACC AA TTTAATTTT GAAGGAGGG TGAAAGA
inhW[m1-i55]-t	CACACTATAATTCCA TCT CATCACACCACCTAT AA TTTAATTTT GGTATATAG TGAAAGA
inhW[m1-i56]-t	CACACTATAATTCCA TCT CAACCAAACATCACT AA TTTAATTTT GAATTGTAG TGAAAGA
inhW[m1-i57]-t	CACACTATAATTCCA TCT CATAACCTCAAATCT AA TTTAATTTT GTGAATATG TGAAAGA
inhW[m1-i58]-t	CACACTATAATTCCA TCT CAATATTCCACATCA AA TTTAATTTT GGAAAGTTT TGAAAGA
inhW[m1-i59]-t	CACACTATAATTCCA TCT CACTATACCAATAAC AA TTTAATTTT GGTGAAGAG TGAAAGA
inhW[m1-i60]-t	CACACTATAATTCCA TCT CAAAATCCCAATCTA AA TTTAATTTT GGGTAATAT TGAAAGA
inhW[m1-i61]-t	CACACTATAATTCCA TCT CACTAAACCATACAT AA TTTAATTTT GGTGTAATG TGAAAGA
inhW[m1-i62]-t	CACACTATAATTCCA TCT CAACAACCCAAATCC AA TTTAATTTT GGTAAAAGT TGAAAGA
inhW[m1-i63]-t	CACACTATAATTCCA TCT CAATAAAACACCCTA AA TTTAATTTT GAATATGGT TGAAAGA
inhW[m1-i64]-t	CACACTATAATTCCA TCT CAAATCACCAAAACA AA TTTAATTTT GTTGGGAGG TGAAAGA
inhW[m1-i65]-t	CACACTATAATTCCA TCT CATCTTATCAAACAC AA TTTAATTTT GGTAGTGGA TGAAAGA
inhW[m1-i66]-t	CACACTATAATTCCA TCT CATACCACCATCCTC AA TTTAATTTT GATAAGGGA TGAAAGA
inhW[m1-i67]-t	CACACTATAATTCCA TCT CACAAACTCACCTCA AA TTTAATTTT GAATGGAAA TGAAAGA
inhW[m1-i68]-t	CACACTATAATTCCA TCT CAACCCACCAATTTT AA TTTAATTTT GAAGTGTGT TGAAAGA
inhW[m1-i69]-t	CACACTATAATTCCA TCT CATTCACCCATATTA AA TTTAATTTT GATTAGTGG TGAAAGA
inhW[m1-i70]-t	CACACTATAATTCCA TCT CACCCAATCATAAAC AA TTTAATTTT GTGTGAATT TGAAAGA
inhW[m1-i71]-t	CACACTATAATTCCA TCT CACACCAACAACCAT AA TTTAATTTT GAGGAATAA TGAAAGA
inhW[m1-i72]-t	CACACTATAATTCCA TCT CACTCTAACACTTAT AA TTTAATTTT GTAGAGAGT TGAAAGA
inhW[m1-i73]-t	CACACTATAATTCCA TCT CAACAATTCACCTAC AA TTTAATTTT GTGTTTGAA TGAAAGA
inhW[m1-i74]-t	CACACTATAATTCCA TCT CACCCTACCACTCCC AA TTTAATTTT GATAATGAG TGAAAGA
inhW[m1-i75]-t	CACACTATAATTCCA TCT CACACTTCCAATTAT AA TTTAATTTT GTGGTGTTA TGAAAGA
inhW[m1-i76]-t	CACACTATAATTCCA TCT CACAATATCATCACC AA TTTAATTTT GATAGATGT TGAAAGA
inhW[m1-i77]-t	CACACTATAATTCCA TCT CAAACTCTCATCATT AA TTTAATTTT GATATGTTG TGAAAGA
inhW[m1-i78]-t	CACACTATAATTCCA TCT CAATTTCACACCACC AA TTTAATTTT GTAAGAGTT TGAAAGA
inhW[m1-i79]-t	CACACTATAATTCCA TCT CAACTTACCATTCAT AA TTTAATTTT GTGATGGAT TGAAAGA
inhW[m1-i80]-t	CACACTATAATTCCA TCT CATCCATACAACTAT AA TTTAATTTT GTTAAGAAG TGAAAGA
inhW[m1-i81]-t	CACACTATAATTCCA TCT CAACTCTACACATTC AA TTTAATTTT GATGGTTAT TGAAAGA
inhW[m1-i82]-t	CACACTATAATTCCA TCT CAAATCCTCATTTTC AA TTTAATTTT GAAATAGTG TGAAAGA
inhW[m1-i83]-t	CACACTATAATTCCA TCT CACCATTACAAACAT AA TTTAATTTT GAGAAGTAT TGAAAGA
inhW[m1-i84]-t	CACACTATAATTCCA TCT CATATATTCAACTCC AA TTTAATTTT GTAGGGTGA TGAAAGA
inhW[m1-i85]-t	CACACTATAATTCCA TCT CAATCACTCATTACA AA TTTAATTTT GTGGGTGTA TGAAAGA
inhW[m1-i86]-t	CACACTATAATTCCA TCT CACCAACACAAAATA AA TTTAATTTT GGAGATTGG TGAAAGA
inhW[m1-i87]-t	CACACTATAATTCCA TCT CAAACCCACATATAT AA TTTAATTTT GTTATTGTG TGAAAGA
inhW[m1-i88]-t	CACACTATAATTCCA TCT CATCTCTCCATAAAT AA TTTAATTTT GGAATTTTG TGAAAGA
inhW[m1-i89]-t	CACACTATAATTCCA TCT CATATCAACATCTCA AA TTTAATTTT GGGAGTATT TGAAAGA
inhW[m1-i90]-t	CACACTATAATTCCA TCT CACATCCACACTACA AA TTTAATTTT GAGGTTAGA TGAAAGA
inhW[m1-i91]-t	CACACTATAATTCCA TCT CATCAAACCACAACA AA TTTAATTTT GGTGAGTGA TGAAAGA
inhW[m1-i92]-t	CACACTATAATTCCA TCT CAAACTTTCACACAC AA TTTAATTTT GGTTGAGAA TGAAAGA
inhW[m1-i93]-t	CACACTATAATTCCA TCT CACCTACCCATCTAT AA TTTAATTTT GTATGTAAG TGAAAGA
inhW[m1-i94]-t	CACACTATAATTCCA TCT CACCTTTACAATTCC AA TTTAATTTT GTAGTAGAT TGAAAGA
inhW[m1-i95]-t	CACACTATAATTCCA TCT CATTCCTTCACATAT AA TTTAATTTT GATGTTTAG TGAAAGA
inhW[m1-i96]-t	CACACTATAATTCCA TCT CACAACTCCACCATT AA TTTAATTTT GGATTAAGT TGAAAGA
inhW[m1-i97]-t	CACACTATAATTCCA TCT CAATACTCCATCAAT AA TTTAATTTT GTGAGAAAA TGAAAGA
inhW[m1-i98]-t	CACACTATAATTCCA TCT CACTTATCCAACATT AA TTTAATTTT GAGTAAAGA TGAAAGA
inhW[m1-i99]-t	CACACTATAATTCCA TCT CACATAATCACCCTT AA TTTAATTTT GTGGATAAA TGAAAGA
inhW[m1-i100]-t	CACACTATAATTCCA TCT CATACCCTCACTAAC AA TTTAATTTT GATTTAGGA TGAAAGA

Table S7: DNA sequences of bottom strands in the weight gates for memory 1.

Name	Sequence
inhW[m1-i1]-b	CA TTATCCTCC AAAATTAAA AAGTGTGGGAAGTTG AGA TG
inhW[m1-i2]-b	CA TCCAAATTC AAAATTAAA TAAGTTGGTTAGATG AGA TG
inhW[m1-i3]-b	CA ACACCATCC AAAATTAAA AGTATTGTTTAGGTG AGA TG
inhW[m1-i4]-b	CA CTCCACTCC AAAATTAAA GATTGTGATGTAATG AGA TG
inhW[m1-i5]-b	CA CCTTATTAC AAAATTAAA GATAATGTAAGGTTG AGA TG
inhW[m1-i6]-b	CA CCTCAAAAC AAAATTAAA TAAGATGATGGATTG AGA TG
inhW[m1-i7]-b	CA ATCTCTCTC AAAATTAAA GTGGATGTAGTGATG AGA TG
inhW[m1-i8]-b	CA CATTTCTTC AAAATTAAA AGGTATGTTGGGATG AGA TG
inhW[m1-i9]-b	CA AACCCTACC AAAATTAAA TGGGTTGGGAGATTG AGA TG
inhW[m1-i10]-b	CA TCATATCTC AAAATTAAA AAAGTTGTGGAAATG AGA TG
inhW[m1-i11]-b	CA CAATCATAC AAAATTAAA AGTGGTGAAAAGATG AGA TG
inhW[m1-i12]-b	CA CCATTTCCC AAAATTAAA GAGTTTGAGTAATTG AGA TG
inhW[m1-i13]-b	CA AACATAACC AAAATTAAA GAGGTTGTGATAGTG AGA TG
inhW[m1-i14]-b	CA CTTTAACAC AAAATTAAA GGGTGTGTTGTTGTG AGA TG
inhW[m1-i15]-b	CA CACCTATAC AAAATTAAA GGATATGTATGATTG AGA TG
inhW[m1-i16]-b	CA CCCATCAAC AAAATTAAA TAGTATGAAAGGGTG AGA TG
inhW[m1-i17]-b	CA ATTCTCATC AAAATTAAA GGGATTGAGTGGATG AGA TG
inhW[m1-i18]-b	CA TAATAACCC AAAATTAAA GATTATGTGAGTATG AGA TG
inhW[m1-i19]-b	CA TACTTATCC AAAATTAAA AGAAGTGGGTGTATG AGA TG
inhW[m1-i20]-b	CA ATTCAACTC AAAATTAAA AGATATGAGAAGGTG AGA TG
inhW[m1-i21]-b	CA CTCACTTCC AAAATTAAA AGGGTTGAAATAATG AGA TG
inhW[m1-i22]-b	CA TATCACATC AAAATTAAA GGAAGTGTATTAATG AGA TG
inhW[m1-i23]-b	CA TCTCATTCC AAAATTAAA ATGTGTGTGTAAATG AGA TG
inhW[m1-i24]-b	CA ATCACAAAC AAAATTAAA GAAGATGGAATTATG AGA TG
inhW[m1-i25]-b	CA ATTTCCAAC AAAATTAAA AGGAGTGAAGTGTTG AGA TG
inhW[m1-i26]-b	CA CATAAAACC AAAATTAAA GATAGTGGATAGATG AGA TG
inhW[m1-i27]-b	CA TTCCCTTTC AAAATTAAA TAATTTGTAGGAGTG AGA TG
inhW[m1-i28]-b	CA ACTATACAC AAAATTAAA GGGTATGAGGTAGTG AGA TG
inhW[m1-i29]-b	CA TCCTTTTAC AAAATTAAA TAGAGTGTAGGTATG AGA TG
inhW[m1-i30]-b	CA TATACCACC AAAATTAAA GAAGGTGTGTGAGTG AGA TG
inhW[m1-i31]-b	CA TTACCTAAC AAAATTAAA TAGAATGTGTAGTTG AGA TG
inhW[m1-i32]-b	CA TCAATATCC AAAATTAAA TGGTTTGTGAATATG AGA TG
inhW[m1-i33]-b	CA CTCCTCCAC AAAATTAAA AGTTTTGAATGTGTG AGA TG
inhW[m1-i34]-b	CA ACCCACAAC AAAATTAAA TAAAGTGGAAAAGTG AGA TG
inhW[m1-i35]-b	CA AACAACTTC AAAATTAAA GTGGGTGGGTATTTG AGA TG
inhW[m1-i36]-b	CA AAATACCTC AAAATTAAA AATAATGATGTGGTG AGA TG
inhW[m1-i37]-b	CA TCCCAATAC AAAATTAAA TATATTGAAGGGATG AGA TG
inhW[m1-i38]-b	CA CATCCTCTC AAAATTAAA GTTGTTGATGAATTG AGA TG
inhW[m1-i39]-b	CA TCTACTTAC AAAATTAAA AGGGATGTGTTTATG AGA TG
inhW[m1-i40]-b	CA AAAACCTAC AAAATTAAA GTTTGTGTATATGTG AGA TG
inhW[m1-i41]-b	CA CAACCACCC AAAATTAAA GAAAGTGATAGGATG AGA TG
inhW[m1-i42]-b	CA TATTCCCTC AAAATTAAA GTGATTGTGGTTTTG AGA TG
inhW[m1-i43]-b	CA ACCTTAAAC AAAATTAAA GGTGTTGTTATGATG AGA TG
inhW[m1-i44]-b	CA CCTATTTTC AAAATTAAA GTGAATGAATTTGTG AGA TG
inhW[m1-i45]-b	CA AACCAAATC AAAATTAAA AGTTATGGGATAATG AGA TG
inhW[m1-i46]-b	CA AATCTACCC AAAATTAAA ATGAGTGAGATATTG AGA TG
inhW[m1-i47]-b	CA TACCCAAAC AAAATTAAA TGAAATGGATTGGTG AGA TG
inhW[m1-i48]-b	CA CCTTCAATC AAAATTAAA GGGAATGTTAAAATG AGA TG
inhW[m1-i49]-b	CA TTAATTCCC AAAATTAAA TGTGTTGAGGAGGTG AGA TG
inhW[m1-i50]-b	CA ACAAAACTC AAAATTAAA GTTGGTGTAAGAATG AGA TG
inhW[m1-i51]-b	CA ATTAATCCC AAAATTAAA GTTAATGGGAATGTG AGA TG

Name	Sequence
inhW[m1-i52]-b	CA CCAACTCAC AAAATTAAA TAGTTTGGATGGTTG AGA TG
inhW[m1-i53]-b	CA TAAACACTC AAAATTAAA GTAAATGGTTGTTTG AGA TG
inhW[m1-i54]-b	CA CCCTCCTTC AAAATTAAA GGTATTGGAGGATTG AGA TG
inhW[m1-i55]-b	CA CTATATACC AAAATTAAA ATAGGTGGTGTGATG AGA TG
inhW[m1-i56]-b	CA CTACAATTC AAAATTAAA AGTGATGTTTGGTTG AGA TG
inhW[m1-i57]-b	CA CATATTCAC AAAATTAAA AGATTTGAGGTTATG AGA TG
inhW[m1-i58]-b	CA AAACTTTCC AAAATTAAA TGATGTGGAATATTG AGA TG
inhW[m1-i59]-b	CA CTCTTCACC AAAATTAAA GTTATTGGTATAGTG AGA TG
inhW[m1-i60]-b	CA ATATTACCC AAAATTAAA TAGATTGGGATTTTG AGA TG
inhW[m1-i61]-b	CA CATTACACC AAAATTAAA ATGTATGGTTTAGTG AGA TG
inhW[m1-i62]-b	CA ACTTTTACC AAAATTAAA GGATTTGGGTTGTTG AGA TG
inhW[m1-i63]-b	CA ACCATATTC AAAATTAAA TAGGGTGTTTTATTG AGA TG
inhW[m1-i64]-b	CA CCTCCCAAC AAAATTAAA TGTTTTGGTGATTTG AGA TG
inhW[m1-i65]-b	CA TCCACTACC AAAATTAAA GTGTTTGATAAGATG AGA TG
inhW[m1-i66]-b	CA TCCCTTATC AAAATTAAA GAGGATGGTGGTATG AGA TG
inhW[m1-i67]-b	CA TTTCCATTC AAAATTAAA TGAGGTGAGTTTGTG AGA TG
inhW[m1-i68]-b	CA ACACACTTC AAAATTAAA AAAATTGGTGGGTTG AGA TG
inhW[m1-i69]-b	CA CCACTAATC AAAATTAAA TAATATGGGTGAATG AGA TG
inhW[m1-i70]-b	CA AATTCACAC AAAATTAAA GTTTATGATTGGGTG AGA TG
inhW[m1-i71]-b	CA TTATTCCTC AAAATTAAA ATGGTTGTTGGTGTG AGA TG
inhW[m1-i72]-b	CA ACTCTCTAC AAAATTAAA ATAAGTGTTAGAGTG AGA TG
inhW[m1-i73]-b	CA TTCAAACAC AAAATTAAA GTAGGTGAATTGTTG AGA TG
inhW[m1-i74]-b	CA CTCATTATC AAAATTAAA GGGAGTGGTAGGGTG AGA TG
inhW[m1-i75]-b	CA TAACACCAC AAAATTAAA ATAATTGGAAGTGTG AGA TG
inhW[m1-i76]-b	CA ACATCTATC AAAATTAAA GGTGATGATATTGTG AGA TG
inhW[m1-i77]-b	CA CAACATATC AAAATTAAA AATGATGAGAGTTTG AGA TG
inhW[m1-i78]-b	CA AACTCTTAC AAAATTAAA GGTGGTGTGAAATTG AGA TG
inhW[m1-i79]-b	CA ATCCATCAC AAAATTAAA ATGAATGGTAAGTTG AGA TG
inhW[m1-i80]-b	CA CTTCTTAAC AAAATTAAA ATAGTTGTATGGATG AGA TG
inhW[m1-i81]-b	CA ATAACCATC AAAATTAAA GAATGTGTAGAGTTG AGA TG
inhW[m1-i82]-b	CA CACTATTTC AAAATTAAA GAAAATGAGGATTTG AGA TG
inhW[m1-i83]-b	CA ATACTTCTC AAAATTAAA ATGTTTGTAATGGTG AGA TG
inhW[m1-i84]-b	CA TCACCCTAC AAAATTAAA GGAGTTGAATATATG AGA TG
inhW[m1-i85]-b	CA TACACCCAC AAAATTAAA TGTAATGAGTGATTG AGA TG
inhW[m1-i86]-b	CA CCAATCTCC AAAATTAAA TATTTTGTGTTGGTG AGA TG
inhW[m1-i87]-b	CA CACAATAAC AAAATTAAA ATATATGTGGGTTTG AGA TG
inhW[m1-i88]-b	CA CAAAATTCC AAAATTAAA ATTTATGGAGAGATG AGA TG
inhW[m1-i89]-b	CA AATACTCCC AAAATTAAA TGAGATGTTGATATG AGA TG
inhW[m1-i90]-b	CA TCTAACCTC AAAATTAAA TGTAGTGTGGATGTG AGA TG
inhW[m1-i91]-b	CA TCACTCACC AAAATTAAA TGTTGTGGTTTGATG AGA TG
inhW[m1-i92]-b	CA TTCTCAACC AAAATTAAA GTGTGTGAAAGTTTG AGA TG
inhW[m1-i93]-b	CA CTTACATAC AAAATTAAA ATAGATGGGTAGGTG AGA TG
inhW[m1-i94]-b	CA ATCTACTAC AAAATTAAA GGAATTGTAAAGGTG AGA TG
inhW[m1-i95]-b	CA CTAAACATC AAAATTAAA ATATGTGAAGGAATG AGA TG
inhW[m1-i96]-b	CA ACTTAATCC AAAATTAAA AATGGTGGAGTTGTG AGA TG
inhW[m1-i97]-b	CA TTTTCTCAC AAAATTAAA ATTGATGGAGTATTG AGA TG
inhW[m1-i98]-b	CA TCTTTACTC AAAATTAAA AATGTTGGATAAGTG AGA TG
inhW[m1-i99]-b	CA TTTATCCAC AAAATTAAA AAGGGTGATTATGTG AGA TG
inhW[m1-i100]-b	CA TCCTAAATC AAAATTAAA GTTAGTGAGGGTATG AGA TG

Table S8: DNA sequences of top strands in the weight gates for memory 2.

Name	Sequence
inhW[m2-i1]-t	CACACAACAACCACA TCT CAACTTCCCACACTT AA TTTAATTTT GGAGGATAA AATAGAG
inhW[m2-i2]-t	CACACAACAACCACA TCT CATCTAACCAACTTA AA TTTAATTTT GAATTTGGA AATAGAG
inhW[m2-i3]-t	CACACAACAACCACA TCT CACCTAAACAATACT AA TTTAATTTT GGATGGTGT AATAGAG
inhW[m2-i4]-t	CACACAACAACCACA TCT CATTACATCACAATC AA TTTAATTTT GGAGTGGAG AATAGAG
inhW[m2-i5]-t	CACACAACAACCACA TCT CAACCTTACATTATC AA TTTAATTTT GTAATAAGG AATAGAG
inhW[m2-i6]-t	CACACAACAACCACA TCT CAATCCATCATCTTA AA TTTAATTTT GTTTTGAGG AATAGAG
inhW[m2-i7]-t	CACACAACAACCACA TCT CATCACTACATCCAC AA TTTAATTTT GAGAGAGAT AATAGAG
inhW[m2-i8]-t	CACACAACAACCACA TCT CATCCCAACATACCT AA TTTAATTTT GAAGAAATG AATAGAG
inhW[m2-i9]-t	CACACAACAACCACA TCT CAATCTCCCAACCCA AA TTTAATTTT GGTAGGGTT AATAGAG
inhW[m2-i10]-t	CACACAACAACCACA TCT CATTTCCACAACTTT AA TTTAATTTT GAGATATGA AATAGAG
inhW[m2-i11]-t	CACACAACAACCACA TCT CATCTTTTCACCACT AA TTTAATTTT GTATGATTG AATAGAG
inhW[m2-i12]-t	CACACAACAACCACA TCT CAATTACTCAAACTC AA TTTAATTTT GGGAAATGG AATAGAG
inhW[m2-i13]-t	CACACAACAACCACA TCT CACTATCACAACCTC AA TTTAATTTT GGTTATGTT AATAGAG
inhW[m2-i14]-t	CACACAACAACCACA TCT CACAACAACACACCC AA TTTAATTTT GTGTTAAAG AATAGAG
inhW[m2-i15]-t	CACACAACAACCACA TCT CAATCATACATATCC AA TTTAATTTT GTATAGGTG AATAGAG
inhW[m2-i16]-t	CACACAACAACCACA TCT CACCCTTTCATACTA AA TTTAATTTT GTTGATGGG AATAGAG
inhW[m2-i17]-t	CACACAACAACCACA TCT CATCCACTCAATCCC AA TTTAATTTT GATGAGAAT AATAGAG
inhW[m2-i18]-t	CACACAACAACCACA TCT CATACTCACATAATC AA TTTAATTTT GGGTTATTA AATAGAG
inhW[m2-i19]-t	CACACAACAACCACA TCT CATACACCCACTTCT AA TTTAATTTT GGATAAGTA AATAGAG
inhW[m2-i20]-t	CACACAACAACCACA TCT CACCTTCTCATATCT AA TTTAATTTT GAGTTGAAT AATAGAG
inhW[m2-i21]-t	CACACAACAACCACA TCT CATTATTTCAACCCT AA TTTAATTTT GGAAGTGAG AATAGAG
inhW[m2-i22]-t	CACACAACAACCACA TCT CATTAATACACTTCC AA TTTAATTTT GATGTGATA AATAGAG
inhW[m2-i23]-t	CACACAACAACCACA TCT CATTTACACACACAT AA TTTAATTTT GGAATGAGA AATAGAG
inhW[m2-i24]-t	CACACAACAACCACA TCT CATAATTCCATCTTC AA TTTAATTTT GTTTGTGAT AATAGAG
inhW[m2-i25]-t	CACACAACAACCACA TCT CAACACTTCACTCCT AA TTTAATTTT GTTGGAAAT AATAGAG
inhW[m2-i26]-t	CACACAACAACCACA TCT CATCTATCCACTATC AA TTTAATTTT GGTTTTATG AATAGAG
inhW[m2-i27]-t	CACACAACAACCACA TCT CACTCCTACAAATTA AA TTTAATTTT GAAAGGGAA AATAGAG
inhW[m2-i28]-t	CACACAACAACCACA TCT CACTACCTCATACCC AA TTTAATTTT GTGTATAGT AATAGAG
inhW[m2-i29]-t	CACACAACAACCACA TCT CATACCTACACTCTA AA TTTAATTTT GTAAAAGGA AATAGAG
inhW[m2-i30]-t	CACACAACAACCACA TCT CACTCACACACCTTC AA TTTAATTTT GGTGGTATA AATAGAG
inhW[m2-i31]-t	CACACAACAACCACA TCT CAACTACACATTCTA AA TTTAATTTT GTTAGGTAA AATAGAG
inhW[m2-i32]-t	CACACAACAACCACA TCT CATATTCACAAACCA AA TTTAATTTT GGATATTGA AATAGAG
inhW[m2-i33]-t	CACACAACAACCACA TCT CACACATTCAAAACT AA TTTAATTTT GTGGAGGAG AATAGAG
inhW[m2-i34]-t	CACACAACAACCACA TCT CACTTTTCCACTTTA AA TTTAATTTT GTTGTGGGT AATAGAG
inhW[m2-i35]-t	CACACAACAACCACA TCT CAAATACCCACCCAC AA TTTAATTTT GAAGTTGTT AATAGAG
inhW[m2-i36]-t	CACACAACAACCACA TCT CACCACATCATTATT AA TTTAATTTT GAGGTATTT AATAGAG
inhW[m2-i37]-t	CACACAACAACCACA TCT CATCCCTTCAATATA AA TTTAATTTT GTATTGGGA AATAGAG
inhW[m2-i38]-t	CACACAACAACCACA TCT CAATTCATCAACAAC AA TTTAATTTT GAGAGGATG AATAGAG
inhW[m2-i39]-t	CACACAACAACCACA TCT CATAAACACATCCCT AA TTTAATTTT GTAAGTAGA AATAGAG
inhW[m2-i40]-t	CACACAACAACCACA TCT CACATATACACAAAC AA TTTAATTTT GTAGGTTTT AATAGAG
inhW[m2-i41]-t	CACACAACAACCACA TCT CATCCTATCACTTTC AA TTTAATTTT GGGTGGTTG AATAGAG
inhW[m2-i42]-t	CACACAACAACCACA TCT CAAAACCACAATCAC AA TTTAATTTT GAGGGAATA AATAGAG
inhW[m2-i43]-t	CACACAACAACCACA TCT CATCATAACAACACC AA TTTAATTTT GTTTAAGGT AATAGAG
inhW[m2-i44]-t	CACACAACAACCACA TCT CACAAATTCATTCAC AA TTTAATTTT GAAAATAGG AATAGAG
inhW[m2-i45]-t	CACACAACAACCACA TCT CATTATCCCATAACT AA TTTAATTTT GATTTGGTT AATAGAG
inhW[m2-i46]-t	CACACAACAACCACA TCT CAATATCTCACTCAT AA TTTAATTTT GGGTAGATT AATAGAG
inhW[m2-i47]-t	CACACAACAACCACA TCT CACCAATCCATTTCA AA TTTAATTTT GTTTGGGTA AATAGAG
inhW[m2-i48]-t	CACACAACAACCACA TCT CATTTTAACATTCCC AA TTTAATTTT GATTGAAGG AATAGAG
inhW[m2-i49]-t	CACACAACAACCACA TCT CACCTCCTCAACACA AA TTTAATTTT GGGAATTAA AATAGAG
inhW[m2-i50]-t	CACACAACAACCACA TCT CATTCTTACACCAAC AA TTTAATTTT GAGTTTTGT AATAGAG
inhW[m2-i51]-t	CACACAACAACCACA TCT CACATTCCCATTAAC AA TTTAATTTT GGGATTAAT AATAGAG

Name	Sequence
inhW[m2-i52]-t	CACACAACAACCACA TCT CAACCATCCAAACTA AA TTTAATTTT GTGAGTTGG AATAGAG
inhW[m2-i53]-t	CACACAACAACCACA TCT CAAACAACCATTTAC AA TTTAATTTT GAGTGTTTA AATAGAG
inhW[m2-i54]-t	CACACAACAACCACA TCT CAATCCTCCAATACC AA TTTAATTTT GAAGGAGGG AATAGAG
inhW[m2-i55]-t	CACACAACAACCACA TCT CATCACACCACCTAT AA TTTAATTTT GGTATATAG AATAGAG
inhW[m2-i56]-t	CACACAACAACCACA TCT CAACCAAACATCACT AA TTTAATTTT GAATTGTAG AATAGAG
inhW[m2-i57]-t	CACACAACAACCACA TCT CATAACCTCAAATCT AA TTTAATTTT GTGAATATG AATAGAG
inhW[m2-i58]-t	CACACAACAACCACA TCT CAATATTCCACATCA AA TTTAATTTT GGAAAGTTT AATAGAG
inhW[m2-i59]-t	CACACAACAACCACA TCT CACTATACCAATAAC AA TTTAATTTT GGTGAAGAG AATAGAG
inhW[m2-i60]-t	CACACAACAACCACA TCT CAAAATCCCAATCTA AA TTTAATTTT GGGTAATAT AATAGAG
inhW[m2-i61]-t	CACACAACAACCACA TCT CACTAAACCATACAT AA TTTAATTTT GGTGTAATG AATAGAG
inhW[m2-i62]-t	CACACAACAACCACA TCT CAACAACCCAAATCC AA TTTAATTTT GGTAAAAGT AATAGAG
inhW[m2-i63]-t	CACACAACAACCACA TCT CAATAAAACACCCTA AA TTTAATTTT GAATATGGT AATAGAG
inhW[m2-i64]-t	CACACAACAACCACA TCT CAAATCACCAAAACA AA TTTAATTTT GTTGGGAGG AATAGAG
inhW[m2-i65]-t	CACACAACAACCACA TCT CATCTTATCAAACAC AA TTTAATTTT GGTAGTGGA AATAGAG
inhW[m2-i66]-t	CACACAACAACCACA TCT CATACCACCATCCTC AA TTTAATTTT GATAAGGGA AATAGAG
inhW[m2-i67]-t	CACACAACAACCACA TCT CACAAACTCACCTCA AA TTTAATTTT GAATGGAAA AATAGAG
inhW[m2-i68]-t	CACACAACAACCACA TCT CAACCCACCAATTTT AA TTTAATTTT GAAGTGTGT AATAGAG
inhW[m2-i69]-t	CACACAACAACCACA TCT CATTCACCCATATTA AA TTTAATTTT GATTAGTGG AATAGAG
inhW[m2-i70]-t	CACACAACAACCACA TCT CACCCAATCATAAAC AA TTTAATTTT GTGTGAATT AATAGAG
inhW[m2-i71]-t	CACACAACAACCACA TCT CACACCAACAACCAT AA TTTAATTTT GAGGAATAA AATAGAG
inhW[m2-i72]-t	CACACAACAACCACA TCT CACTCTAACACTTAT AA TTTAATTTT GTAGAGAGT AATAGAG
inhW[m2-i73]-t	CACACAACAACCACA TCT CAACAATTCACCTAC AA TTTAATTTT GTGTTTGAA AATAGAG
inhW[m2-i74]-t	CACACAACAACCACA TCT CACCCTACCACTCCC AA TTTAATTTT GATAATGAG AATAGAG
inhW[m2-i75]-t	CACACAACAACCACA TCT CACACTTCCAATTAT AA TTTAATTTT GTGGTGTTA AATAGAG
inhW[m2-i76]-t	CACACAACAACCACA TCT CACAATATCATCACC AA TTTAATTTT GATAGATGT AATAGAG
inhW[m2-i77]-t	CACACAACAACCACA TCT CAAACTCTCATCATT AA TTTAATTTT GATATGTTG AATAGAG
inhW[m2-i78]-t	CACACAACAACCACA TCT CAATTTCACACCACC AA TTTAATTTT GTAAGAGTT AATAGAG
inhW[m2-i79]-t	CACACAACAACCACA TCT CAACTTACCATTCAT AA TTTAATTTT GTGATGGAT AATAGAG
inhW[m2-i80]-t	CACACAACAACCACA TCT CATCCATACAACTAT AA TTTAATTTT GTTAAGAAG AATAGAG
inhW[m2-i81]-t	CACACAACAACCACA TCT CAACTCTACACATTC AA TTTAATTTT GATGGTTAT AATAGAG
inhW[m2-i82]-t	CACACAACAACCACA TCT CAAATCCTCATTTTC AA TTTAATTTT GAAATAGTG AATAGAG
inhW[m2-i83]-t	CACACAACAACCACA TCT CACCATTACAAACAT AA TTTAATTTT GAGAAGTAT AATAGAG
inhW[m2-i84]-t	CACACAACAACCACA TCT CATATATTCAACTCC AA TTTAATTTT GTAGGGTGA AATAGAG
inhW[m2-i85]-t	CACACAACAACCACA TCT CAATCACTCATTACA AA TTTAATTTT GTGGGTGTA AATAGAG
inhW[m2-i86]-t	CACACAACAACCACA TCT CACCAACACAAAATA AA TTTAATTTT GGAGATTGG AATAGAG
inhW[m2-i87]-t	CACACAACAACCACA TCT CAAACCCACATATAT AA TTTAATTTT GTTATTGTG AATAGAG
inhW[m2-i88]-t	CACACAACAACCACA TCT CATCTCTCCATAAAT AA TTTAATTTT GGAATTTTG AATAGAG
inhW[m2-i89]-t	CACACAACAACCACA TCT CATATCAACATCTCA AA TTTAATTTT GGGAGTATT AATAGAG
inhW[m2-i90]-t	CACACAACAACCACA TCT CACATCCACACTACA AA TTTAATTTT GAGGTTAGA AATAGAG
inhW[m2-i91]-t	CACACAACAACCACA TCT CATCAAACCACAACA AA TTTAATTTT GGTGAGTGA AATAGAG
inhW[m2-i92]-t	CACACAACAACCACA TCT CAAACTTTCACACAC AA TTTAATTTT GGTTGAGAA AATAGAG
inhW[m2-i93]-t	CACACAACAACCACA TCT CACCTACCCATCTAT AA TTTAATTTT GTATGTAAG AATAGAG
inhW[m2-i94]-t	CACACAACAACCACA TCT CACCTTTACAATTCC AA TTTAATTTT GTAGTAGAT AATAGAG
inhW[m2-i95]-t	CACACAACAACCACA TCT CATTCCTTCACATAT AA TTTAATTTT GATGTTTAG AATAGAG
inhW[m2-i96]-t	CACACAACAACCACA TCT CACAACTCCACCATT AA TTTAATTTT GGATTAAGT AATAGAG
inhW[m2-i97]-t	CACACAACAACCACA TCT CAATACTCCATCAAT AA TTTAATTTT GTGAGAAAA AATAGAG
inhW[m2-i98]-t	CACACAACAACCACA TCT CACTTATCCAACATT AA TTTAATTTT GAGTAAAGA AATAGAG
inhW[m2-i99]-t	CACACAACAACCACA TCT CACATAATCACCCTT AA TTTAATTTT GTGGATAAA AATAGAG
inhW[m2-i100]-t	CACACAACAACCACA TCT CATACCCTCACTAAC AA TTTAATTTT GATTTAGGA AATAGAG

Table S9: DNA sequences of bottom strands in the weight gates for memory 2.

Name	Sequence
inhW[m2-i1]-b	TT TTATCCTCC AAAATTAAA AAGTGTGGGAAGTTG AGA TG
inhW[m2-i2]-b	TT TCCAAATTC AAAATTAAA TAAGTTGGTTAGATG AGA TG
inhW[m2-i3]-b	TT ACACCATCC AAAATTAAA AGTATTGTTTAGGTG AGA TG
inhW[m2-i4]-b	TT CTCCACTCC AAAATTAAA GATTGTGATGTAATG AGA TG
inhW[m2-i5]-b	TT CCTTATTAC AAAATTAAA GATAATGTAAGGTTG AGA TG
inhW[m2-i6]-b	TT CCTCAAAAC AAAATTAAA TAAGATGATGGATTG AGA TG
inhW[m2-i7]-b	TT ATCTCTCTC AAAATTAAA GTGGATGTAGTGATG AGA TG
inhW[m2-i8]-b	TT CATTTCTTC AAAATTAAA AGGTATGTTGGGATG AGA TG
inhW[m2-i9]-b	TT AACCCTACC AAAATTAAA TGGGTTGGGAGATTG AGA TG
inhW[m2-i10]-b	TT TCATATCTC AAAATTAAA AAAGTTGTGGAAATG AGA TG
inhW[m2-i11]-b	TT CAATCATAC AAAATTAAA AGTGGTGAAAAGATG AGA TG
inhW[m2-i12]-b	TT CCATTTCCC AAAATTAAA GAGTTTGAGTAATTG AGA TG
inhW[m2-i13]-b	TT AACATAACC AAAATTAAA GAGGTTGTGATAGTG AGA TG
inhW[m2-i14]-b	TT CTTTAACAC AAAATTAAA GGGTGTGTTGTTGTG AGA TG
inhW[m2-i15]-b	TT CACCTATAC AAAATTAAA GGATATGTATGATTG AGA TG
inhW[m2-i16]-b	TT CCCATCAAC AAAATTAAA TAGTATGAAAGGGTG AGA TG
inhW[m2-i17]-b	TT ATTCTCATC AAAATTAAA GGGATTGAGTGGATG AGA TG
inhW[m2-i18]-b	TT TAATAACCC AAAATTAAA GATTATGTGAGTATG AGA TG
inhW[m2-i19]-b	TT TACTTATCC AAAATTAAA AGAAGTGGGTGTATG AGA TG
inhW[m2-i20]-b	TT ATTCAACTC AAAATTAAA AGATATGAGAAGGTG AGA TG
inhW[m2-i21]-b	TT CTCACTTCC AAAATTAAA AGGGTTGAAATAATG AGA TG
inhW[m2-i22]-b	TT TATCACATC AAAATTAAA GGAAGTGTATTAATG AGA TG
inhW[m2-i23]-b	TT TCTCATTCC AAAATTAAA ATGTGTGTGTAAATG AGA TG
inhW[m2-i24]-b	TT ATCACAAAC AAAATTAAA GAAGATGGAATTATG AGA TG
inhW[m2-i25]-b	TT ATTTCCAAC AAAATTAAA AGGAGTGAAGTGTTG AGA TG
inhW[m2-i26]-b	TT CATAAAACC AAAATTAAA GATAGTGGATAGATG AGA TG
inhW[m2-i27]-b	TT TTCCCTTTC AAAATTAAA TAATTTGTAGGAGTG AGA TG
inhW[m2-i28]-b	TT ACTATACAC AAAATTAAA GGGTATGAGGTAGTG AGA TG
inhW[m2-i29]-b	TT TCCTTTTAC AAAATTAAA TAGAGTGTAGGTATG AGA TG
inhW[m2-i30]-b	TT TATACCACC AAAATTAAA GAAGGTGTGTGAGTG AGA TG
inhW[m2-i31]-b	TT TTACCTAAC AAAATTAAA TAGAATGTGTAGTTG AGA TG
inhW[m2-i32]-b	TT TCAATATCC AAAATTAAA TGGTTTGTGAATATG AGA TG
inhW[m2-i33]-b	TT CTCCTCCAC AAAATTAAA AGTTTTGAATGTGTG AGA TG
inhW[m2-i34]-b	TT ACCCACAAC AAAATTAAA TAAAGTGGAAAAGTG AGA TG
inhW[m2-i35]-b	TT AACAACTTC AAAATTAAA GTGGGTGGGTATTTG AGA TG
inhW[m2-i36]-b	TT AAATACCTC AAAATTAAA AATAATGATGTGGTG AGA TG
inhW[m2-i37]-b	TT TCCCAATAC AAAATTAAA TATATTGAAGGGATG AGA TG
inhW[m2-i38]-b	TT CATCCTCTC AAAATTAAA GTTGTTGATGAATTG AGA TG
inhW[m2-i39]-b	TT TCTACTTAC AAAATTAAA AGGGATGTGTTTATG AGA TG
inhW[m2-i40]-b	TT AAAACCTAC AAAATTAAA GTTTGTGTATATGTG AGA TG
inhW[m2-i41]-b	TT CAACCACCC AAAATTAAA GAAAGTGATAGGATG AGA TG
inhW[m2-i42]-b	TT TATTCCCTC AAAATTAAA GTGATTGTGGTTTTG AGA TG
inhW[m2-i43]-b	TT ACCTTAAAC AAAATTAAA GGTGTTGTTATGATG AGA TG
inhW[m2-i44]-b	TT CCTATTTTC AAAATTAAA GTGAATGAATTTGTG AGA TG
inhW[m2-i45]-b	TT AACCAAATC AAAATTAAA AGTTATGGGATAATG AGA TG
inhW[m2-i46]-b	TT AATCTACCC AAAATTAAA ATGAGTGAGATATTG AGA TG
inhW[m2-i47]-b	TT TACCCAAAC AAAATTAAA TGAAATGGATTGGTG AGA TG
inhW[m2-i48]-b	TT CCTTCAATC AAAATTAAA GGGAATGTTAAAATG AGA TG
inhW[m2-i49]-b	TT TTAATTCCC AAAATTAAA TGTGTTGAGGAGGTG AGA TG
inhW[m2-i50]-b	TT ACAAAACTC AAAATTAAA GTTGGTGTAAGAATG AGA TG
inhW[m2-i51]-b	TT ATTAATCCC AAAATTAAA GTTAATGGGAATGTG AGA TG

Name	Sequence
inhW[m2-i52]-b	TT CCAACTCAC AAAATTAAA TAGTTTGGATGGTTG AGA TG
inhW[m2-i53]-b	TT TAAACACTC AAAATTAAA GTAAATGGTTGTTTG AGA TG
inhW[m2-i54]-b	TT CCCTCCTTC AAAATTAAA GGTATTGGAGGATTG AGA TG
inhW[m2-i55]-b	TT CTATATACC AAAATTAAA ATAGGTGGTGTGATG AGA TG
inhW[m2-i56]-b	TT CTACAATTC AAAATTAAA AGTGATGTTTGGTTG AGA TG
inhW[m2-i57]-b	TT CATATTCAC AAAATTAAA AGATTTGAGGTTATG AGA TG
inhW[m2-i58]-b	TT AAACTTTCC AAAATTAAA TGATGTGGAATATTG AGA TG
inhW[m2-i59]-b	TT CTCTTCACC AAAATTAAA GTTATTGGTATAGTG AGA TG
inhW[m2-i60]-b	TT ATATTACCC AAAATTAAA TAGATTGGGATTTTG AGA TG
inhW[m2-i61]-b	TT CATTACACC AAAATTAAA ATGTATGGTTTAGTG AGA TG
inhW[m2-i62]-b	TT ACTTTTACC AAAATTAAA GGATTTGGGTTGTTG AGA TG
inhW[m2-i63]-b	TT ACCATATTC AAAATTAAA TAGGGTGTTTTATTG AGA TG
inhW[m2-i64]-b	TT CCTCCCAAC AAAATTAAA TGTTTTGGTGATTTG AGA TG
inhW[m2-i65]-b	TT TCCACTACC AAAATTAAA GTGTTTGATAAGATG AGA TG
inhW[m2-i66]-b	TT TCCCTTATC AAAATTAAA GAGGATGGTGGTATG AGA TG
inhW[m2-i67]-b	TT TTTCCATTC AAAATTAAA TGAGGTGAGTTTGTG AGA TG
inhW[m2-i68]-b	TT ACACACTTC AAAATTAAA AAAATTGGTGGGTTG AGA TG
inhW[m2-i69]-b	TT CCACTAATC AAAATTAAA TAATATGGGTGAATG AGA TG
inhW[m2-i70]-b	TT AATTCACAC AAAATTAAA GTTTATGATTGGGTG AGA TG
inhW[m2-i71]-b	TT TTATTCCTC AAAATTAAA ATGGTTGTTGGTGTG AGA TG
inhW[m2-i72]-b	TT ACTCTCTAC AAAATTAAA ATAAGTGTTAGAGTG AGA TG
inhW[m2-i73]-b	TT TTCAAACAC AAAATTAAA GTAGGTGAATTGTTG AGA TG
inhW[m2-i74]-b	TT CTCATTATC AAAATTAAA GGGAGTGGTAGGGTG AGA TG
inhW[m2-i75]-b	TT TAACACCAC AAAATTAAA ATAATTGGAAGTGTG AGA TG
inhW[m2-i76]-b	TT ACATCTATC AAAATTAAA GGTGATGATATTGTG AGA TG
inhW[m2-i77]-b	TT CAACATATC AAAATTAAA AATGATGAGAGTTTG AGA TG
inhW[m2-i78]-b	TT AACTCTTAC AAAATTAAA GGTGGTGTGAAATTG AGA TG
inhW[m2-i79]-b	TT ATCCATCAC AAAATTAAA ATGAATGGTAAGTTG AGA TG
inhW[m2-i80]-b	TT CTTCTTAAC AAAATTAAA ATAGTTGTATGGATG AGA TG
inhW[m2-i81]-b	TT ATAACCATC AAAATTAAA GAATGTGTAGAGTTG AGA TG
inhW[m2-i82]-b	TT CACTATTTC AAAATTAAA GAAAATGAGGATTTG AGA TG
inhW[m2-i83]-b	TT ATACTTCTC AAAATTAAA ATGTTTGTAATGGTG AGA TG
inhW[m2-i84]-b	TT TCACCCTAC AAAATTAAA GGAGTTGAATATATG AGA TG
inhW[m2-i85]-b	TT TACACCCAC AAAATTAAA TGTAATGAGTGATTG AGA TG
inhW[m2-i86]-b	TT CCAATCTCC AAAATTAAA TATTTTGTGTTGGTG AGA TG
inhW[m2-i87]-b	TT CACAATAAC AAAATTAAA ATATATGTGGGTTTG AGA TG
inhW[m2-i88]-b	TT CAAAATTCC AAAATTAAA ATTTATGGAGAGATG AGA TG
inhW[m2-i89]-b	TT AATACTCCC AAAATTAAA TGAGATGTTGATATG AGA TG
inhW[m2-i90]-b	TT TCTAACCTC AAAATTAAA TGTAGTGTGGATGTG AGA TG
inhW[m2-i91]-b	TT TCACTCACC AAAATTAAA TGTTGTGGTTTGATG AGA TG
inhW[m2-i92]-b	TT TTCTCAACC AAAATTAAA GTGTGTGAAAGTTTG AGA TG
inhW[m2-i93]-b	TT CTTACATAC AAAATTAAA ATAGATGGGTAGGTG AGA TG
inhW[m2-i94]-b	TT ATCTACTAC AAAATTAAA GGAATTGTAAAGGTG AGA TG
inhW[m2-i95]-b	TT CTAAACATC AAAATTAAA ATATGTGAAGGAATG AGA TG
inhW[m2-i96]-b	TT ACTTAATCC AAAATTAAA AATGGTGGAGTTGTG AGA TG
inhW[m2-i97]-b	TT TTTTCTCAC AAAATTAAA ATTGATGGAGTATTG AGA TG
inhW[m2-i98]-b	TT TCTTTACTC AAAATTAAA AATGTTGGATAAGTG AGA TG
inhW[m2-i99]-b	TT TTTATCCAC AAAATTAAA AAGGGTGATTATGTG AGA TG
inhW[m2-i100]-b	TT TCCTAAATC AAAATTAAA GTTAGTGAGGGTATG AGA TG

Table S10: DNA sequences of top strands in the weight gates for reading out the learned weights in memory 1.

Name	Sequence
inhW[m1-i1]-Y1	CATAACACAATCACA TCT CAACTTCCCACACTT AA TTTAATTTT GGAGGATAA TGAAAGA
inhW[m1-i2]-Y1	CATAACACAATCACA TCT CATCTAACCAACTTA AA TTTAATTTT GAATTTGGA TGAAAGA
inhW[m1-i3]-Y1	CATAACACAATCACA TCT CACCTAAACAATACT AA TTTAATTTT GGATGGTGT TGAAAGA
inhW[m1-i4]-Y1	CATAACACAATCACA TCT CATTACATCACAATC AA TTTAATTTT GGAGTGGAG TGAAAGA
inhW[m1-i5]-Y1	CATAACACAATCACA TCT CAACCTTACATTATC AA TTTAATTTT GTAATAAGG TGAAAGA
inhW[m1-i6]-Y1	CATAACACAATCACA TCT CAATCCATCATCTTA AA TTTAATTTT GTTTTGAGG TGAAAGA
inhW[m1-i7]-Y1	CATAACACAATCACA TCT CATCACTACATCCAC AA TTTAATTTT GAGAGAGAT TGAAAGA
inhW[m1-i8]-Y1	CATAACACAATCACA TCT CATCCCAACATACCT AA TTTAATTTT GAAGAAATG TGAAAGA
inhW[m1-i9]-Y1	CATAACACAATCACA TCT CAATCTCCCAACCCA AA TTTAATTTT GGTAGGGTT TGAAAGA
inhW[m1-i10]-Y1	CATAACACAATCACA TCT CATTTCCACAACTTT AA TTTAATTTT GAGATATGA TGAAAGA
inhW[m1-i11]-Y1	CATAACACAATCACA TCT CATCTTTTCACCACT AA TTTAATTTT GTATGATTG TGAAAGA
inhW[m1-i12]-Y1	CATAACACAATCACA TCT CAATTACTCAAACTC AA TTTAATTTT GGGAAATGG TGAAAGA
inhW[m1-i13]-Y1	CATAACACAATCACA TCT CACTATCACAACCTC AA TTTAATTTT GGTTATGTT TGAAAGA
inhW[m1-i14]-Y1	CATAACACAATCACA TCT CACAACAACACACCC AA TTTAATTTT GTGTTAAAG TGAAAGA
inhW[m1-i15]-Y1	CATAACACAATCACA TCT CAATCATACATATCC AA TTTAATTTT GTATAGGTG TGAAAGA
inhW[m1-i16]-Y1	CATAACACAATCACA TCT CACCCTTTCATACTA AA TTTAATTTT GTTGATGGG TGAAAGA
inhW[m1-i17]-Y1	CATAACACAATCACA TCT CATCCACTCAATCCC AA TTTAATTTT GATGAGAAT TGAAAGA
inhW[m1-i18]-Y1	CATAACACAATCACA TCT CATACTCACATAATC AA TTTAATTTT GGGTTATTA TGAAAGA
inhW[m1-i19]-Y1	CATAACACAATCACA TCT CATACACCCACTTCT AA TTTAATTTT GGATAAGTA TGAAAGA
inhW[m1-i20]-Y1	CATAACACAATCACA TCT CACCTTCTCATATCT AA TTTAATTTT GAGTTGAAT TGAAAGA
inhW[m1-i21]-Y1	CATAACACAATCACA TCT CATTATTTCAACCCT AA TTTAATTTT GGAAGTGAG TGAAAGA
inhW[m1-i22]-Y1	CATAACACAATCACA TCT CATTAATACACTTCC AA TTTAATTTT GATGTGATA TGAAAGA
inhW[m1-i23]-Y1	CATAACACAATCACA TCT CATTTACACACACAT AA TTTAATTTT GGAATGAGA TGAAAGA
inhW[m1-i24]-Y1	CATAACACAATCACA TCT CATAATTCCATCTTC AA TTTAATTTT GTTTGTGAT TGAAAGA
inhW[m1-i25]-Y1	CATAACACAATCACA TCT CAACACTTCACTCCT AA TTTAATTTT GTTGGAAAT TGAAAGA
inhW[m1-i26]-Y1	CATAACACAATCACA TCT CATCTATCCACTATC AA TTTAATTTT GGTTTTATG TGAAAGA
inhW[m1-i27]-Y1	CATAACACAATCACA TCT CACTCCTACAAATTA AA TTTAATTTT GAAAGGGAA TGAAAGA
inhW[m1-i28]-Y1	CATAACACAATCACA TCT CACTACCTCATACCC AA TTTAATTTT GTGTATAGT TGAAAGA
inhW[m1-i29]-Y1	CATAACACAATCACA TCT CATACCTACACTCTA AA TTTAATTTT GTAAAAGGA TGAAAGA
inhW[m1-i30]-Y1	CATAACACAATCACA TCT CACTCACACACCTTC AA TTTAATTTT GGTGGTATA TGAAAGA
inhW[m1-i31]-Y1	CATAACACAATCACA TCT CAACTACACATTCTA AA TTTAATTTT GTTAGGTAA TGAAAGA
inhW[m1-i32]-Y1	CATAACACAATCACA TCT CATATTCACAAACCA AA TTTAATTTT GGATATTGA TGAAAGA
inhW[m1-i33]-Y1	CATAACACAATCACA TCT CACACATTCAAAACT AA TTTAATTTT GTGGAGGAG TGAAAGA
inhW[m1-i34]-Y1	CATAACACAATCACA TCT CACTTTTCCACTTTA AA TTTAATTTT GTTGTGGGT TGAAAGA
inhW[m1-i35]-Y1	CATAACACAATCACA TCT CAAATACCCACCCAC AA TTTAATTTT GAAGTTGTT TGAAAGA
inhW[m1-i36]-Y1	CATAACACAATCACA TCT CACCACATCATTATT AA TTTAATTTT GAGGTATTT TGAAAGA
inhW[m1-i37]-Y1	CATAACACAATCACA TCT CATCCCTTCAATATA AA TTTAATTTT GTATTGGGA TGAAAGA
inhW[m1-i38]-Y1	CATAACACAATCACA TCT CAATTCATCAACAAC AA TTTAATTTT GAGAGGATG TGAAAGA
inhW[m1-i39]-Y1	CATAACACAATCACA TCT CATAAACACATCCCT AA TTTAATTTT GTAAGTAGA TGAAAGA
inhW[m1-i40]-Y1	CATAACACAATCACA TCT CACATATACACAAAC AA TTTAATTTT GTAGGTTTT TGAAAGA
inhW[m1-i41]-Y1	CATAACACAATCACA TCT CATCCTATCACTTTC AA TTTAATTTT GGGTGGTTG TGAAAGA
inhW[m1-i42]-Y1	CATAACACAATCACA TCT CAAAACCACAATCAC AA TTTAATTTT GAGGGAATA TGAAAGA
inhW[m1-i43]-Y1	CATAACACAATCACA TCT CATCATAACAACACC AA TTTAATTTT GTTTAAGGT TGAAAGA
inhW[m1-i44]-Y1	CATAACACAATCACA TCT CACAAATTCATTCAC AA TTTAATTTT GAAAATAGG TGAAAGA
inhW[m1-i45]-Y1	CATAACACAATCACA TCT CATTATCCCATAACT AA TTTAATTTT GATTTGGTT TGAAAGA
inhW[m1-i46]-Y1	CATAACACAATCACA TCT CAATATCTCACTCAT AA TTTAATTTT GGGTAGATT TGAAAGA
inhW[m1-i47]-Y1	CATAACACAATCACA TCT CACCAATCCATTTCA AA TTTAATTTT GTTTGGGTA TGAAAGA
inhW[m1-i48]-Y1	CATAACACAATCACA TCT CATTTTAACATTCCC AA TTTAATTTT GATTGAAGG TGAAAGA
inhW[m1-i49]-Y1	CATAACACAATCACA TCT CACCTCCTCAACACA AA TTTAATTTT GGGAATTAA TGAAAGA
inhW[m1-i50]-Y1	CATAACACAATCACA TCT CATTCTTACACCAAC AA TTTAATTTT GAGTTTTGT TGAAAGA

Name	Sequence
inhW[m1-i51]-Y1	CATAACACAATCACA TCT CACATTCCCATTAAC AA TTTAATTTT GGGATTAAT TGAAAGA
inhW[m1-i52]-Y1	CATAACACAATCACA TCT CAACCATCCAAACTA AA TTTAATTTT GTGAGTTGG TGAAAGA
inhW[m1-i53]-Y1	CATAACACAATCACA TCT CAAACAACCATTTAC AA TTTAATTTT GAGTGTTTA TGAAAGA
inhW[m1-i54]-Y1	CATAACACAATCACA TCT CAATCCTCCAATACC AA TTTAATTTT GAAGGAGGG TGAAAGA
inhW[m1-i55]-Y1	CATAACACAATCACA TCT CATCACACCACCTAT AA TTTAATTTT GGTATATAG TGAAAGA
inhW[m1-i56]-Y1	CATAACACAATCACA TCT CAACCAAACATCACT AA TTTAATTTT GAATTGTAG TGAAAGA
inhW[m1-i57]-Y1	CATAACACAATCACA TCT CATAACCTCAAATCT AA TTTAATTTT GTGAATATG TGAAAGA
inhW[m1-i58]-Y1	CATAACACAATCACA TCT CAATATTCCACATCA AA TTTAATTTT GGAAAGTTT TGAAAGA
inhW[m1-i59]-Y1	CATAACACAATCACA TCT CACTATACCAATAAC AA TTTAATTTT GGTGAAGAG TGAAAGA
inhW[m1-i60]-Y1	CATAACACAATCACA TCT CAAAATCCCAATCTA AA TTTAATTTT GGGTAATAT TGAAAGA
inhW[m1-i61]-Y1	CATAACACAATCACA TCT CACTAAACCATACAT AA TTTAATTTT GGTGTAATG TGAAAGA
inhW[m1-i62]-Y1	CATAACACAATCACA TCT CAACAACCCAAATCC AA TTTAATTTT GGTAAAAGT TGAAAGA
inhW[m1-i63]-Y1	CATAACACAATCACA TCT CAATAAAACACCCTA AA TTTAATTTT GAATATGGT TGAAAGA
inhW[m1-i64]-Y1	CATAACACAATCACA TCT CAAATCACCAAAACA AA TTTAATTTT GTTGGGAGG TGAAAGA
inhW[m1-i65]-Y1	CATAACACAATCACA TCT CATCTTATCAAACAC AA TTTAATTTT GGTAGTGGA TGAAAGA
inhW[m1-i66]-Y1	CATAACACAATCACA TCT CATACCACCATCCTC AA TTTAATTTT GATAAGGGA TGAAAGA
inhW[m1-i67]-Y1	CATAACACAATCACA TCT CACAAACTCACCTCA AA TTTAATTTT GAATGGAAA TGAAAGA
inhW[m1-i68]-Y1	CATAACACAATCACA TCT CAACCCACCAATTTT AA TTTAATTTT GAAGTGTGT TGAAAGA
inhW[m1-i69]-Y1	CATAACACAATCACA TCT CATTCACCCATATTA AA TTTAATTTT GATTAGTGG TGAAAGA
inhW[m1-i70]-Y1	CATAACACAATCACA TCT CACCCAATCATAAAC AA TTTAATTTT GTGTGAATT TGAAAGA
inhW[m1-i71]-Y1	CATAACACAATCACA TCT CACACCAACAACCAT AA TTTAATTTT GAGGAATAA TGAAAGA
inhW[m1-i72]-Y1	CATAACACAATCACA TCT CACTCTAACACTTAT AA TTTAATTTT GTAGAGAGT TGAAAGA
inhW[m1-i73]-Y1	CATAACACAATCACA TCT CAACAATTCACCTAC AA TTTAATTTT GTGTTTGAA TGAAAGA
inhW[m1-i74]-Y1	CATAACACAATCACA TCT CACCCTACCACTCCC AA TTTAATTTT GATAATGAG TGAAAGA
inhW[m1-i75]-Y1	CATAACACAATCACA TCT CACACTTCCAATTAT AA TTTAATTTT GTGGTGTTA TGAAAGA
inhW[m1-i76]-Y1	CATAACACAATCACA TCT CACAATATCATCACC AA TTTAATTTT GATAGATGT TGAAAGA
inhW[m1-i77]-Y1	CATAACACAATCACA TCT CAAACTCTCATCATT AA TTTAATTTT GATATGTTG TGAAAGA
inhW[m1-i78]-Y1	CATAACACAATCACA TCT CAATTTCACACCACC AA TTTAATTTT GTAAGAGTT TGAAAGA
inhW[m1-i79]-Y1	CATAACACAATCACA TCT CAACTTACCATTCAT AA TTTAATTTT GTGATGGAT TGAAAGA
inhW[m1-i80]-Y1	CATAACACAATCACA TCT CATCCATACAACTAT AA TTTAATTTT GTTAAGAAG TGAAAGA
inhW[m1-i81]-Y1	CATAACACAATCACA TCT CAACTCTACACATTC AA TTTAATTTT GATGGTTAT TGAAAGA
inhW[m1-i82]-Y1	CATAACACAATCACA TCT CAAATCCTCATTTTC AA TTTAATTTT GAAATAGTG TGAAAGA
inhW[m1-i83]-Y1	CATAACACAATCACA TCT CACCATTACAAACAT AA TTTAATTTT GAGAAGTAT TGAAAGA
inhW[m1-i84]-Y1	CATAACACAATCACA TCT CATATATTCAACTCC AA TTTAATTTT GTAGGGTGA TGAAAGA
inhW[m1-i85]-Y1	CATAACACAATCACA TCT CAATCACTCATTACA AA TTTAATTTT GTGGGTGTA TGAAAGA
inhW[m1-i86]-Y1	CATAACACAATCACA TCT CACCAACACAAAATA AA TTTAATTTT GGAGATTGG TGAAAGA
inhW[m1-i87]-Y1	CATAACACAATCACA TCT CAAACCCACATATAT AA TTTAATTTT GTTATTGTG TGAAAGA
inhW[m1-i88]-Y1	CATAACACAATCACA TCT CATCTCTCCATAAAT AA TTTAATTTT GGAATTTTG TGAAAGA
inhW[m1-i89]-Y1	CATAACACAATCACA TCT CATATCAACATCTCA AA TTTAATTTT GGGAGTATT TGAAAGA
inhW[m1-i90]-Y1	CATAACACAATCACA TCT CACATCCACACTACA AA TTTAATTTT GAGGTTAGA TGAAAGA
inhW[m1-i91]-Y1	CATAACACAATCACA TCT CATCAAACCACAACA AA TTTAATTTT GGTGAGTGA TGAAAGA
inhW[m1-i92]-Y1	CATAACACAATCACA TCT CAAACTTTCACACAC AA TTTAATTTT GGTTGAGAA TGAAAGA
inhW[m1-i93]-Y1	CATAACACAATCACA TCT CACCTACCCATCTAT AA TTTAATTTT GTATGTAAG TGAAAGA
inhW[m1-i94]-Y1	CATAACACAATCACA TCT CACCTTTACAATTCC AA TTTAATTTT GTAGTAGAT TGAAAGA
inhW[m1-i95]-Y1	CATAACACAATCACA TCT CATTCCTTCACATAT AA TTTAATTTT GATGTTTAG TGAAAGA
inhW[m1-i96]-Y1	CATAACACAATCACA TCT CACAACTCCACCATT AA TTTAATTTT GGATTAAGT TGAAAGA
inhW[m1-i97]-Y1	CATAACACAATCACA TCT CAATACTCCATCAAT AA TTTAATTTT GTGAGAAAA TGAAAGA
inhW[m1-i98]-Y1	CATAACACAATCACA TCT CACTTATCCAACATT AA TTTAATTTT GAGTAAAGA TGAAAGA
inhW[m1-i99]-Y1	CATAACACAATCACA TCT CACATAATCACCCTT AA TTTAATTTT GTGGATAAA TGAAAGA
inhW[m1-i100]-Y1	CATAACACAATCACA TCT CATACCCTCACTAAC AA TTTAATTTT GATTTAGGA TGAAAGA

Table S11: DNA sequences of top strands in the weight gates for reading out the learned weights in memory 2.

Name	Sequence
inhW[m2-i1]-Y2	CAAATCTTCATCCCA TCT CAACTTCCCACACTT AA TTTAATTTT GGAGGATAA AATAGAG
inhW[m2-i2]-Y2	CAAATCTTCATCCCA TCT CATCTAACCAACTTA AA TTTAATTTT GAATTTGGA AATAGAG
inhW[m2-i3]-Y2	CAAATCTTCATCCCA TCT CACCTAAACAATACT AA TTTAATTTT GGATGGTGT AATAGAG
inhW[m2-i4]-Y2	CAAATCTTCATCCCA TCT CATTACATCACAATC AA TTTAATTTT GGAGTGGAG AATAGAG
inhW[m2-i5]-Y2	CAAATCTTCATCCCA TCT CAACCTTACATTATC AA TTTAATTTT GTAATAAGG AATAGAG
inhW[m2-i6]-Y2	CAAATCTTCATCCCA TCT CAATCCATCATCTTA AA TTTAATTTT GTTTTGAGG AATAGAG
inhW[m2-i7]-Y2	CAAATCTTCATCCCA TCT CATCACTACATCCAC AA TTTAATTTT GAGAGAGAT AATAGAG
inhW[m2-i8]-Y2	CAAATCTTCATCCCA TCT CATCCCAACATACCT AA TTTAATTTT GAAGAAATG AATAGAG
inhW[m2-i9]-Y2	CAAATCTTCATCCCA TCT CAATCTCCCAACCCA AA TTTAATTTT GGTAGGGTT AATAGAG
inhW[m2-i10]-Y2	CAAATCTTCATCCCA TCT CATTTCCACAACTTT AA TTTAATTTT GAGATATGA AATAGAG
inhW[m2-i11]-Y2	CAAATCTTCATCCCA TCT CATCTTTTCACCACT AA TTTAATTTT GTATGATTG AATAGAG
inhW[m2-i12]-Y2	CAAATCTTCATCCCA TCT CAATTACTCAAACTC AA TTTAATTTT GGGAAATGG AATAGAG
inhW[m2-i13]-Y2	CAAATCTTCATCCCA TCT CACTATCACAACCTC AA TTTAATTTT GGTTATGTT AATAGAG
inhW[m2-i14]-Y2	CAAATCTTCATCCCA TCT CACAACAACACACCC AA TTTAATTTT GTGTTAAAG AATAGAG
inhW[m2-i15]-Y2	CAAATCTTCATCCCA TCT CAATCATACATATCC AA TTTAATTTT GTATAGGTG AATAGAG
inhW[m2-i16]-Y2	CAAATCTTCATCCCA TCT CACCCTTTCATACTA AA TTTAATTTT GTTGATGGG AATAGAG
inhW[m2-i17]-Y2	CAAATCTTCATCCCA TCT CATCCACTCAATCCC AA TTTAATTTT GATGAGAAT AATAGAG
inhW[m2-i18]-Y2	CAAATCTTCATCCCA TCT CATACTCACATAATC AA TTTAATTTT GGGTTATTA AATAGAG
inhW[m2-i19]-Y2	CAAATCTTCATCCCA TCT CATACACCCACTTCT AA TTTAATTTT GGATAAGTA AATAGAG
inhW[m2-i20]-Y2	CAAATCTTCATCCCA TCT CACCTTCTCATATCT AA TTTAATTTT GAGTTGAAT AATAGAG
inhW[m2-i21]-Y2	CAAATCTTCATCCCA TCT CATTATTTCAACCCT AA TTTAATTTT GGAAGTGAG AATAGAG
inhW[m2-i22]-Y2	CAAATCTTCATCCCA TCT CATTAATACACTTCC AA TTTAATTTT GATGTGATA AATAGAG
inhW[m2-i23]-Y2	CAAATCTTCATCCCA TCT CATTTACACACACAT AA TTTAATTTT GGAATGAGA AATAGAG
inhW[m2-i24]-Y2	CAAATCTTCATCCCA TCT CATAATTCCATCTTC AA TTTAATTTT GTTTGTGAT AATAGAG
inhW[m2-i25]-Y2	CAAATCTTCATCCCA TCT CAACACTTCACTCCT AA TTTAATTTT GTTGGAAAT AATAGAG
inhW[m2-i26]-Y2	CAAATCTTCATCCCA TCT CATCTATCCACTATC AA TTTAATTTT GGTTTTATG AATAGAG
inhW[m2-i27]-Y2	CAAATCTTCATCCCA TCT CACTCCTACAAATTA AA TTTAATTTT GAAAGGGAA AATAGAG
inhW[m2-i28]-Y2	CAAATCTTCATCCCA TCT CACTACCTCATACCC AA TTTAATTTT GTGTATAGT AATAGAG
inhW[m2-i29]-Y2	CAAATCTTCATCCCA TCT CATACCTACACTCTA AA TTTAATTTT GTAAAAGGA AATAGAG
inhW[m2-i30]-Y2	CAAATCTTCATCCCA TCT CACTCACACACCTTC AA TTTAATTTT GGTGGTATA AATAGAG
inhW[m2-i31]-Y2	CAAATCTTCATCCCA TCT CAACTACACATTCTA AA TTTAATTTT GTTAGGTAA AATAGAG
inhW[m2-i32]-Y2	CAAATCTTCATCCCA TCT CATATTCACAAACCA AA TTTAATTTT GGATATTGA AATAGAG
inhW[m2-i33]-Y2	CAAATCTTCATCCCA TCT CACACATTCAAAACT AA TTTAATTTT GTGGAGGAG AATAGAG
inhW[m2-i34]-Y2	CAAATCTTCATCCCA TCT CACTTTTCCACTTTA AA TTTAATTTT GTTGTGGGT AATAGAG
inhW[m2-i35]-Y2	CAAATCTTCATCCCA TCT CAAATACCCACCCAC AA TTTAATTTT GAAGTTGTT AATAGAG
inhW[m2-i36]-Y2	CAAATCTTCATCCCA TCT CACCACATCATTATT AA TTTAATTTT GAGGTATTT AATAGAG
inhW[m2-i37]-Y2	CAAATCTTCATCCCA TCT CATCCCTTCAATATA AA TTTAATTTT GTATTGGGA AATAGAG
inhW[m2-i38]-Y2	CAAATCTTCATCCCA TCT CAATTCATCAACAAC AA TTTAATTTT GAGAGGATG AATAGAG
inhW[m2-i39]-Y2	CAAATCTTCATCCCA TCT CATAAACACATCCCT AA TTTAATTTT GTAAGTAGA AATAGAG
inhW[m2-i40]-Y2	CAAATCTTCATCCCA TCT CACATATACACAAAC AA TTTAATTTT GTAGGTTTT AATAGAG
inhW[m2-i41]-Y2	CAAATCTTCATCCCA TCT CATCCTATCACTTTC AA TTTAATTTT GGGTGGTTG AATAGAG
inhW[m2-i42]-Y2	CAAATCTTCATCCCA TCT CAAAACCACAATCAC AA TTTAATTTT GAGGGAATA AATAGAG
inhW[m2-i43]-Y2	CAAATCTTCATCCCA TCT CATCATAACAACACC AA TTTAATTTT GTTTAAGGT AATAGAG
inhW[m2-i44]-Y2	CAAATCTTCATCCCA TCT CACAAATTCATTCAC AA TTTAATTTT GAAAATAGG AATAGAG
inhW[m2-i45]-Y2	CAAATCTTCATCCCA TCT CATTATCCCATAACT AA TTTAATTTT GATTTGGTT AATAGAG
inhW[m2-i46]-Y2	CAAATCTTCATCCCA TCT CAATATCTCACTCAT AA TTTAATTTT GGGTAGATT AATAGAG
inhW[m2-i47]-Y2	CAAATCTTCATCCCA TCT CACCAATCCATTTCA AA TTTAATTTT GTTTGGGTA AATAGAG
inhW[m2-i48]-Y2	CAAATCTTCATCCCA TCT CATTTTAACATTCCC AA TTTAATTTT GATTGAAGG AATAGAG
inhW[m2-i49]-Y2	CAAATCTTCATCCCA TCT CACCTCCTCAACACA AA TTTAATTTT GGGAATTAA AATAGAG
inhW[m2-i50]-Y2	CAAATCTTCATCCCA TCT CATTCTTACACCAAC AA TTTAATTTT GAGTTTTGT AATAGAG

Name	Sequence
inhW[m2-i51]-Y2	CAAATCTTCATCCCA TCT CACATTCCCATTAAC AA TTTAATTTT GGGATTAAT AATAGAG
inhW[m2-i52]-Y2	CAAATCTTCATCCCA TCT CAACCATCCAAACTA AA TTTAATTTT GTGAGTTGG AATAGAG
inhW[m2-i53]-Y2	CAAATCTTCATCCCA TCT CAAACAACCATTTAC AA TTTAATTTT GAGTGTTTA AATAGAG
inhW[m2-i54]-Y2	CAAATCTTCATCCCA TCT CAATCCTCCAATACC AA TTTAATTTT GAAGGAGGG AATAGAG
inhW[m2-i55]-Y2	CAAATCTTCATCCCA TCT CATCACACCACCTAT AA TTTAATTTT GGTATATAG AATAGAG
inhW[m2-i56]-Y2	CAAATCTTCATCCCA TCT CAACCAAACATCACT AA TTTAATTTT GAATTGTAG AATAGAG
inhW[m2-i57]-Y2	CAAATCTTCATCCCA TCT CATAACCTCAAATCT AA TTTAATTTT GTGAATATG AATAGAG
inhW[m2-i58]-Y2	CAAATCTTCATCCCA TCT CAATATTCCACATCA AA TTTAATTTT GGAAAGTTT AATAGAG
inhW[m2-i59]-Y2	CAAATCTTCATCCCA TCT CACTATACCAATAAC AA TTTAATTTT GGTGAAGAG AATAGAG
inhW[m2-i60]-Y2	CAAATCTTCATCCCA TCT CAAAATCCCAATCTA AA TTTAATTTT GGGTAATAT AATAGAG
inhW[m2-i61]-Y2	CAAATCTTCATCCCA TCT CACTAAACCATACAT AA TTTAATTTT GGTGTAATG AATAGAG
inhW[m2-i62]-Y2	CAAATCTTCATCCCA TCT CAACAACCCAAATCC AA TTTAATTTT GGTAAAAGT AATAGAG
inhW[m2-i63]-Y2	CAAATCTTCATCCCA TCT CAATAAAACACCCTA AA TTTAATTTT GAATATGGT AATAGAG
inhW[m2-i64]-Y2	CAAATCTTCATCCCA TCT CAAATCACCAAAACA AA TTTAATTTT GTTGGGAGG AATAGAG
inhW[m2-i65]-Y2	CAAATCTTCATCCCA TCT CATCTTATCAAACAC AA TTTAATTTT GGTAGTGGA AATAGAG
inhW[m2-i66]-Y2	CAAATCTTCATCCCA TCT CATACCACCATCCTC AA TTTAATTTT GATAAGGGA AATAGAG
inhW[m2-i67]-Y2	CAAATCTTCATCCCA TCT CACAAACTCACCTCA AA TTTAATTTT GAATGGAAA AATAGAG
inhW[m2-i68]-Y2	CAAATCTTCATCCCA TCT CAACCCACCAATTTT AA TTTAATTTT GAAGTGTGT AATAGAG
inhW[m2-i69]-Y2	CAAATCTTCATCCCA TCT CATTCACCCATATTA AA TTTAATTTT GATTAGTGG AATAGAG
inhW[m2-i70]-Y2	CAAATCTTCATCCCA TCT CACCCAATCATAAAC AA TTTAATTTT GTGTGAATT AATAGAG
inhW[m2-i71]-Y2	CAAATCTTCATCCCA TCT CACACCAACAACCAT AA TTTAATTTT GAGGAATAA AATAGAG
inhW[m2-i72]-Y2	CAAATCTTCATCCCA TCT CACTCTAACACTTAT AA TTTAATTTT GTAGAGAGT AATAGAG
inhW[m2-i73]-Y2	CAAATCTTCATCCCA TCT CAACAATTCACCTAC AA TTTAATTTT GTGTTTGAA AATAGAG
inhW[m2-i74]-Y2	CAAATCTTCATCCCA TCT CACCCTACCACTCCC AA TTTAATTTT GATAATGAG AATAGAG
inhW[m2-i75]-Y2	CAAATCTTCATCCCA TCT CACACTTCCAATTAT AA TTTAATTTT GTGGTGTTA AATAGAG
inhW[m2-i76]-Y2	CAAATCTTCATCCCA TCT CACAATATCATCACC AA TTTAATTTT GATAGATGT AATAGAG
inhW[m2-i77]-Y2	CAAATCTTCATCCCA TCT CAAACTCTCATCATT AA TTTAATTTT GATATGTTG AATAGAG
inhW[m2-i78]-Y2	CAAATCTTCATCCCA TCT CAATTTCACACCACC AA TTTAATTTT GTAAGAGTT AATAGAG
inhW[m2-i79]-Y2	CAAATCTTCATCCCA TCT CAACTTACCATTCAT AA TTTAATTTT GTGATGGAT AATAGAG
inhW[m2-i80]-Y2	CAAATCTTCATCCCA TCT CATCCATACAACTAT AA TTTAATTTT GTTAAGAAG AATAGAG
inhW[m2-i81]-Y2	CAAATCTTCATCCCA TCT CAACTCTACACATTC AA TTTAATTTT GATGGTTAT AATAGAG
inhW[m2-i82]-Y2	CAAATCTTCATCCCA TCT CAAATCCTCATTTTC AA TTTAATTTT GAAATAGTG AATAGAG
inhW[m2-i83]-Y2	CAAATCTTCATCCCA TCT CACCATTACAAACAT AA TTTAATTTT GAGAAGTAT AATAGAG
inhW[m2-i84]-Y2	CAAATCTTCATCCCA TCT CATATATTCAACTCC AA TTTAATTTT GTAGGGTGA AATAGAG
inhW[m2-i85]-Y2	CAAATCTTCATCCCA TCT CAATCACTCATTACA AA TTTAATTTT GTGGGTGTA AATAGAG
inhW[m2-i86]-Y2	CAAATCTTCATCCCA TCT CACCAACACAAAATA AA TTTAATTTT GGAGATTGG AATAGAG
inhW[m2-i87]-Y2	CAAATCTTCATCCCA TCT CAAACCCACATATAT AA TTTAATTTT GTTATTGTG AATAGAG
inhW[m2-i88]-Y2	CAAATCTTCATCCCA TCT CATCTCTCCATAAAT AA TTTAATTTT GGAATTTTG AATAGAG
inhW[m2-i89]-Y2	CAAATCTTCATCCCA TCT CATATCAACATCTCA AA TTTAATTTT GGGAGTATT AATAGAG
inhW[m2-i90]-Y2	CAAATCTTCATCCCA TCT CACATCCACACTACA AA TTTAATTTT GAGGTTAGA AATAGAG
inhW[m2-i91]-Y2	CAAATCTTCATCCCA TCT CATCAAACCACAACA AA TTTAATTTT GGTGAGTGA AATAGAG
inhW[m2-i92]-Y2	CAAATCTTCATCCCA TCT CAAACTTTCACACAC AA TTTAATTTT GGTTGAGAA AATAGAG
inhW[m2-i93]-Y2	CAAATCTTCATCCCA TCT CACCTACCCATCTAT AA TTTAATTTT GTATGTAAG AATAGAG
inhW[m2-i94]-Y2	CAAATCTTCATCCCA TCT CACCTTTACAATTCC AA TTTAATTTT GTAGTAGAT AATAGAG
inhW[m2-i95]-Y2	CAAATCTTCATCCCA TCT CATTCCTTCACATAT AA TTTAATTTT GATGTTTAG AATAGAG
inhW[m2-i96]-Y2	CAAATCTTCATCCCA TCT CACAACTCCACCATT AA TTTAATTTT GGATTAAGT AATAGAG
inhW[m2-i97]-Y2	CAAATCTTCATCCCA TCT CAATACTCCATCAAT AA TTTAATTTT GTGAGAAAA AATAGAG
inhW[m2-i98]-Y2	CAAATCTTCATCCCA TCT CACTTATCCAACATT AA TTTAATTTT GAGTAAAGA AATAGAG
inhW[m2-i99]-Y2	CAAATCTTCATCCCA TCT CACATAATCACCCTT AA TTTAATTTT GTGGATAAA AATAGAG
inhW[m2-i100]-Y2	CAAATCTTCATCCCA TCT CATACCCTCACTAAC AA TTTAATTTT GATTTAGGA AATAGAG

Table S12: DNA sequences of activator strands for testing the activatable weights in memory 1.

Name	Sequence
Act[m1-i1]	TCTTTCA TTATCCTCC AAAATTAAA TT
Act[m1-i2]	TCTTTCA TCCAAATTC AAAATTAAA TT
Act[m1-i3]	TCTTTCA ACACCATCC AAAATTAAA TT
Act[m1-i4]	TCTTTCA CTCCACTCC AAAATTAAA TT
Act[m1-i5]	TCTTTCA CCTTATTAC AAAATTAAA TT
Act[m1-i6]	TCTTTCA CCTCAAAAC AAAATTAAA TT
Act[m1-i7]	TCTTTCA ATCTCTCTC AAAATTAAA TT
Act[m1-i8]	TCTTTCA CATTTCTTC AAAATTAAA TT
Act[m1-i9]	TCTTTCA AACCCTACC AAAATTAAA TT
Act[m1-i10]	TCTTTCA TCATATCTC AAAATTAAA TT
Act[m1-i11]	TCTTTCA CAATCATAC AAAATTAAA TT
Act[m1-i12]	TCTTTCA CCATTTCCC AAAATTAAA TT
Act[m1-i13]	TCTTTCA AACATAACC AAAATTAAA TT
Act[m1-i14]	TCTTTCA CTTTAACAC AAAATTAAA TT
Act[m1-i15]	TCTTTCA CACCTATAC AAAATTAAA TT
Act[m1-i16]	TCTTTCA CCCATCAAC AAAATTAAA TT
Act[m1-i17]	TCTTTCA ATTCTCATC AAAATTAAA TT
Act[m1-i18]	TCTTTCA TAATAACCC AAAATTAAA TT
Act[m1-i19]	TCTTTCA TACTTATCC AAAATTAAA TT
Act[m1-i20]	TCTTTCA ATTCAACTC AAAATTAAA TT
Act[m1-i21]	TCTTTCA CTCACTTCC AAAATTAAA TT
Act[m1-i22]	TCTTTCA TATCACATC AAAATTAAA TT
Act[m1-i23]	TCTTTCA TCTCATTCC AAAATTAAA TT
Act[m1-i24]	TCTTTCA ATCACAAAC AAAATTAAA TT
Act[m1-i25]	TCTTTCA ATTTCCAAC AAAATTAAA TT
Act[m1-i26]	TCTTTCA CATAAAACC AAAATTAAA TT
Act[m1-i27]	TCTTTCA TTCCCTTTC AAAATTAAA TT
Act[m1-i28]	TCTTTCA ACTATACAC AAAATTAAA TT
Act[m1-i29]	TCTTTCA TCCTTTTAC AAAATTAAA TT
Act[m1-i30]	TCTTTCA TATACCACC AAAATTAAA TT
Act[m1-i31]	TCTTTCA TTACCTAAC AAAATTAAA TT
Act[m1-i32]	TCTTTCA TCAATATCC AAAATTAAA TT
Act[m1-i33]	TCTTTCA CTCCTCCAC AAAATTAAA TT
Act[m1-i34]	TCTTTCA ACCCACAAC AAAATTAAA TT
Act[m1-i35]	TCTTTCA AACAACTTC AAAATTAAA TT
Act[m1-i36]	TCTTTCA AAATACCTC AAAATTAAA TT
Act[m1-i37]	TCTTTCA TCCCAATAC AAAATTAAA TT
Act[m1-i38]	TCTTTCA CATCCTCTC AAAATTAAA TT
Act[m1-i39]	TCTTTCA TCTACTTAC AAAATTAAA TT
Act[m1-i40]	TCTTTCA AAAACCTAC AAAATTAAA TT
Act[m1-i41]	TCTTTCA CAACCACCC AAAATTAAA TT
Act[m1-i42]	TCTTTCA TATTCCCTC AAAATTAAA TT
Act[m1-i43]	TCTTTCA ACCTTAAAC AAAATTAAA TT
Act[m1-i44]	TCTTTCA CCTATTTTC AAAATTAAA TT
Act[m1-i45]	TCTTTCA AACCAAATC AAAATTAAA TT
Act[m1-i46]	TCTTTCA AATCTACCC AAAATTAAA TT
Act[m1-i47]	TCTTTCA TACCCAAAC AAAATTAAA TT
Act[m1-i48]	TCTTTCA CCTTCAATC AAAATTAAA TT
Act[m1-i49]	TCTTTCA TTAATTCCC AAAATTAAA TT
Act[m1-i50]	TCTTTCA ACAAAACTC AAAATTAAA TT
Act[m1-i51]	TCTTTCA ATTAATCCC AAAATTAAA TT

Name	Sequence
Act[m1-i52]	TCTTTCA CCAACTCAC AAAATTAAA TT
Act[m1-i53]	TCTTTCA TAAACACTC AAAATTAAA TT
Act[m1-i54]	TCTTTCA CCCTCCTTC AAAATTAAA TT
Act[m1-i55]	TCTTTCA CTATATACC AAAATTAAA TT
Act[m1-i56]	TCTTTCA CTACAATTC AAAATTAAA TT
Act[m1-i57]	TCTTTCA CATATTCAC AAAATTAAA TT
Act[m1-i58]	TCTTTCA AAACTTTCC AAAATTAAA TT
Act[m1-i59]	TCTTTCA CTCTTCACC AAAATTAAA TT
Act[m1-i60]	TCTTTCA ATATTACCC AAAATTAAA TT
Act[m1-i61]	TCTTTCA CATTACACC AAAATTAAA TT
Act[m1-i62]	TCTTTCA ACTTTTACC AAAATTAAA TT
Act[m1-i63]	TCTTTCA ACCATATTC AAAATTAAA TT
Act[m1-i64]	TCTTTCA CCTCCCAAC AAAATTAAA TT
Act[m1-i65]	TCTTTCA TCCACTACC AAAATTAAA TT
Act[m1-i66]	TCTTTCA TCCCTTATC AAAATTAAA TT
Act[m1-i67]	TCTTTCA TTTCCATTC AAAATTAAA TT
Act[m1-i68]	TCTTTCA ACACACTTC AAAATTAAA TT
Act[m1-i69]	TCTTTCA CCACTAATC AAAATTAAA TT
Act[m1-i70]	TCTTTCA AATTCACAC AAAATTAAA TT
Act[m1-i71]	TCTTTCA TTATTCCTC AAAATTAAA TT
Act[m1-i72]	TCTTTCA ACTCTCTAC AAAATTAAA TT
Act[m1-i73]	TCTTTCA TTCAAACAC AAAATTAAA TT
Act[m1-i74]	TCTTTCA CTCATTATC AAAATTAAA TT
Act[m1-i75]	TCTTTCA TAACACCAC AAAATTAAA TT
Act[m1-i76]	TCTTTCA ACATCTATC AAAATTAAA TT
Act[m1-i77]	TCTTTCA CAACATATC AAAATTAAA TT
Act[m1-i78]	TCTTTCA AACTCTTAC AAAATTAAA TT
Act[m1-i79]	TCTTTCA ATCCATCAC AAAATTAAA TT
Act[m1-i80]	TCTTTCA CTTCTTAAC AAAATTAAA TT
Act[m1-i81]	TCTTTCA ATAACCATC AAAATTAAA TT
Act[m1-i82]	TCTTTCA CACTATTTC AAAATTAAA TT
Act[m1-i83]	TCTTTCA ATACTTCTC AAAATTAAA TT
Act[m1-i84]	TCTTTCA TCACCCTAC AAAATTAAA TT
Act[m1-i85]	TCTTTCA TACACCCAC AAAATTAAA TT
Act[m1-i86]	TCTTTCA CCAATCTCC AAAATTAAA TT
Act[m1-i87]	TCTTTCA CACAATAAC AAAATTAAA TT
Act[m1-i88]	TCTTTCA CAAAATTCC AAAATTAAA TT
Act[m1-i89]	TCTTTCA AATACTCCC AAAATTAAA TT
Act[m1-i90]	TCTTTCA TCTAACCTC AAAATTAAA TT
Act[m1-i91]	TCTTTCA TCACTCACC AAAATTAAA TT
Act[m1-i92]	TCTTTCA TTCTCAACC AAAATTAAA TT
Act[m1-i93]	TCTTTCA CTTACATAC AAAATTAAA TT
Act[m1-i94]	TCTTTCA ATCTACTAC AAAATTAAA TT
Act[m1-i95]	TCTTTCA CTAAACATC AAAATTAAA TT
Act[m1-i96]	TCTTTCA ACTTAATCC AAAATTAAA TT
Act[m1-i97]	TCTTTCA TTTTCTCAC AAAATTAAA TT
Act[m1-i98]	TCTTTCA TCTTTACTC AAAATTAAA TT
Act[m1-i99]	TCTTTCA TTTATCCAC AAAATTAAA TT
Act[m1-i100]	TCTTTCA TCCTAAATC AAAATTAAA TT

Table S13: DNA sequences of activator strands for testing the activatable weights in memory 2.

Name	Sequence
Act[m2-i1]	CTCTATT TTATCCTCC AAAATTAAA TT
Act[m2-i2]	CTCTATT TCCAAATTC AAAATTAAA TT
Act[m2-i3]	CTCTATT ACACCATCC AAAATTAAA TT
Act[m2-i4]	CTCTATT CTCCACTCC AAAATTAAA TT
Act[m2-i5]	CTCTATT CCTTATTAC AAAATTAAA TT
Act[m2-i6]	CTCTATT CCTCAAAAC AAAATTAAA TT
Act[m2-i7]	CTCTATT ATCTCTCTC AAAATTAAA TT
Act[m2-i8]	CTCTATT CATTTCTTC AAAATTAAA TT
Act[m2-i9]	CTCTATT AACCCTACC AAAATTAAA TT
Act[m2-i10]	CTCTATT TCATATCTC AAAATTAAA TT
Act[m2-i11]	CTCTATT CAATCATAC AAAATTAAA TT
Act[m2-i12]	CTCTATT CCATTTCCC AAAATTAAA TT
Act[m2-i13]	CTCTATT AACATAACC AAAATTAAA TT
Act[m2-i14]	CTCTATT CTTTAACAC AAAATTAAA TT
Act[m2-i15]	CTCTATT CACCTATAC AAAATTAAA TT
Act[m2-i16]	CTCTATT CCCATCAAC AAAATTAAA TT
Act[m2-i17]	CTCTATT ATTCTCATC AAAATTAAA TT
Act[m2-i18]	CTCTATT TAATAACCC AAAATTAAA TT
Act[m2-i19]	CTCTATT TACTTATCC AAAATTAAA TT
Act[m2-i20]	CTCTATT ATTCAACTC AAAATTAAA TT
Act[m2-i21]	CTCTATT CTCACTTCC AAAATTAAA TT
Act[m2-i22]	CTCTATT TATCACATC AAAATTAAA TT
Act[m2-i23]	CTCTATT TCTCATTCC AAAATTAAA TT
Act[m2-i24]	CTCTATT ATCACAAAC AAAATTAAA TT
Act[m2-i25]	CTCTATT ATTTCCAAC AAAATTAAA TT
Act[m2-i26]	CTCTATT CATAAAACC AAAATTAAA TT
Act[m2-i27]	CTCTATT TTCCCTTTC AAAATTAAA TT
Act[m2-i28]	CTCTATT ACTATACAC AAAATTAAA TT
Act[m2-i29]	CTCTATT TCCTTTTAC AAAATTAAA TT
Act[m2-i30]	CTCTATT TATACCACC AAAATTAAA TT
Act[m2-i31]	CTCTATT TTACCTAAC AAAATTAAA TT
Act[m2-i32]	CTCTATT TCAATATCC AAAATTAAA TT
Act[m2-i33]	CTCTATT CTCCTCCAC AAAATTAAA TT
Act[m2-i34]	CTCTATT ACCCACAAC AAAATTAAA TT
Act[m2-i35]	CTCTATT AACAACTTC AAAATTAAA TT
Act[m2-i36]	CTCTATT AAATACCTC AAAATTAAA TT
Act[m2-i37]	CTCTATT TCCCAATAC AAAATTAAA TT
Act[m2-i38]	CTCTATT CATCCTCTC AAAATTAAA TT
Act[m2-i39]	CTCTATT TCTACTTAC AAAATTAAA TT
Act[m2-i40]	CTCTATT AAAACCTAC AAAATTAAA TT
Act[m2-i41]	CTCTATT CAACCACCC AAAATTAAA TT
Act[m2-i42]	CTCTATT TATTCCCTC AAAATTAAA TT
Act[m2-i43]	CTCTATT ACCTTAAAC AAAATTAAA TT
Act[m2-i44]	CTCTATT CCTATTTTC AAAATTAAA TT
Act[m2-i45]	CTCTATT AACCAAATC AAAATTAAA TT
Act[m2-i46]	CTCTATT AATCTACCC AAAATTAAA TT
Act[m2-i47]	CTCTATT TACCCAAAC AAAATTAAA TT
Act[m2-i48]	CTCTATT CCTTCAATC AAAATTAAA TT
Act[m2-i49]	CTCTATT TTAATTCCC AAAATTAAA TT
Act[m2-i50]	CTCTATT ACAAAACTC AAAATTAAA TT
Act[m2-i51]	CTCTATT ATTAATCCC AAAATTAAA TT

Name	Sequence
Act[m2-i52]	CTCTATT CCAACTCAC AAAATTAAA TT
Act[m2-i53]	CTCTATT TAAACACTC AAAATTAAA TT
Act[m2-i54]	CTCTATT CCCTCCTTC AAAATTAAA TT
Act[m2-i55]	CTCTATT CTATATACC AAAATTAAA TT
Act[m2-i56]	CTCTATT CTACAATTC AAAATTAAA TT
Act[m2-i57]	CTCTATT CATATTCAC AAAATTAAA TT
Act[m2-i58]	CTCTATT AAACTTTCC AAAATTAAA TT
Act[m2-i59]	CTCTATT CTCTTCACC AAAATTAAA TT
Act[m2-i60]	CTCTATT ATATTACCC AAAATTAAA TT
Act[m2-i61]	CTCTATT CATTACACC AAAATTAAA TT
Act[m2-i62]	CTCTATT ACTTTTACC AAAATTAAA TT
Act[m2-i63]	CTCTATT ACCATATTC AAAATTAAA TT
Act[m2-i64]	CTCTATT CCTCCCAAC AAAATTAAA TT
Act[m2-i65]	CTCTATT TCCACTACC AAAATTAAA TT
Act[m2-i66]	CTCTATT TCCCTTATC AAAATTAAA TT
Act[m2-i67]	CTCTATT TTTCCATTC AAAATTAAA TT
Act[m2-i68]	CTCTATT ACACACTTC AAAATTAAA TT
Act[m2-i69]	CTCTATT CCACTAATC AAAATTAAA TT
Act[m2-i70]	CTCTATT AATTCACAC AAAATTAAA TT
Act[m2-i71]	CTCTATT TTATTCCTC AAAATTAAA TT
Act[m2-i72]	CTCTATT ACTCTCTAC AAAATTAAA TT
Act[m2-i73]	CTCTATT TTCAAACAC AAAATTAAA TT
Act[m2-i74]	CTCTATT CTCATTATC AAAATTAAA TT
Act[m2-i75]	CTCTATT TAACACCAC AAAATTAAA TT
Act[m2-i76]	CTCTATT ACATCTATC AAAATTAAA TT
Act[m2-i77]	CTCTATT CAACATATC AAAATTAAA TT
Act[m2-i78]	CTCTATT AACTCTTAC AAAATTAAA TT
Act[m2-i79]	CTCTATT ATCCATCAC AAAATTAAA TT
Act[m2-i80]	CTCTATT CTTCTTAAC AAAATTAAA TT
Act[m2-i81]	CTCTATT ATAACCATC AAAATTAAA TT
Act[m2-i82]	CTCTATT CACTATTTC AAAATTAAA TT
Act[m2-i83]	CTCTATT ATACTTCTC AAAATTAAA TT
Act[m2-i84]	CTCTATT TCACCCTAC AAAATTAAA TT
Act[m2-i85]	CTCTATT TACACCCAC AAAATTAAA TT
Act[m2-i86]	CTCTATT CCAATCTCC AAAATTAAA TT
Act[m2-i87]	CTCTATT CACAATAAC AAAATTAAA TT
Act[m2-i88]	CTCTATT CAAAATTCC AAAATTAAA TT
Act[m2-i89]	CTCTATT AATACTCCC AAAATTAAA TT
Act[m2-i90]	CTCTATT TCTAACCTC AAAATTAAA TT
Act[m2-i91]	CTCTATT TCACTCACC AAAATTAAA TT
Act[m2-i92]	CTCTATT TTCTCAACC AAAATTAAA TT
Act[m2-i93]	CTCTATT CTTACATAC AAAATTAAA TT
Act[m2-i94]	CTCTATT ATCTACTAC AAAATTAAA TT
Act[m2-i95]	CTCTATT CTAAACATC AAAATTAAA TT
Act[m2-i96]	CTCTATT ACTTAATCC AAAATTAAA TT
Act[m2-i97]	CTCTATT TTTTCTCAC AAAATTAAA TT
Act[m2-i98]	CTCTATT TCTTTACTC AAAATTAAA TT
Act[m2-i99]	CTCTATT TTTATCCAC AAAATTAAA TT
Act[m2-i100]	CTCTATT TCCTAAATC AAAATTAAA TT

Table S14: DNA sequences of top strands in the thresholds.

Name	Sequence
Th[i1]-t	CAACTTCCCACACTT
Th[i2]-t	CATCTAACCAACTTA
Th[i3]-t	CACCTAAACAATACT
Th[i4]-t	CATTACATCACAATC
Th[i5]-t	CAACCTTACATTATC
Th[i6]-t	CAATCCATCATCTTA
Th[i7]-t	CATCACTACATCCAC
Th[i8]-t	CATCCCAACATACCT
Th[i9]-t	CAATCTCCCAACCCA
Th[i10]-t	CATTTCCACAACTTT
Th[i11]-t	CATCTTTTCACCACT
Th[i12]-t	CAATTACTCAAACTC
Th[i13]-t	CACTATCACAACCTC
Th[i14]-t	CACAACAACACACCC
Th[i15]-t	CAATCATACATATCC
Th[i16]-t	CACCCTTTCATACTA
Th[i17]-t	CATCCACTCAATCCC
Th[i18]-t	CATACTCACATAATC
Th[i19]-t	CATACACCCACTTCT
Th[i20]-t	CACCTTCTCATATCT
Th[i21]-t	CATTATTTCAACCCT
Th[i22]-t	CATTAATACACTTCC
Th[i23]-t	CATTTACACACACAT
Th[i24]-t	CATAATTCCATCTTC
Th[i25]-t	CAACACTTCACTCCT
Th[i26]-t	CATCTATCCACTATC
Th[i27]-t	CACTCCTACAAATTA
Th[i28]-t	CACTACCTCATACCC
Th[i29]-t	CATACCTACACTCTA
Th[i30]-t	CACTCACACACCTTC
Th[i31]-t	CAACTACACATTCTA
Th[i32]-t	CATATTCACAAACCA
Th[i33]-t	CACACATTCAAAACT
Th[i34]-t	CACTTTTCCACTTTA
Th[i35]-t	CAAATACCCACCCAC
Th[i36]-t	CACCACATCATTATT
Th[i37]-t	CATCCCTTCAATATA
Th[i38]-t	CAATTCATCAACAAC
Th[i39]-t	CATAAACACATCCCT
Th[i40]-t	CACATATACACAAAC
Th[i41]-t	CATCCTATCACTTTC
Th[i42]-t	CAAAACCACAATCAC
Th[i43]-t	CATCATAACAACACC
Th[i44]-t	CACAAATTCATTCAC
Th[i45]-t	CATTATCCCATAACT
Th[i46]-t	CAATATCTCACTCAT
Th[i47]-t	CACCAATCCATTTCA
Th[i48]-t	CATTTTAACATTCCC
Th[i49]-t	CACCTCCTCAACACA
Th[i50]-t	CATTCTTACACCAAC
Th[i51]-t	CACATTCCCATTAAC

Name	Sequence
Th[i52]-t	CAACCATCCAAACTA
Th[i53]-t	CAAACAACCATTTAC
Th[i54]-t	CAATCCTCCAATACC
Th[i55]-t	CATCACACCACCTAT
Th[i56]-t	CAACCAAACATCACT
Th[i57]-t	CATAACCTCAAATCT
Th[i58]-t	CAATATTCCACATCA
Th[i59]-t	CACTATACCAATAAC
Th[i60]-t	CAAAATCCCAATCTA
Th[i61]-t	CACTAAACCATACAT
Th[i62]-t	CAACAACCCAAATCC
Th[i63]-t	CAATAAAACACCCTA
Th[i64]-t	CAAATCACCAAAACA
Th[i65]-t	CATCTTATCAAACAC
Th[i66]-t	CATACCACCATCCTC
Th[i67]-t	CACAAACTCACCTCA
Th[i68]-t	CAACCCACCAATTTT
Th[i69]-t	CATTCACCCATATTA
Th[i70]-t	CACCCAATCATAAAC
Th[i71]-t	CACACCAACAACCAT
Th[i72]-t	CACTCTAACACTTAT
Th[i73]-t	CAACAATTCACCTAC
Th[i74]-t	CACCCTACCACTCCC
Th[i75]-t	CACACTTCCAATTAT
Th[i76]-t	CACAATATCATCACC
Th[i77]-t	CAAACTCTCATCATT
Th[i78]-t	CAATTTCACACCACC
Th[i79]-t	CAACTTACCATTCAT
Th[i80]-t	CATCCATACAACTAT
Th[i81]-t	CAACTCTACACATTC
Th[i82]-t	CAAATCCTCATTTTC
Th[i83]-t	CACCATTACAAACAT
Th[i84]-t	CATATATTCAACTCC
Th[i85]-t	CAATCACTCATTACA
Th[i86]-t	CACCAACACAAAATA
Th[i87]-t	CAAACCCACATATAT
Th[i88]-t	CATCTCTCCATAAAT
Th[i89]-t	CATATCAACATCTCA
Th[i90]-t	CACATCCACACTACA
Th[i91]-t	CATCAAACCACAACA
Th[i92]-t	CAAACTTTCACACAC
Th[i93]-t	CACCTACCCATCTAT
Th[i94]-t	CACCTTTACAATTCC
Th[i95]-t	CATTCCTTCACATAT
Th[i96]-t	CACAACTCCACCATT
Th[i97]-t	CAATACTCCATCAAT
Th[i98]-t	CACTTATCCAACATT
Th[i99]-t	CACATAATCACCCTT
Th[i100]-t	CATACCCTCACTAAC

Table S15: DNA sequences of bottom strands in the thresholds.

Name	Sequence
Th[i1]-b	AAAATTAAA AAGTGTGGGAAGTTG
Th[i2]-b	AAAATTAAA TAAGTTGGTTAGATG
Th[i3]-b	AAAATTAAA AGTATTGTTTAGGTG
Th[i4]-b	AAAATTAAA GATTGTGATGTAATG
Th[i5]-b	AAAATTAAA GATAATGTAAGGTTG
Th[i6]-b	AAAATTAAA TAAGATGATGGATTG
Th[i7]-b	AAAATTAAA GTGGATGTAGTGATG
Th[i8]-b	AAAATTAAA AGGTATGTTGGGATG
Th[i9]-b	AAAATTAAA TGGGTTGGGAGATTG
Th[i10]-b	AAAATTAAA AAAGTTGTGGAAATG
Th[i11]-b	AAAATTAAA AGTGGTGAAAAGATG
Th[i12]-b	AAAATTAAA GAGTTTGAGTAATTG
Th[i13]-b	AAAATTAAA GAGGTTGTGATAGTG
Th[i14]-b	AAAATTAAA GGGTGTGTTGTTGTG
Th[i15]-b	AAAATTAAA GGATATGTATGATTG
Th[i16]-b	AAAATTAAA TAGTATGAAAGGGTG
Th[i17]-b	AAAATTAAA GGGATTGAGTGGATG
Th[i18]-b	AAAATTAAA GATTATGTGAGTATG
Th[i19]-b	AAAATTAAA AGAAGTGGGTGTATG
Th[i20]-b	AAAATTAAA AGATATGAGAAGGTG
Th[i21]-b	AAAATTAAA AGGGTTGAAATAATG
Th[i22]-b	AAAATTAAA GGAAGTGTATTAATG
Th[i23]-b	AAAATTAAA ATGTGTGTGTAAATG
Th[i24]-b	AAAATTAAA GAAGATGGAATTATG
Th[i25]-b	AAAATTAAA AGGAGTGAAGTGTTG
Th[i26]-b	AAAATTAAA GATAGTGGATAGATG
Th[i27]-b	AAAATTAAA TAATTTGTAGGAGTG
Th[i28]-b	AAAATTAAA GGGTATGAGGTAGTG
Th[i29]-b	AAAATTAAA TAGAGTGTAGGTATG
Th[i30]-b	AAAATTAAA GAAGGTGTGTGAGTG
Th[i31]-b	AAAATTAAA TAGAATGTGTAGTTG
Th[i32]-b	AAAATTAAA TGGTTTGTGAATATG
Th[i33]-b	AAAATTAAA AGTTTTGAATGTGTG
Th[i34]-b	AAAATTAAA TAAAGTGGAAAAGTG
Th[i35]-b	AAAATTAAA GTGGGTGGGTATTTG
Th[i36]-b	AAAATTAAA AATAATGATGTGGTG
Th[i37]-b	AAAATTAAA TATATTGAAGGGATG
Th[i38]-b	AAAATTAAA GTTGTTGATGAATTG
Th[i39]-b	AAAATTAAA AGGGATGTGTTTATG
Th[i40]-b	AAAATTAAA GTTTGTGTATATGTG
Th[i41]-b	AAAATTAAA GAAAGTGATAGGATG
Th[i42]-b	AAAATTAAA GTGATTGTGGTTTTG
Th[i43]-b	AAAATTAAA GGTGTTGTTATGATG
Th[i44]-b	AAAATTAAA GTGAATGAATTTGTG
Th[i45]-b	AAAATTAAA AGTTATGGGATAATG
Th[i46]-b	AAAATTAAA ATGAGTGAGATATTG
Th[i47]-b	AAAATTAAA TGAAATGGATTGGTG
Th[i48]-b	AAAATTAAA GGGAATGTTAAAATG
Th[i49]-b	AAAATTAAA TGTGTTGAGGAGGTG
Th[i50]-b	AAAATTAAA GTTGGTGTAAGAATG
Th[i51]-b	AAAATTAAA GTTAATGGGAATGTG

Name	Sequence
Th[i52]-b	AAAATTAAA TAGTTTGGATGGTTG
Th[i53]-b	AAAATTAAA GTAAATGGTTGTTTG
Th[i54]-b	AAAATTAAA GGTATTGGAGGATTG
Th[i55]-b	AAAATTAAA ATAGGTGGTGTGATG
Th[i56]-b	AAAATTAAA AGTGATGTTTGGTTG
Th[i57]-b	AAAATTAAA AGATTTGAGGTTATG
Th[i58]-b	AAAATTAAA TGATGTGGAATATTG
Th[i59]-b	AAAATTAAA GTTATTGGTATAGTG
Th[i60]-b	AAAATTAAA TAGATTGGGATTTTG
Th[i61]-b	AAAATTAAA ATGTATGGTTTAGTG
Th[i62]-b	AAAATTAAA GGATTTGGGTTGTTG
Th[i63]-b	AAAATTAAA TAGGGTGTTTTATTG
Th[i64]-b	AAAATTAAA TGTTTTGGTGATTTG
Th[i65]-b	AAAATTAAA GTGTTTGATAAGATG
Th[i66]-b	AAAATTAAA GAGGATGGTGGTATG
Th[i67]-b	AAAATTAAA TGAGGTGAGTTTGTG
Th[i68]-b	AAAATTAAA AAAATTGGTGGGTTG
Th[i69]-b	AAAATTAAA TAATATGGGTGAATG
Th[i70]-b	AAAATTAAA GTTTATGATTGGGTG
Th[i71]-b	AAAATTAAA ATGGTTGTTGGTGTG
Th[i72]-b	AAAATTAAA ATAAGTGTTAGAGTG
Th[i73]-b	AAAATTAAA GTAGGTGAATTGTTG
Th[i74]-b	AAAATTAAA GGGAGTGGTAGGGTG
Th[i75]-b	AAAATTAAA ATAATTGGAAGTGTG
Th[i76]-b	AAAATTAAA GGTGATGATATTGTG
Th[i77]-b	AAAATTAAA AATGATGAGAGTTTG
Th[i78]-b	AAAATTAAA GGTGGTGTGAAATTG
Th[i79]-b	AAAATTAAA ATGAATGGTAAGTTG
Th[i80]-b	AAAATTAAA ATAGTTGTATGGATG
Th[i81]-b	AAAATTAAA GAATGTGTAGAGTTG
Th[i82]-b	AAAATTAAA GAAAATGAGGATTTG
Th[i83]-b	AAAATTAAA ATGTTTGTAATGGTG
Th[i84]-b	AAAATTAAA GGAGTTGAATATATG
Th[i85]-b	AAAATTAAA TGTAATGAGTGATTG
Th[i86]-b	AAAATTAAA TATTTTGTGTTGGTG
Th[i87]-b	AAAATTAAA ATATATGTGGGTTTG
Th[i88]-b	AAAATTAAA ATTTATGGAGAGATG
Th[i89]-b	AAAATTAAA TGAGATGTTGATATG
Th[i90]-b	AAAATTAAA TGTAGTGTGGATGTG
Th[i91]-b	AAAATTAAA TGTTGTGGTTTGATG
Th[i92]-b	AAAATTAAA GTGTGTGAAAGTTTG
Th[i93]-b	AAAATTAAA ATAGATGGGTAGGTG
Th[i94]-b	AAAATTAAA GGAATTGTAAAGGTG
Th[i95]-b	AAAATTAAA ATATGTGAAGGAATG
Th[i96]-b	AAAATTAAA AATGGTGGAGTTGTG
Th[i97]-b	AAAATTAAA ATTGATGGAGTATTG
Th[i98]-b	AAAATTAAA AATGTTGGATAAGTG
Th[i99]-b	AAAATTAAA AAGGGTGATTATGTG
Th[i100]-b	AAAATTAAA GTTAGTGAGGGTATG

Table S16: DNA sequences of weight fuels.

Name	Sequence
XF[i1]	CA TCT CAACTTCCCACACTT
XF[i2]	CA TCT CATCTAACCAACTTA
XF[i3]	CA TCT CACCTAAACAATACT
XF[i4]	CA TCT CATTACATCACAATC
XF[i5]	CA TCT CAACCTTACATTATC
XF[i6]	CA TCT CAATCCATCATCTTA
XF[i7]	CA TCT CATCACTACATCCAC
XF[i8]	CA TCT CATCCCAACATACCT
XF[i9]	CA TCT CAATCTCCCAACCCA
XF[i10]	CA TCT CATTTCCACAACTTT
XF[i11]	CA TCT CATCTTTTCACCACT
XF[i12]	CA TCT CAATTACTCAAACTC
XF[i13]	CA TCT CACTATCACAACCTC
XF[i14]	CA TCT CACAACAACACACCC
XF[i15]	CA TCT CAATCATACATATCC
XF[i16]	CA TCT CACCCTTTCATACTA
XF[i17]	CA TCT CATCCACTCAATCCC
XF[i18]	CA TCT CATACTCACATAATC
XF[i19]	CA TCT CATACACCCACTTCT
XF[i20]	CA TCT CACCTTCTCATATCT
XF[i21]	CA TCT CATTATTTCAACCCT
XF[i22]	CA TCT CATTAATACACTTCC
XF[i23]	CA TCT CATTTACACACACAT
XF[i24]	CA TCT CATAATTCCATCTTC
XF[i25]	CA TCT CAACACTTCACTCCT
XF[i26]	CA TCT CATCTATCCACTATC
XF[i27]	CA TCT CACTCCTACAAATTA
XF[i28]	CA TCT CACTACCTCATACCC
XF[i29]	CA TCT CATACCTACACTCTA
XF[i30]	CA TCT CACTCACACACCTTC
XF[i31]	CA TCT CAACTACACATTCTA
XF[i32]	CA TCT CATATTCACAAACCA
XF[i33]	CA TCT CACACATTCAAAACT
XF[i34]	CA TCT CACTTTTCCACTTTA
XF[i35]	CA TCT CAAATACCCACCCAC
XF[i36]	CA TCT CACCACATCATTATT
XF[i37]	CA TCT CATCCCTTCAATATA
XF[i38]	CA TCT CAATTCATCAACAAC
XF[i39]	CA TCT CATAAACACATCCCT
XF[i40]	CA TCT CACATATACACAAAC
XF[i41]	CA TCT CATCCTATCACTTTC
XF[i42]	CA TCT CAAAACCACAATCAC
XF[i43]	CA TCT CATCATAACAACACC
XF[i44]	CA TCT CACAAATTCATTCAC
XF[i45]	CA TCT CATTATCCCATAACT
XF[i46]	CA TCT CAATATCTCACTCAT
XF[i47]	CA TCT CACCAATCCATTTCA
XF[i48]	CA TCT CATTTTAACATTCCC
XF[i49]	CA TCT CACCTCCTCAACACA
XF[i50]	CA TCT CATTCTTACACCAAC
XF[i51]	CA TCT CACATTCCCATTAAC

Name	Sequence
XF[i52]	CA TCT CAACCATCCAAACTA
XF[i53]	CA TCT CAAACAACCATTTAC
XF[i54]	CA TCT CAATCCTCCAATACC
XF[i55]	CA TCT CATCACACCACCTAT
XF[i56]	CA TCT CAACCAAACATCACT
XF[i57]	CA TCT CATAACCTCAAATCT
XF[i58]	CA TCT CAATATTCCACATCA
XF[i59]	CA TCT CACTATACCAATAAC
XF[i60]	CA TCT CAAAATCCCAATCTA
XF[i61]	CA TCT CACTAAACCATACAT
XF[i62]	CA TCT CAACAACCCAAATCC
XF[i63]	CA TCT CAATAAAACACCCTA
XF[i64]	CA TCT CAAATCACCAAAACA
XF[i65]	CA TCT CATCTTATCAAACAC
XF[i66]	CA TCT CATACCACCATCCTC
XF[i67]	CA TCT CACAAACTCACCTCA
XF[i68]	CA TCT CAACCCACCAATTTT
XF[i69]	CA TCT CATTCACCCATATTA
XF[i70]	CA TCT CACCCAATCATAAAC
XF[i71]	CA TCT CACACCAACAACCAT
XF[i72]	CA TCT CACTCTAACACTTAT
XF[i73]	CA TCT CAACAATTCACCTAC
XF[i74]	CA TCT CACCCTACCACTCCC
XF[i75]	CA TCT CACACTTCCAATTAT
XF[i76]	CA TCT CACAATATCATCACC
XF[i77]	CA TCT CAAACTCTCATCATT
XF[i78]	CA TCT CAATTTCACACCACC
XF[i79]	CA TCT CAACTTACCATTCAT
XF[i80]	CA TCT CATCCATACAACTAT
XF[i81]	CA TCT CAACTCTACACATTC
XF[i82]	CA TCT CAAATCCTCATTTTC
XF[i83]	CA TCT CACCATTACAAACAT
XF[i84]	CA TCT CATATATTCAACTCC
XF[i85]	CA TCT CAATCACTCATTACA
XF[i86]	CA TCT CACCAACACAAAATA
XF[i87]	CA TCT CAAACCCACATATAT
XF[i88]	CA TCT CATCTCTCCATAAAT
XF[i89]	CA TCT CATATCAACATCTCA
XF[i90]	CA TCT CACATCCACACTACA
XF[i91]	CA TCT CATCAAACCACAACA
XF[i92]	CA TCT CAAACTTTCACACAC
XF[i93]	CA TCT CACCTACCCATCTAT
XF[i94]	CA TCT CACCTTTACAATTCC
XF[i95]	CA TCT CATTCCTTCACATAT
XF[i96]	CA TCT CACAACTCCACCATT
XF[i97]	CA TCT CAATACTCCATCAAT
XF[i98]	CA TCT CACTTATCCAACATT
XF[i99]	CA TCT CACATAATCACCCTT
XF[i100]	CA TCT CATACCCTCACTAAC

Table S17: DNA sequences of label and inhibitor strands.

Name	Sequence
L1	GAAGGTT GG AAAATTAAA
L2	GTAGTGA GT AAAATTAAA
Inh1	TTTAATTTT CC AACCTTC
Inh2	TTTAATTTT AC TCACTAC

Table S18: DNA sequences of strands in the summation, annihilation, and signal restoration layers.

Name	Sequence
S1	CACTTCATAAATCCA TCT CACACTATAATTCCA
S2	CAACATATCAATTCA TCT CACACAACAACCACA
SG1-b	TG AGA TGGAATTATAGTGTG AGA TG
SG2-b	TG AGA TGTGGTTGTTGTGTG AGA TG
Anh12-t	TG TG AGA TGAATTGATATGTTG CACTTCATAAATCCA
Anh12-b	TG TG AGA TGGATTTATGAAGTG CAACATATCAATTCA
Y1	CATAACACAATCACA TCT CACTTCATAAATCCA
Y2	CAAATCTTCATCCCA TCT CAACATATCAATTCA
RG1-b	TG AGA TGGATTTATGAAGTG AGA TG
RG1-b	TG AGA TGAATTGATATGTTG AGA TG
YF1	CA TCT CACTTCATAAATCCA
YF2	CA TCT CAACATATCAATTCA

Table S19: DNA sequences of fluorophore and quencher modified strands.

Name	Sequence
Rep1-t-RQ	/5IAbRQ/ CATAACACAATCACA
Rep1-b-ATTO590	TG AGA TGTGATTGTGTTATG /3ATTO590N/
Rep2-t-FQ	/5IABkFQ/ CAAATCTTCATCCCA
Rep2-b-ATTO488	TG AGA TGGGATGAAGATTTG /3ATTO488N/
RepP1-t-RQ	/5IAbRQ/ CACACTATAATTCCA
RepP1-b-ATTO590	TG AGA TGGAATTATAGTGTG /3ATTO590N/
RepP2-t-RQ	/5IAbRQ/ CACACAACAACCACA
RepP2-b-ATTO647	TG AGA TGTGGTTGTTGTGTG /3ATTO647NN/
X[i1]-RQ	CAACTTCCCACACTT TTTAATTTT /3IAbRQSp/
X[i3]-FQ	CACCTAAACAATACT TTTAATTTT /3IABkFQ/
X[i5]-RQ	CAACCTTACATTATC TTTAATTTT /3IAbRQSp/
X[i7]-RQ	CATCACTACATCCAC TTTAATTTT /3IAbRQSp/
inhAct[m1-i1]-b-ATTO550	/5ATTO550N/ GG AAAATTAAA AAGTGTGGGAAGTTG TCTTTCA TTATCCTCC AAAATTAAA TT
inhAct[m1-i3]-b-ATTO488	/5ATTO488N/ GG AAAATTAAA AGTATTGTTTAGGTG TCTTTCA ACACCATCC AAAATTAAA TT
inhAct[m1-i5]-b-ATTO590	/5ATTO590N/ GG AAAATTAAA GATAATGTAAGGTTG TCTTTCA CCTTATTAC AAAATTAAA TT
inhAct[m1-i7]-b-ATTO647	/5ATTO647NN/ GG AAAATTAAA GTGGATGTAGTGATG TCTTTCA ATCTCTCTC AAAATTAAA TT
inhAct[m2-i1]-b-ATTO647	/5ATTO647NN/ GT AAAATTAAA AAGTGTGGGAAGTTG CTCTATT TTATCCTCC AAAATTAAA TT
inhAct[m2-i3]-b-ATTO488	/5ATTO488N/ GT AAAATTAAA AGTATTGTTTAGGTG CTCTATT ACACCATCC AAAATTAAA TT
inhAct[m2-i5]-b-ATTO550	/5ATTO550N/ GT AAAATTAAA GATAATGTAAGGTTG CTCTATT CCTTATTAC AAAATTAAA TT
inhAct[m2-i7]-b-ATTO590	/5ATTO590N/ GT AAAATTAAA GTGGATGTAGTGATG CTCTATT ATCTCTCTC AAAATTAAA TT

References

[1] Evans, C. G. & Winfree, E. DNA sticky end design and assignment for robust algorithmic self-assembly. In *DNA Computing and Molecular Programming: 19th International Conference, DNA 19, Tempe, AZ, USA, September 22-27, 2013. Proceedings 19*, 61–75 (Springer, 2013).

[2] Qian, L. & Winfree, E. Scaling up digital circuit computation with DNA strand displacement cascades. *Science* **332**, 1196–1201 (2011).

[3] Cherry, K. M. & Qian, L. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* **559**, 370–376 (2018).

[4] SantaLucia Jr, J. & Hicks, D. The thermodynamics of DNA structural motifs. *Annual Review of Biophysics* **33**, 415–440 (2004).

[5] Cardelli, L. Two-domain DNA strand displacement. *Mathematical Structures in Computer Science* **23**, 247–271 (2013).

[6] Chen, Y.-J. *et al.* Programmable chemical controllers made from DNA. *Nature Nanotechnology* **8**, 755–762 (2013).

[7] Zhang, D. Y. Cooperative hybridization of oligonucleotides. *Journal of the American Chemical Society* **133**, 1077–1086 (2011).

[8] Taylor, D. N., Davidson, S. R. & Qian, L. A cooperative DNA catalyst. *Journal of the American Chemical Society* **143**, 15567–15571 (2021).

[9] Yang, X., Tang, Y., Traynor, S. M. & Li, F. Regulation of DNA strand displacement using an allosteric DNA toehold. *Journal of the American Chemical Society* **138**, 14076–14082 (2016).

[10] Johnson, H. A. & Condon, A. A coupled reconfiguration mechanism for single-stranded DNA strand displacement systems. In *28th International Conference on DNA Computing and Molecular Programming (DNA 28)* (Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022).

[11] Zhang, D. Y. & Winfree, E. Robustness and modularity properties of a non-covalent DNA catalytic reaction. *Nucleic Acids Research* **38**, 4182–4197 (2010).

[12] Braich, R. S., Chelyapov, N., Johnson, C., Rothemund, P. W. & Adleman, L. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science* **296**, 499–502 (2002).

[13] Zhang, D. Y. & Winfree, E. Control of DNA strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society* **131**, 17303–17314 (2009).

[14] Genot, A. J., Zhang, D. Y., Bath, J. & Turberfield, A. J. Remote toehold: a mechanism for flexible control of DNA hybridization kinetics. *Journal of the American Chemical Society* **133**, 2177–2182 (2011).

[15] Badelt, S. *et al.* A domain-level DNA strand displacement reaction enumerator allowing arbitrary non-pseudoknotted secondary structures. *Journal of the Royal Society Interface* **17**, 20190866 (2020).

[16] Badelt, S. *et al.* A general-purpose CRN-to-DSD compiler with formal verification, optimization, and simulation capabilities. In *DNA Computing and Molecular Programming: 23rd International Conference, DNA 23, Austin, TX, USA, September 24–28, 2017, Proceedings 23*, 232–248 (Springer, 2017).

[17] Wang, J. S., Yan, Y. H. & Zhang, D. Y. Modular probes for enriching and detecting complex nucleic acid sequences. *Nature Chemistry* **9**, 1222–1228 (2017).

[18] Zhang, J. X. *et al.* Predicting DNA hybridization kinetics from sequence. *Nature Chemistry* **10**, 91–98 (2018).

[19] Machinek, R. R., Ouldridge, T. E., Haley, N. E., Bath, J. & Turberfield, A. J. Programmable energy landscapes for kinetic control of DNA strand displacement. *Nature Communications* **5**, 5324 (2014).

[20] Haley, N. E. *et al.* Design of hidden thermodynamic driving for non-equilibrium systems via mismatch elimination during DNA strand displacement. *Nature Communications* **11**, 2562 (2020).

[21] Zimmers, Z. A., Adams, N. M., Gabella, W. E. & Haselton, F. R. Fluorophore-quencher interactions effect on hybridization characteristics of complementary oligonucleotides. *Analytical Methods* **11**, 2862–2867 (2019).

[22] Yakovchuk, P., Protozanova, E. & Frank-Kamenetskii, M. D. Base-stacking and base-pairing contributions into thermal stability of the DNA double helix. *Nucleic Acids Research* **34**, 564–574 (2006).

[23] Irmisch, P., Ouldridge, T. E. & Seidel, R. Modeling DNA-strand displacement reactions in the presence of base-pair mismatches. *Journal of the American Chemical Society* **142**, 11451–11463 (2020).

[24] Huguet, J. M. *et al.* Single-molecule derivation of salt dependent base-pair free energies in DNA. *Proceedings of the National Academy of Sciences* **107**, 15431–15436 (2010).

[25] Fornace, M. E. *et al.* NUPACK: analysis and design of nucleic acid structures, devices, and systems. *ChemRxiv* (2022).

[26] Which type of oligo purification should I choose? `https://www.idtdna.com/pages/education/decoded/article/which-type-of-purification-should-i-choose`. Accessed: 2024-11-24.

[27] Seelig, G., Soloveichik, D., Zhang, D. Y. & Winfree, E. Enzyme-free nucleic acid logic circuits. *Science* **314**, 1585–1588 (2006).

[28] Wang, B., Thachuk, C. & Soloveichik, D. Speed and correctness guarantees for programmable enthalpy-neutral DNA reactions. *ACS Synthetic Biology* **12**, 993–1006 (2023).

[29] Doyle, A. C. *The adventures of Sherlock Holmes* (Wordsworth Editions, 1992).