# Contrastive Hebbian Learning for Multicomponent Liquids

**Yancheng Du**
California Institute of Technology, Pasadena, CA, USA

**Cameron Chalk**
California Institute of Technology, Pasadena, CA, USA

**Salvador Buse**
California Institute of Technology, Pasadena, CA, USA

**Lulu Qian**
California Institute of Technology, Pasadena, CA, USA

**Erik Winfree**
California Institute of Technology, Pasadena, CA, USA

## Abstract

Molecules with designed interactions can serve as substrates for information processing; demonstrated examples include algorithmic self-assembly of DNA tiles and DNA strand displacement reactions. These two well-established paradigms correspond to solid-phase-like behavior, where spatially structured molecular assemblies grow by crystalline attachment, and gas-phase-like behavior, where a dilute mixture of freely diffusing molecules is governed by mass-action kinetics of reactions. A third paradigm, liquid-liquid phase separation, is a fundamental phenomenon in physical and biological systems, giving rise to membraneless compartments that facilitate complex information processing. Recent studies have shown that multicomponent liquid mixtures described by lattice models are analogous to Boltzmann machines and can implement neural computation. Mean-field models have also suggested a connection to Hopfield networks and illustrated complex decision-making during condensation from a reservoir. As an alternative to previous approaches to train liquid interaction energies using backpropagation through a loss function, here we derive "local" learning rules based on contrastive Hebbian learning that may be more biologically and physically plausible. The trained systems demonstrate a range of computational tasks, including associative recall, linear and nonlinear input-output relations, pattern classification, and spatial phase separation. Our work suggests that the computational potential of liquid-phase molecular systems could be unlocked in real physical systems with local learning rules.

**2012 ACM Subject Classification** Hardware → Neural systems; Applied computing → Physics; Applied computing → Systems biology

**Keywords and phrases** multicomponent liquid, liquid-liquid phase separation, continuous Hopfield network, contrastive Hebbian learning

**Digital Object Identifier** 10.4230/LIPIcs.DNA.32.13

**Supplementary Material** *Software (Source Code, Notebooks, and Supplementary Information)*: `https://github.com/YanchengDu/LiquidCHL`
archived at `swh:1:dir:b89e452e1236f62498ae55537e3dd62972e6d523`

**Funding** *Yancheng Du*: Schmidt Sciences Polymath Award to LQ.
*Cameron Chalk*: NSF CCF/FET 2212546.
*Salvador Buse*: Open Philanthropy Good Ventures Foundation, NSF CCF/FET Award 2212546.
*Lulu Qian*: Schmidt Sciences Polymath Award, NSF CCF/FET 2212546.
*Erik Winfree*: NSF CCF/FET 2212546.

**Acknowledgements** The authors want to thank Inhoo Lee, Sandra Temgoua, Daichi Hayakawa, Arvind Murugan, Krishna Shrinivas, and others for helpful discussions.





## 1 Introduction

Molecular interactions are fundamental to cellular activities [66, 39]. While molecular biology has traditionally studied how molecular structure governs mechanical and biochemical function [68, 45, 7], the perspective of information processing at the systems level is increasingly appreciated [75, 3, 71, 9]. The design of synthetic biomolecular systems that perform computation, both theoretical and experimental, provides an approach to clearly isolate and understand how specific molecular mechanisms and phenomena relate to information processing [51, 64, 77, 52, 50]. Two prominent paradigms within DNA nanotechnology [57] are algorithmic self-assembly of DNA tiles, where information is encoded spatially within multicomponent aperiodic crystalline structures whose growth by cooperative attachment events can emulate cellular automata and Boolean logic circuits and neural pattern recognition [77, 76, 23, 78, 22, 24], and DNA strand displacement reaction networks, where information is encoded in the concentrations of freely diffusing molecules whose mass-action reaction kinetics can emulate analog dynamical circuits and Boolean logic circuits and neural computation [53, 82, 19, 65, 62, 43], including supervised learning and reservoir computing [42, 20, 28, 80]. These two paradigms respectively represent solid-phase molecular mechanisms, in the sense that DNA tiles form (relatively) permanent bonds with their neighbors, and gas-phase mechanisms, in the sense that DNA strand displacement complexes operate at dilute concentrations such that they are mostly not in contact with any neighbor except during brief reactive interactions.

The liquid phase is conceptually situated between these two states of matter, in the sense that molecules are constantly in contact with their neighbors but rapidly changing which molecules are their neighbors. In living cells, liquid-liquid phase separation (LLPS) drives the formation of membraneless organelles, which are condensates consisting of proteins and nucleic acids that are involved in the regulation of intracellular biochemical processes [8, 35]. Condensate assembly is governed by the interaction strengths of many molecular species, a feature that gives them a rich potential for complex phase behavior [36, 37, 47] that can be explored synthetically with multicomponent DNA nanostar systems [49, 38, 15]. Together, the biological ubiquity of liquid-phase condensates and the demonstrated programmability of synthetic nucleic acid systems establish multicomponent liquids as a strong candidate for a third paradigm for molecular information processing based on liquid-phase mechanisms [72].

Recent work using several complementary modeling frameworks has highlighted the potential for multicomponent liquid mixtures to process information by phase separation. Multicomponent liquid lattice models explicitly capture local spatial arrangements of molecules and solvent; Chalk et al. showed that such a framework has a formal connection to the Boltzmann machine, providing both theoretical grounding and an established learning method [16]. The mean-field model, by contrast, averages over spatial positions and describes the mixture solely through bulk composition vectors, making it analytically tractable. Teixeira et al. pointed out that bulk phases of multicomponent liquids can be metastable, supporting associative memory retrieval analogous to the Hopfield network when a Hebbian rule was used for learning molecular interaction energies, albeit only with a cubic modification to the standard energy model [12]. This "liquid Hopfield model" was then extended to a continuous spatial model of LLPS, which exhibited complex domains of the metastable phases [70]. Zentner et al. considered the case of condensation in a "surface volume" around fixed "input" molecules and used backpropagation training to find interaction energies and reservoir potential energies such that condensation of different phases implemented complex classification tasks [81]. The rich behaviors exhibited in these works motivate the search for

simpler "local" learning algorithms that don't require global loss functions and backpropagation of gradient terms yet are still sufficient for tuning multicomponent phase separation for a variety of information processing tasks.

The continuous Hopfield network (CHN) model is a well-studied framework in neural networks for associative memory and optimization [31, 32, 33]. In this model, neurons occupy continuous states and are connected by symmetric interaction weights that define an energy landscape. The system follows deterministic dynamics that monotonically descend on the energy, as guaranteed by the existence of a Lyapunov function, and converge to stable attractors that act as stored memories or solutions to optimization problems. Contrastive Hebbian learning (CHL) is a natural training method for this framework, having originally been introduced for continuous Hopfield networks. The training method relies on energy minimization with local update rules rather than explicit loss functions, where local means that states of two interacting neurons are sufficient to determine their weight update. CHL has been shown to be equivalent to backpropagation under certain conditions [48, 79].

In this paper, we show that CHL can be adapted to two previously-studied models of multicomponent LLPS: the surface condensation model of Zentner et al. [81] and the spatial phase separation model of Teixeira et al. [70] without the cubic terms. The trained systems demonstrate a range of computational tasks, including associative recall, linear and nonlinear input-output relations, classification of MNIST digits, and programmable spatial phase separation with coexisting target phases. These results help establish multicomponent liquid mixtures as a physically realizable and trainable computational substrate, and cement CHL as a versatile learning rule for exploring the design space.

## 2 Learning in a surface model of multicomponent liquids

In this section, we introduce a mean-field model of a single-compartment liquid mixture in exchange with a reservoir that is equivalent to the surface condensation model of Zentner et al. [81]. Their paper discusses in detail the physical and biological credibility of the model. The mean-field energy can be derived from a lattice model (Figure 1a) and the following assumptions [26, 34, 81]: we consider a single well-mixed compartment of an incompressible liquid mixture with $n$ equal-volume components and a fixed volume $v$. The composition vector fully characterizes the system state of the mixture: $\vec{\phi} = (\phi_1, \phi_2, \ldots, \phi_n)$, where $\phi_i$ is the volume fraction of the $i^{th}$ component and is subject to $\sum_{i=1}^{n} \phi_i = 1$. Thus, an $n$-component system has $n-1$ independent variables. $\chi_{ij}$ represents the effective interaction between the liquid components, with positive values indicating a tendency to demix and negative values indicating a tendency to mix. It is defined as $\chi_{ij} = \frac{z}{RT}(G_{ij} - \frac{G_{ii}+G_{jj}}{2})$ , where $z$ is the average number of interactions (valency) and $G_{ij}$ is the microscopic interaction between components $i$ and $j$. Consequently, $\chi_{ij}$ is symmetric and diagonal terms $\chi_{ii}$ are equal to 0. Then the free energy of the reservoir-coupled compartment is described by a Flory-Huggins energy density with equal-volume species [26, 34]:

$$f_{FH}(\vec{\phi}) = cRT\left(\sum_{i,j=1}^{n} \frac{1}{2}\phi_i\chi_{ij}\phi_j + \sum_{i=1}^{n} \phi_i \log \phi_i\right) - c\sum_{i=1}^{n} \mu_{i,res}\phi_i \tag{1}$$

where $c$ is the total molecular concentration, $R$ is the gas constant, $T$ is the absolute temperature, and $\mu_{i,res}$ is the external reservoir potential. For simplicity, we will make energy density dimensionless by setting $c = 1$ and $RT = 1$.

Remarkably, this energy is formally similar to the standard Lyapunov function for continuous Hopfield networks [32, 73]

$$E_{CHN}(\vec{x}) = -\frac{1}{2}\sum_{i,j} w_{ij}x_i x_j - \sum_i b_i x_i + \sum_i \int_0^{x_i} g^{-1}(x)\, dx. \tag{2}$$

The corresponding dynamics are

$$\frac{ds_i}{dt} = -\frac{\partial E_{CHN}}{\partial x_i} = \sum_j w_{ij}x_j + b_i - s_i \qquad \text{where} \qquad x_i = g(s_i) \tag{3}$$

where $x_i$ is the continuous activity of neuron $i$, $w_{ij}$ gives the symmetric zero-diagonal synaptic weight matrix, $b_i$ is the bias, and $g$ is the activation function relating the neural input $s_i$ to its output $x_i$. When $g(s) = 1/(1+\exp(-s))$ so $0 < x_i < 1$, the last term of eqn 2 is $\sum_i x_i \log x_i + (1-x_i)\log(1-x_i)$, which is similar to the middle term of eqn 1, while the other terms are identical if $x_i = \phi_i$ and $w_{ij} = -\chi_{ij}$ and $b_i = \mu_{i,res}$. This leaves the major difference of the constraint, present only in surface condensation, that $\sum \phi_i = 1$. Interestingly, a known generalization of the continuous Hopfield model also bakes in this constraint, with the motivation that constraints of this sort are useful for optimization tasks such as the traveling salesman problem, in which case the natural Lyapunov function is identical to the Flory-Huggins energy [73]. Figure 1b illustrates the connection to the continuous Hopfield network for the case of a simple 2-component system, and emphasizes the impact of the constraint. Figures 1c and d illustrate how, like the continuous Hopfield model despite differences due to the constraint, the number of metastable states in the Flory-Huggins landscape can scale with the number of components.

A natural dynamics for the surface model is the model A dynamics [30, 81], which corresponds to gradient descent on the energy $f_{FH}$ subject to the $\sum_i \phi_i = 1$ constraint while allowing mass exchange with the reservoir. Here, we use a Lagrange multiplier $\lambda$ to enforce the constraint. We will use the modified free energy $f_{\lambda FH} = f_{FH} + \lambda(\sum_{i=1}^n \phi_i - 1)$ where $\lambda = -\frac{1}{n}\sum_{i=1}^n \mu_{i,FH} = -\langle \mu_{FH} \rangle$ is dynamically determined so as to ensure the constraint remains satisfied (in which case $f_{\lambda FH} = f_{FH}$). The modified dynamics then become

$$\frac{d\phi_i}{dt} = -M\frac{\partial f_{\lambda FH}}{\partial \phi_i} = -M(\mu_{i,FH} + \lambda) \tag{4}$$

where we define $\mu_{i,FH} = \frac{\partial f_{FH}}{\partial \phi_i} = 1 + \log\phi_i + \sum_{j=1}^n \chi_{ij}\phi_j - \mu_{i,res}$ as the chemical potential of species $i$ and $M > 0$ is the constant mobility. The dynamics monotonically decrease this energy with an equilibrium condition $\mu_{i,FH} = \langle \mu_{FH} \rangle$ (see Appendix A.2 for detailed proof). Thus, the constrained minima of the liquid system's energy are dynamically stable fixed points under model A dynamics, making them the physical analogs of memory attractors in Hopfield networks. However, their usefulness as associative memories depends on whether energy parameters can be learned, whether arbitrary memory patterns can be stored or a limited subset (Hopfield's result pertained to random patterns that may have many overlapped bits but are statistically orthogonal), and whether the basins are structured such that the dynamics starting from perturbed patterns generally leads to correct recall (as opposed to spurious states or reversion to a single dominant memory). In the classical continuous Hopfield model, a Hebbian learning rule is used to set these parameters [31, 32]. However, the volume-fraction constraint reduces the effective degrees of freedom in the liquid mixture and alters the stability conditions relative to the unconstrained case [74]. Consequently, the Hebbian learning rule in the classical model may not be as effective here.

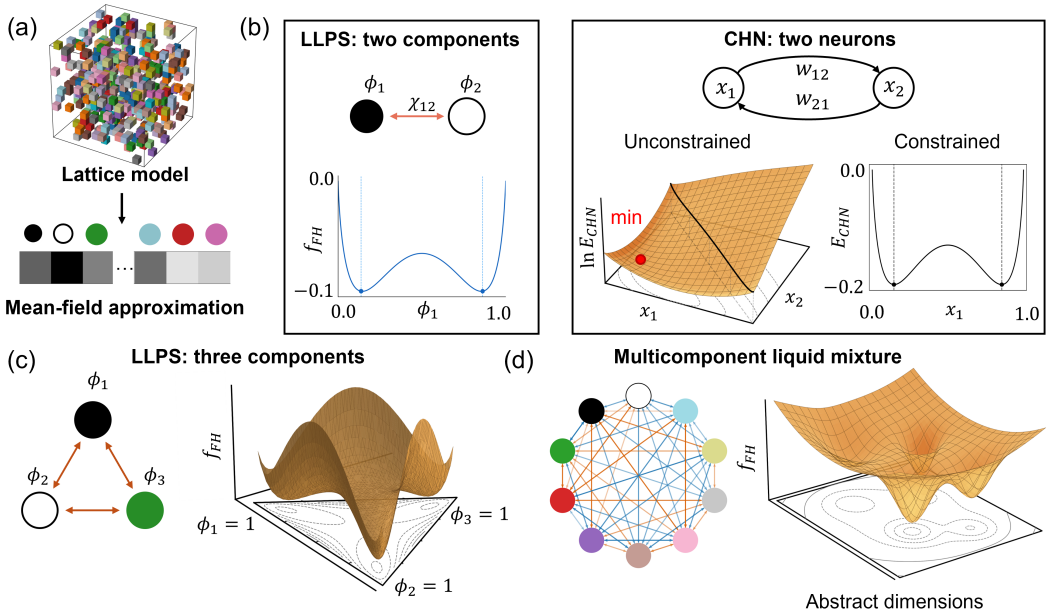


**Figure 1** Energy landscapes of multicomponent liquids. (a) The liquid mixture model with Flory-Huggins energy is an approximation of a lattice model in the mean-field limit. (b) Left: A representative Flory-Huggins energy for phase separation of a pair of repulsive components ($\chi_{12} = \chi_{21} = 5$) and zero reservoir potential ($\mu_{i,res} = 0$). Two stable states can be found, where one phase has a depleted $\phi_1$ component and the other is enriched in $\phi_1$. Right: A continuous Hopfield network (CHN) with two mutually inhibitory neurons ($w_{12} = w_{21} = -10$ and $b_1 = b_2 = 0$). The unconstrained energy landscape has only one global minimum. When the constraint $x_1 + x_2 = 1$ is applied, the system has two stable attractors. (c) A ternary phase system with strong repulsive interactions shows three stable states in the energy landscape. (d) A conceptual illustration showing that the number of stable phases in an LLPS system scales with the number of components.

## 2.1 Contrastive Hebbian learning of stable target states

We now show that CHL provides a general method for training stable attractors in a liquid-phase mixture. CHL was originally introduced by Movellan to train continuous Hopfield networks for constrained optimization problems [48]. It belongs to a family of contrastive learning methods first developed for Boltzmann machine models [14, 29, 1]. The CHL framework has several distinctive properties worth noting. First, unlike backpropagation-based methods, the training is not specified by an explicit loss function on output states; the system is instead guided by example compositions. Second, hidden units can be incorporated naturally, enriching the representational capacity of the system. Third, the weight update rule takes the Hebbian-like "fire together, wire together" form, depending only on the local states of the two connected components. Unlike pure Hebbian learning that positively reinforces desired activity, contrastive learning also has a similarly local anti-Hebbian phase that disrupts undesired activity. We derive the weight update rules for our model below.

Because model A dynamics are deterministic, given any initial composition $\vec{\phi}^o$, the system always converges to a unique final phase vector $\check{\phi}$ with associated free energy $\check{f}_{FH}$, where a $(\check{})$ denotes an equilibrium value. CHL minimizes the energy difference between the target phases and the system's natural equilibrium, training the system to store target phases as stable local minima of the free-energy landscape.

When the system is clamped, it is partitioned into two subsets $\vec{\phi} = (\vec{\phi}_{\text{fixed}}, \vec{\phi}_{\text{free}})$, where a subset of components is held fixed at target values while the remaining components are free to relax. Physically, clamping a component corresponds to maintaining its volume fraction at a prescribed fixed value (for example through exchange with a modified external reservoir held at the necessary corresponding chemical potentials). The system starts from an initial clamped state and is driven by the dynamics while holding clamped variables fixed. At equilibrium, the clamped state $\check{\phi}^{(+)}$ is associated with energy $\check{f}_{FH}^{(+)}$. Next, the clamps are released, and the system further relaxes to the free state $\check{\phi}^{(-)}$ with energy $\check{f}_{FH}^{(-)}$. The increased degrees of freedom guarantee $\check{f}_{FH}^{(+)} \geq \check{f}_{FH}^{(-)}$.

By minimizing the contrastive objective representing the energy difference between the two states,

$$J = \check{f}_{FH}^{(+)} - \check{f}_{FH}^{(-)} \ , \tag{5}$$

we can train the system to place stable attractors at target phases that respect the clamped compositions $\vec{\phi}_{\text{fixed}}$. This is accomplished by learning the interaction matrix $\chi_{ij}$ and the reservoir chemical potentials $\mu_{i,res}$. A conceptual schematic is presented in Figure 2a.

We now derive the learning rules for the interaction matrix $\chi_{ij}$:

$$\frac{\partial \check{f}_{FH}}{\partial \chi_{ij}} = \check{\phi}_i\check{\phi}_j + \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}}(1 + \log \check{\phi}_k + \sum_{l=1}^{n} \chi_{kl}\check{\phi}_l - \mu_{k,res}) = \check{\phi}_i\check{\phi}_j + \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}}\mu_{k,FH} \ . \tag{6}$$

Given the previously noted fact that at equilibrium $\mu_{i,FH} = \langle \mu_{FH} \rangle$ and by differentiating the constraint $\sum_{k=1}^{n} \check{\phi}_k = 1$ to obtain $\sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}} = 0$, we can see that

$$\frac{\partial \check{f}_{FH}}{\partial \chi_{ij}} = \check{\phi}_i\check{\phi}_j + \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}}\langle \mu_{FH} \rangle = \check{\phi}_i\check{\phi}_j + \langle \mu_{FH} \rangle \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}} = \check{\phi}_i\check{\phi}_j \ . \tag{7}$$

As this equation holds for any initial state or clamping condition, we can now obtain

$$\frac{\partial J}{\partial \chi_{ij}} = \frac{\partial \check{f}_{FH}^{(+)}}{\partial \chi_{ij}} - \frac{\partial \check{f}_{FH}^{(-)}}{\partial \chi_{ij}} = \check{\phi}_i^{(+)}\check{\phi}_j^{(+)} - \check{\phi}_i^{(-)}\check{\phi}_j^{(-)} \ . \tag{8}$$

To learn interaction strengths by performing gradient descent on $J$, we need $\frac{d\chi_{ij}}{dt} = -\frac{\partial J}{\partial \chi_{ij}}$, and thus the update rule is $\Delta\chi_{ij} \propto -\check{\phi}_i^{(+)}\check{\phi}_j^{(+)} + \check{\phi}_i^{(-)}\check{\phi}_j^{(-)}$. By a similar derivation (given in Appendix A.3), the learning rule for the reservoir potentials is $\Delta\mu_{i,res} \propto \check{\phi}_i^{(+)} - \check{\phi}_i^{(-)}$.

Several aspects of the training procedure are worth noting. First, the initial composition affects which attractor the system converges to. A common strategy is for the clamped stage to be initialized with the fixed components set to the target and the remaining components starting from uniform values plus random perturbations, which corresponds to a fully mixed, disordered state. This uniform starting point ensures the dynamics are not biased toward any particular attractor, and the small noise breaks the compositional symmetry to allow convergence to one of the stored phases. For the free stage, the system is initialized from the equilibrium of the clamped stage. The process is illustrated in Figure 2a. Second, although the learning rule is derived for a single target phase, it naturally extends to storing multiple phases by defining the contrastive objective as an expectation over a distribution $P$ of target phases $\langle J \rangle_P = \mathbb{E}_{\vec{\phi}^\circ \in P}[J]$. Gradient descent of $\langle J \rangle_P$ therefore corresponds to training that cycles through the set of target phases and averages the updates in each epoch. Thus the updates accumulate and superimpose to encode multiple attractors simultaneously, in direct

analogy to how the classical Hopfield network stores multiple patterns via summed Hebbian outer products. To encourage the learned system to remain "liquid-like", we clip energies such that $|\chi_{ij}| < 25$ after every update. Finally, as with any Hopfield-like system, the trained network may develop spurious attractors arising from interference among the encoded phases. Spurious attractors become more prevalent as the number of stored phases approaches the storage capacity of the system [5, 13]. In the following sections, we demonstrate the method through three examples. Details of training methods and simulation methods are in Appendices A.1 and A.2.

## 2.2 Storing and retrieving target phases

Pattern storage and retrieval is the natural entry point, as it most directly tests whether the trained energy landscape stores the desired attractors. Here we consider binary patterns $\vec{x} = (x_1, \ldots, x_n), x_i \in \{0, 1\}$. Each binary pattern is mapped to a composition $\vec{\phi}$ via a softmax function $\phi_i = e^{\beta x_i} / \sum_{i=1}^{n} e^{\beta x_i}$ and $\beta = 0.5$, ensuring positivity and sum to unity.

As a proof of concept, we train a 32-component mixture to store and retrieve five target phases with overlapping components. In binary phase separation (for example water/oil separation), components can be distinguished by their repulsion from components in the opposing phase. Here, each component must play a distinct role in each of the five trained phases, and the interaction matrix encodes a rich associative structure in which every component's identity is defined by its relationships with other components across all stored phases simultaneously. The task complexity is approaching the storage capacity ($\sim 0.14n$) predicted for classic Hopfield networks [4]. The set of memories is shown in Figure 2c. Training proceeds by gradient descent on the contrastive objective defined above. In each epoch, the clamped target phase $\check{\phi}^{(+)}$ is sampled uniformly from the set of memories and the free phase $\check{\phi}^{(-)}$ is obtained by releasing from the clamped target phase. The interaction matrix is initialized using the Hebbian rule, and the reservoir chemical potentials are initialized to zero. The contrastive objective $J$ averaged over each epoch, $\langle J \rangle_{epoch}$, tends to decrease with epochs and roughly converges to near 0 (Figure 2b).

Retrieval is tested by initializing the system with randomly perturbed versions of the stored memories. Figure 2d shows an example trajectory in which the free energy decreases from the initial random composition and the system converges to one of the stored phases. We characterize the retrieval performance of the trained system as a function of the flip ratio, defined as the fraction of components in the initial composition that are randomly reassigned relative to the target phase. We perform a systematic study by randomly generating sets of five memories, where each component appears in at least one memory but bits are otherwise unbiased. Target memory sets were sorted into bins based on the average number of overlapping components. We test memory retrieval for five memory sets each from bins centered on 4 to 12 overlaps and compare the performance to the classical Hopfield network on the same memory sets. The results are shown in Figure 2e. Retrieval rates decline with increasing flip ratio as expected, and for memories with low average overlap (4 and 6 components) the liquid-phase system performs comparably to the classical Hopfield network. At higher overlaps (greater than 8), performance degrades relative to the Hopfield baseline. These results confirm, consistent with [12, 70], that a multicomponent liquid mixture can perform associative recall across a non-trivial memory load – although, consistent with prior investigations of the capacity of multicomponent liquids to encode multiple phases [59, 83, 17, 54], their memory capacity may be lower than the $\sim 0.14n$ statistical capacity of classical Hopfield networks.

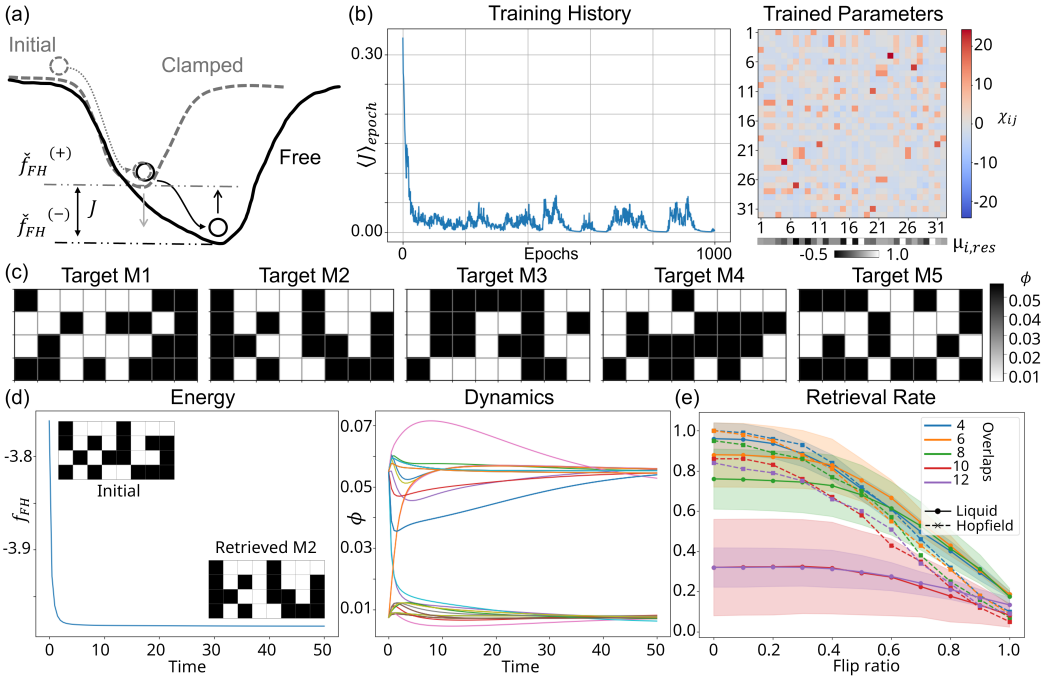


**Figure 2** Training an associative memory. (a) Conceptual schematic of CHL. From an initial state with some components clamped, the system relaxes to an equilibrium state. In the free phase, some clamps are removed, and the system relaxes again to a new lower equilibrium, with increased degrees of freedom. The learning rule minimizes the contrastive objective to place stable attractors at the target compositions. (b) The contrastive objective $\langle J \rangle$ averaged in each epoch. CHL drives $\langle J \rangle$ to near zero. (c) A set of five 32-component memories used for training. Each pixel represents the volume fraction of one component; memories share several overlapping "on" components. (d) An example trajectory in which an initial composition vector relaxes to memory 2 by minimizing the free energy. (e) Retrieval rate vs. flip ratio for varying numbers of overlapping components. Solid lines show mean and shaded bands show the standard deviation across five samples of sets of five 32-bit memories. Dashed lines show the benchmark results from classical Hopfield network.

## 2.3 Input-output relations and simple classification

A strength of the Hopfield network is that its all-to-all architecture allows neurons to be flexibly designated as inputs, outputs or hidden units (Figure 3a). This makes the system capable of solving some constrained optimization problems. By clamping subsets of neurons, the network explores a constrained energy landscape, enabling tasks such as learning input-output relations and performing classification. In this section, we demonstrate that CHL can train a liquid-phase mixture to represent input-output relations and perform classification tasks similar to those demonstrated by Zentner et al. using backpropagation [81].

To define the tasks, the components of the phase vector are partitioned into three categories: inputs, outputs, and hidden components $\vec{\phi} = (\vec{\phi^I}, \vec{\phi^O}, \vec{\phi^H})$. Training samples are drawn from the target input-output relations. In the clamped phase, the input and output components are fixed to the target values while the hidden components are initialized to uniform values with random noise; the system then relaxes to equilibrium. In the free phase, only the input components remain fixed; the output and hidden components are released, and the system relaxes to a new equilibrium. The CHL rules are then applied by contrasting the equilibrium states from the clamped and free phase, shaping the free-energy landscape on

the constraint manifold toward the target output states. In all input-output and classification examples, an additional inert solvent component is included to absorb the remainder of the volume fraction, decoupling the constraint from the task-relevant components and allowing the input-output mapping to be defined independently of the normalization. The astute reader will notice that the quantity being minimized is no longer a measure of the average stability of target phases in the free energy landscape, $\langle J \rangle_P$, but rather the average stability of target phases in constrained (input-clamped) energy landscapes, which we may also write as $\langle J \rangle_P$ but where the distribution $P$ contains more information, as described in Section 4.

As a baseline, we train simple linear relations (Figure 3b), sampling input-output pairs uniformly from the target functions. A single input–output pair without hidden components can represent a linear relation, but the volume-fraction constraint severely limits the accessible slopes. With additional hidden components, the system can represent more complex nonlinear analog functions, including a sigmoid and an inverted parabola (Figure 3b). Error bars reflect the standard deviation over test inputs evaluated with fixed trained parameters.

Next, we train a two-input, two-output system to perform binary classification across several decision boundaries. In biological contexts, classification-like behavior arises in the spatial condensation of regulatory proteins: condensates form at specific genomic loci to promote or suppress gene expression [61]. In our model, the two output components are trained to respond with a high volume fraction to combinations of the two input components. We begin with a simple linear example as shown in Figure 3c. We then test nonlinear classification boundaries including AND, XOR, and circular boundaries. All interaction matrices are provided in Appendix Figure 7. Notably, we found that successful classification required weighted sampling of training examples, with higher sampling density near the decision boundary. In particular, we draw 70% of our training samples from a region defined as being within distance 0.05 from the decision boundary. This suggests that while CHL eliminates the need for an explicit loss function, the trade-off is that the sampling distribution may need to reflect the task structure. These limitations are discussed further in Section 5.

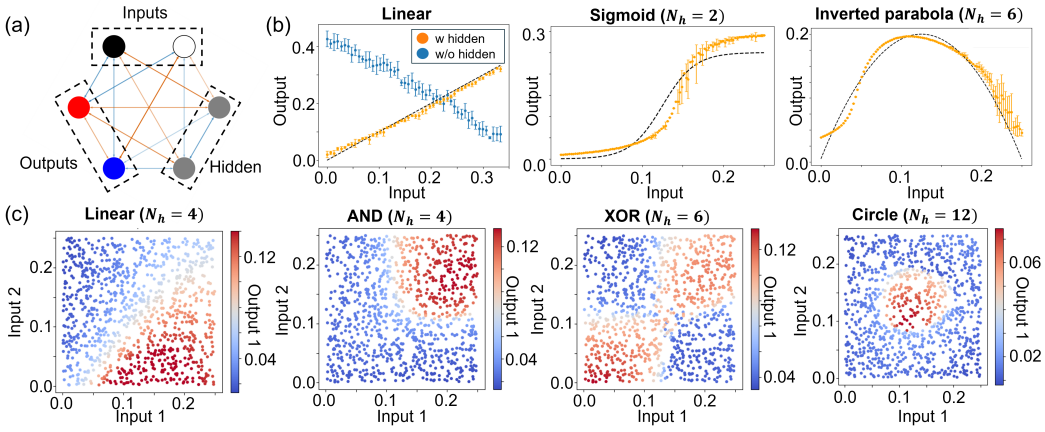


**Figure 3** Training low-dimensional input-output functions. (a) Schematic showing the partitioning of components into input, output, and hidden classes. (b) Examples of trained analog input-output relations. The linear case illustrates that without hidden components the system cannot match an arbitrary slope. With hidden components, sigmoid and inverted parabola relations can be trained, demonstrating continuous function representation. Error bars show the standard deviation over test inputs (fixed trained parameters). (c) Trained 2-input, 2-output classification across decision boundaries ranging from linear to nonlinear. In all panels, $N_h$ is the number of hidden species.

## 2.4 Classification of the MNIST dataset

Hopfield-type networks extend to high-dimensional classification tasks [10]. Here we benchmark the system on MNIST classification. The MNIST dataset consists of images of 28×28 pixels, corresponding to 784 input components at a 1:1 mapping. At this number of components, our integration of model A dynamics exhibits numerical instabilities; we reduce the input to 7×7=49 components by average pooling with a 4×4 kernel. Ten output components represent digits 0 through 9. We add 40 hidden components and one solvent component, bringing the total to 100 components (Figure 4a). CHL training is performed by sampling uniformly from the dataset, with the correct output component clamped to a high volume fraction and all other output components held low. We define a successful classification as one in which the correct output has the highest volume fraction. The best-performing trained system achieves an accuracy of ∼75.5%, in the ballpark of prior results [16, 81]. The confusion matrix is shown in Figure 4c, with trained parameters in Figure 7c.

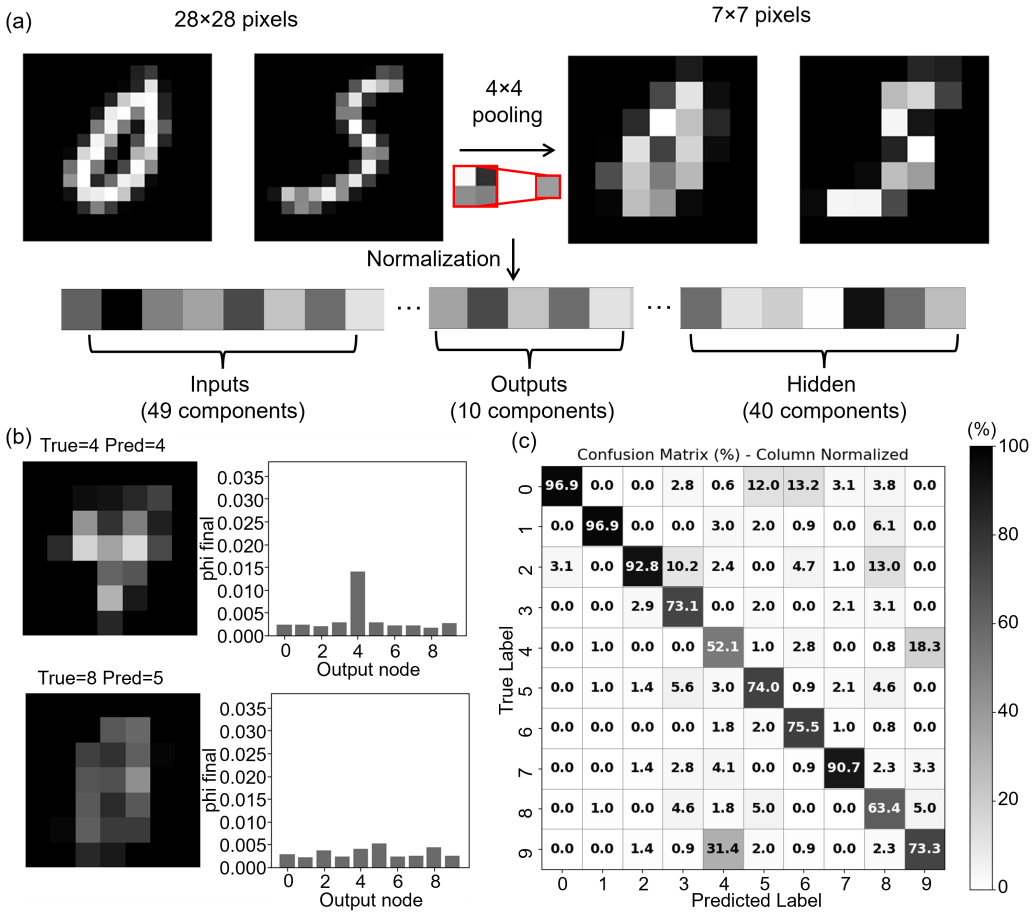


**Figure 4** Training a high-dimensional classification task. (a) The MNIST dataset is downscaled to 49 input components. Inputs are normalized to occupy a total volume fraction of 0.5, with the remainder consisting of output, hidden, and solvent components. (b) Classification examples. Digit 4 is correctly classified, whereas digit 8 is incorrectly classified. (c) Confusion matrix for digit classification, where values indicate the probability of the true label given the predicted label.

## 3 Learning in spatial phase separation systems

In the preceding section, we explored the computational capacity of a well-mixed, single-compartment liquid mixture. Here, we extend the framework to spatially resolved phase separation. The ability to train coexisting spatial phases expands the liquid-phase computing paradigm to encompass pattern formation problems, with implications for understanding intracellular organization and designing programmable materials.

Consider a liquid of volume $v$ with $n$ components. The local composition at position $\mathbf{x}$ and time $t$ is described by $\vec{\phi}(\mathbf{x},t) = (\phi_1, \phi_2, \ldots, \phi_n)$, where $\phi_i(\mathbf{x},t)$ is the volume fraction of component $i$, subject to the local normalization constraint $\sum_{i=1}^{n} \phi_i(\mathbf{x},t) = 1$. Further, the volume will be closed (not in exchange with a reservoir) and thus the total mass of each species will be conserved, so $\int_v \phi_i \, d\mathbf{x}$ is constant. Our implementation is in two dimensions with periodic boundaries, but generalization to three dimensions is straightforward.

The total free energy combines the Flory-Huggins mixing energy, omitting the reservoir potential terms, with a Cahn-Hilliard interfacial penalty [47, 60]. The training parameters therefore consist only of the interaction matrix $\chi_{ij}$. We use a fixed and uniform spatial gradient penalty coefficient $\kappa$; more general treatments could allow $\kappa_{ij}$ to vary between pairs of components. The energy density is $f_{CH} = f_{FH} + \sum_{i=1}^{n} \frac{\kappa}{2} |\nabla \phi_i|^2$, and the total energy is

$$F_{CH} = \int_v \left( f_{FH} + \sum_{i=1}^{n} \frac{\kappa}{2} |\nabla \phi_i|^2 \right) d\mathbf{x}, \tag{9}$$

where $\nabla \phi_i = (\frac{\partial \phi_i}{\partial x}, \frac{\partial \phi_i}{\partial y})$ is the spatial gradient of species $i$. We can define the chemical potential $\mu_{i,CH}(\mathbf{x},t) = \frac{\delta F_{CH}}{\delta \phi_i(\mathbf{x},t)} = \sum_{j=1}^{n} \chi_{ij}\phi_j(\mathbf{x},t) + 1 + \log \phi_i(\mathbf{x},t) - \kappa \nabla^2 \phi_i(\mathbf{x},t)$. Model B [30] gives us the appropriate dynamics for this mass-conserving setting, as has been used in prior spatial LLPS simulations [47, 70, 60]:

$$\frac{\partial \phi_i}{\partial t} = \nabla \cdot \left( M_i \nabla \left( \frac{\delta F_{CH}}{\delta \phi_i} \right) \right). \tag{10}$$

Again setting $M_i = M$ (constant), we obtain $\frac{\partial \phi_i}{\partial t} = M \nabla^2 \left( \frac{\delta F_{CH}}{\delta \phi_i} \right)$, where $\nabla^2 \phi_i = \frac{\partial^2 \phi_i}{\partial x^2} + \frac{\partial^2 \phi_i}{\partial y^2}$ is the Laplacian, or divergence of the gradient. In our results, the conserved quantities are within the range of machine precision.

As in the single-compartment case, a Lagrange multiplier $\lambda(\mathbf{x},t)$ is introduced to enforce the local normalization constraint $\sum_{i=1}^{n} \phi_i(\mathbf{x},t) = 1$. The modified free-energy density becomes $f_{\lambda CH} = f_{CH} + \lambda \left( \sum_{i=1}^{n} \phi_i - 1 \right)$, which is integrated over the space to give $F_{\lambda CH}$. The modified dynamics become $\frac{\partial \phi_i}{\partial t} = M \nabla^2 \left( \frac{\delta F_{\lambda CH}}{\delta \phi_i} \right)$, with $\lambda = -\frac{1}{n} \sum_{i=1}^{n} \mu_{i,CH} = -\langle \mu_{CH} \rangle$. Under these dynamics, the equilibrium condition is $\mu_{k,CH}(\mathbf{x}) - \langle \mu_{CH} \rangle(\mathbf{x}) = C_k$, where $C_k$ is a constant independent of $\mathbf{x}$. A full derivation is given in Appendix A.4.

Analogous to the CHL picture in Section 2.1, we can initialize the spatial system with various properties clamped: for instance, the spatial configuration of particular phases, or the global compositions of the particular components. Ideally, it will relax to the clamped equilibrium $\check{\phi}(\mathbf{x})^{(+)}$, with the corresponding $\check{F}_{CH}^{(+)}$. Next, the clamp is fully or partially released, allowing the system to settle to a new equilibrium $\check{\phi}(\mathbf{x})^{(-)}$ with associated $\check{F}_{CH}^{(-)}$.

As before, our contrastive objective is $J = \check{F}_{CH}^{(+)} - \check{F}_{CH}^{(-)}$. Since the free state has more degrees of freedom, $\check{F}_{CH}^{(+)} \geq \check{F}_{CH}^{(-)}$ so again $J \geq 0$. The CHL rule for the spatial model, derived in Appendix A.5, is:

$$\Delta \chi_{ij} \propto -\int_v \check{\phi}_i^{(+)} \check{\phi}_j^{(+)} d\mathbf{x} + \int_v \check{\phi}_i^{(-)} \check{\phi}_j^{(-)} d\mathbf{x}. \tag{11}$$

This has the same Hebbian structure as the rule for the surface model, integrating $\check{\phi}_i\check{\phi}_j$ pointwise over space. The update promotes interactions between species that co-occur in the target pattern, and decreases them for species that co-occur in the free equilibrium, sculpting the free energy landscape to favor the desired spatial morphology.

## 3.1 Training for stable coexisting phases

We now demonstrate training of the interaction matrix to stabilize prescribed coexisting phases. This is directly related to the design of synthetic condensates with specified compositions, or to biological pattern formation, where distinct spatial domains must be robustly maintained. Suppose we wish to train a system to stably support a set of $m$ coexisting phases $S = \{\vec{\phi}^{\alpha}, \vec{\phi}^{\beta}, ...\}$. For clamping, we assign a composition $\vec{\phi}(\mathbf{x}) \in S$ to each location $\mathbf{x}$ with uniform probability in all demonstrations below; thus $\int_v \check{\phi}_i^{(+)}\check{\phi}_j^{(+)} = \frac{1}{m}\sum_{\sigma\in\{\alpha,\beta,...\}} \vec{\phi}_i^{\sigma}\vec{\phi}_j^{\sigma}$. For the free phase of each training epoch, the composition field is initialized uniformly at the mean phase $\vec{\phi}^{\text{average}} = \frac{1}{|S|}\sum_{\sigma\in\{\alpha,\beta,...\}} \vec{\phi}^{\sigma}$ with small additive noise, and evolved under model B dynamics until the free energy falls below a convergence threshold. Since late-stage coarsening is a notably slow process, we consider the phase field to have converged sufficiently once it enters the coarsening regime (the threshold for convergence is provided in Appendix A.1). The CHL update rule is then applied to the interaction matrix. Here the learning aims to ensure that random initial states evolve to an energy similar to the mixture of target phases – there is no direct assessment that a given target state is stable and cannot relax further. In this sense, the training procedure here is disrupting spurious minima in the energy landscape until the correlation statistics of evolved states match the target mixture, making it conceptually more akin to the Daydreaming method for training classical Hopfield networks [58] than to the training procedure we used to justify the learning rule. This discrepancy between training methods is discussed in more detail in Section 4.

We demonstrate spatial CHL with a 32-component system trained to encode five phases with overlapping components (Figure 5). Simulations are halted in the coarsening regime, and so simulation times vary over training. By the end of training, we apply CHL at $t_{\text{train}} \approx 30$; to test, we simulate for longer, with $t_{\text{test}} \approx 70$. To determine the retrieved phases, we map the composition of each position to the target phase with the lowest Euclidean distance, $D_\sigma = ||\vec{\phi}(\mathbf{x}) - \vec{\phi}^{\sigma}||_2$, with $\text{Index}(\mathbf{x}) = \arg\min_{\sigma\in\{\alpha,\beta,...\}} D_\sigma$, and then average (over space) the positions assigned to the same phase. The system successfully retrieves all five target phases from a disordered initial condition, with the retrieved phases closely matching the prescribed targets. The interaction matrix (Figure 5b) exhibits a rich variety of attractive and repulsive interactions, showing components play different roles to encode five phases. Figure 5c shows the closest phase mapping and energy density $f_{CH}$ of the composition field. The space separates into regions each close to one of the target phases, with low $D_\sigma$ confirming that each composition is close to a target. For simulation times similar to those used during training, deviations are mainly confined to the domain boundaries where the composition profile is dominated by the interfacial energy penalty $\kappa$, while longer simulations develop regions of phase dissimilarity – highlighting the sensitivity of the technique to the "convergence" stopping criterion. Nonetheless, this result, which yields phase separation domains visually comparable to those obtained by an inverse design method [70], demonstrates that the local CHL rule can encode multi-phase associative memory directly into the interaction matrix of a spatially resolved system, providing a route to programmable spatial organization. The Supplementary Information includes additional results for the spatial model: a time series and spatial distributions of each component for the system in Figure 5; trained outputs of some simpler examples; and a demonstration of associative recall in the nucleation regime.

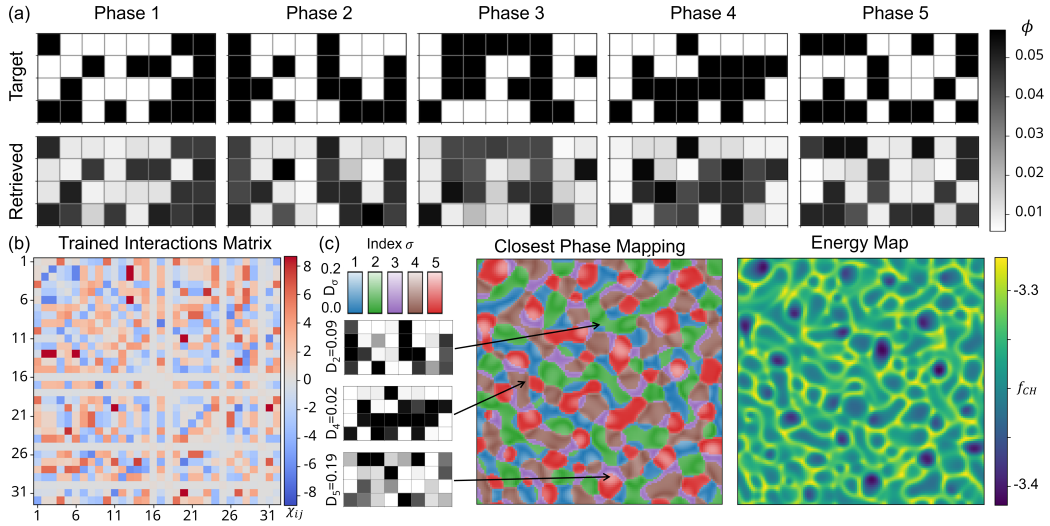


**Figure 5** Training for spatial mixtures of target phases. (a) Retrieved phases at $t_{\text{test}}$ match the target phases. (b) The trained interaction matrix. (c) The output of a simulation at $t_{\text{test}}$, showing the spatial distribution of the phases. For each point in space $\mathbf{x}$, we calculate the distance $D_\sigma$ to each of the five target phases; assign the point to the nearest phase; and then color by distance to that phase, with darker colors indicating smaller distances. On the left, we show compositions of three pixels with varying distances to the target phases. On the right, we plot the energy density $f_{CH}$. Boundaries between phases are further from the minima, and so have higher $f_{CH}$.

## 4 Unified learning with positive and negative design requirements

The preceding examples of CHL learning for multicomponent mixtures employed several variations of the method that was formally proven, including differences in what states were initialized and when and how much clamping was applied. To clarify how our proofs do, or do not, apply to these situations, we formulate a generalized framework for how contrastive learning can sculpt energy landscapes according to positive and negative design requirements. In this context, positive design requests that certain states be stable under certain conditions, while negative design requests that certain states not be stable. This is achieved by either pulling the energy down or up, respectively, for those states.

Consider a continuous dynamical system with energy landscape $f(\vec{\phi})$ on coordinate $\vec{\phi}$, so $\phi_i$ is the coordinate with index $i \in I$. Equilibration corresponds to downhill motion on the landscape. The landscape will be parameterized by model coefficients $\chi$ with respect to which we will perform gradient descent to optimize a loss function $\mathcal{L}$ specified below.

A learning task (or energy landscape sculpting task) is defined in terms of samples $s = (\vec{\phi}^o, c^{(+)}, c^{(-)}, \vec{\phi}^*)$, where $\vec{\phi}^o$ is an initial state for positive design, $c^{(+)} \subseteq I$ is a set of target indices being clamped, $c^{(-)} \subset c^{(+)}$ is a less constrained set of indices indicating how much the clamp will be relaxed, and the (typically random) initial state $\vec{\phi}^*$ will be used for negative design. There will be a probability distribution $P(s)$ that governs the selection of desired target states and clamping conditions, the assignment of initial states of hidden species, and the random initial states that should not lead to new spurious minima.

For positive design, define $\check{\phi}^{(+)}$ to be the equilibrium state achieved from $\vec{\phi}^o$ when clamping variables in $c^{(+)}$, while $\check{\phi}^{(-)}$ is the subsequent equilibrium state after relaxing the clamping to $c^{(-)}$. As before, the drop in energy $J(s) = f(\check{\phi}^{(+)}) - f(\check{\phi}^{(-)}) \geq 0$ with equality if the equilibrium is stable under these conditions.

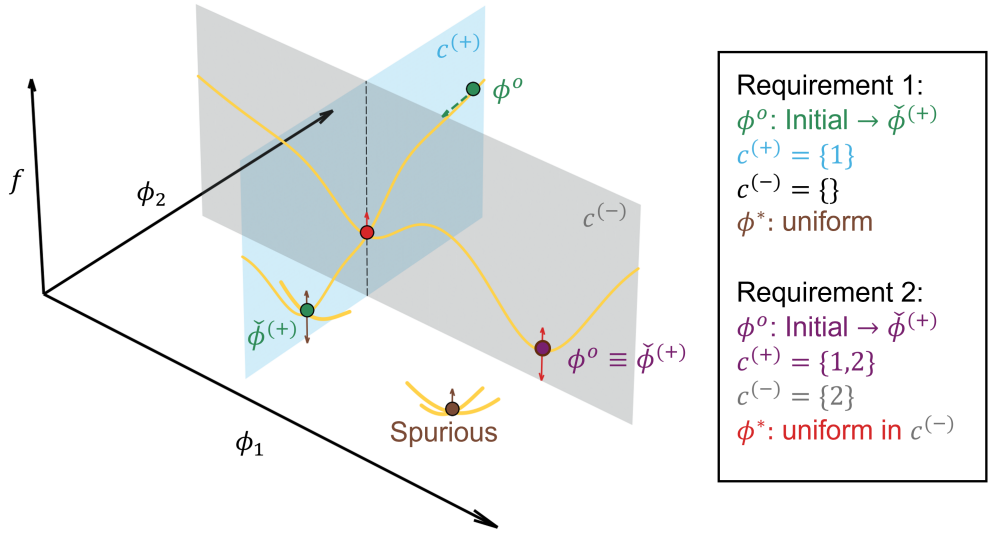


**Figure 6** Sculpting an energy landscape. Two types of design requirements are illustrated for a two-dimensional energy landscape $f$. Samples for requirement 1 chose a random value for the initial $\phi_2$ (the "hidden species") while clamping $\phi_1$ to the blue subspace; the positive design criterion is that when the clamp is released, $\phi_1$ remains at a local minimum; the negative design criterion is that there are no unconstrained local minima (such as the brown state). Samples for requirement 2 initially clamp both $\phi_1$ and $\phi_2$, and then release $\phi_1$ so dynamics explores the grey subspace; the positive design criterion is that $\phi^o$ is a constrained local minimum in the grey subspace; the negative design criterion is that there are no other constrained local minima (such as the red state). Vertical arrows indicate the effects of positive and negative learning. The illustrated energy landscape is at an intermediate stage of learning where the positive design requirements are met but the negative design requirements are not yet met.

For negative design, the goal is to destabilize spurious minima in the $c^{(-)}$ subspace, i.e. those other than the targeted states that are being trained to be stable when $c^{(-)}$ is clamped. Defining $\check{\phi}^{(*,-)}$ to be the equilibrium state evolved from $\vec{\phi}^*$ while clamping $c^{(-)}$, we are interested in the difference $H(s) = f(\check{\phi}^{(+)}) - f(\check{\phi}^{(*,-)})$ that, on average, measures how much lower the energies reached from random initial states are below those achieved by the desired states. The justification for this criterion is subtle and not universal, as discussed below.

Figure 6 illustrates positive and negative design of an energy landscape. It should be clear that with multiple requirements in place, for example many parallel slices through the landscape via clamping, many detailed features of the landscape can be designed, including valleys and pathways, not just the minima.

The unified contrastive learning rule performs gradient descent on the loss function

$$\mathcal{L}(\chi) = \alpha \sum_s P(s)J(s) + (1-\alpha)\sum_s P(s)H(s) = \alpha\langle J\rangle_P + (1-\alpha)\langle H\rangle_P \tag{12}$$

where $0 \le \alpha \le 1$ is the user-defined balance between positive and negative design requirements. Prior derivations of $\frac{\partial f(\check{\phi})}{\partial \chi_{ij}}$ can be applied without change to obtain

$$\Delta\chi_{ij} \propto -\check{\phi}_i^{(+)}\check{\phi}_j^{(+)} + \alpha\check{\phi}_i^{(-)}\check{\phi}_j^{(-)} + (1-\alpha)\check{\phi}_i^{(*,-)}\check{\phi}_j^{(*,-)} \tag{13}$$

for sample $s$ chosen from $P(s)$, where a spatial integral is added for the spatial case. In other words, the energies of desired states are pushed down, while the energies of the two kinds of undesired states are pushed up.

The learning examples demonstrated earlier can be interpreted in this context. For training associative memories (Figure 2), sample $s = (\vec{\phi}^o, c^{(+)}, c^{(-)}, \vec{\phi}^*)$ is chosen with $\vec{\phi}^o$ being uniformly one of the five memories, $c^{(+)}$ clamping all 32 species, $c^{(-)}$ clamping none, and $\vec{\phi}^*$ being irrelevant since $\alpha = 1$. For training 2-input classification (Figure 3c), sample $s$ is chosen with $\vec{\phi}^o$ being uniform for non-input species and preferentially near the decision boundary for input species, $c^{(+)}$ clamping the input and output species, $c^{(-)}$ clamping just the input species, and $\vec{\phi}^*$ being irrelevant since again $\alpha = 1$. For training coexisting spatial phases, sample $s$ is chosen with $\vec{\phi}^o$ being (effectively equivalent to) a single spatial distribution with one large domain for each of the five memories, $c^{(+)}$ clamping all species, $c^{(-)}$ clamping none, and $\vec{\phi}^*$ being the uniform average of the memories plus a small bit of noise, since $\alpha = 0$. In all these examples, it is easy to imagine that performance could have been better with an intermediate value of $\alpha$ and, for non-spatial models, $\vec{\phi}^*$ being uniform for unclamped species.

Although the local update rule for $\Delta\chi_{ij}$ does perform stochastic-sampled gradient descent on $\mathcal{L}(\chi)$, how do we justify this loss function in general? Since $J(s) \geq 0$ with equality if and only if the minimum is stable with the lower level of clamping, any training run that successfully reaches near zero should satisfy the positive requirements. However, $H(s)$ may be positive or negative on any given sample, and one can concoct example landscapes and samples such that $\langle H \rangle_P$ is positive or negative as well – for example, $\langle H \rangle_P < 0$ for a landscape with one very deep and narrow target minimum and one very shallow but broad target minimum, and changes to $\chi$ that make them more extreme will continue to decrease $\langle H \rangle_P$, consistent with gradient descent but presumably undesired. We can only heuristically observe that for Hopfield networks and for multicomponent liquids, changes that deepen an energy basin tend to also broaden it, while changes that fill in a basin also make it narrower – in which case we expect the negative design learning to successfully eliminate spurious minima. In that case, we can observe that $\langle H \rangle_P = 0$ if all spurious minima have been eliminated and the basin size of target minima is in proportion to its probability $P(s)$ for each clamping condition; this appears to us as a plausible but not guaranteed outcome. The negative design principle here is closely related to the Daydreaming method for training classical Hopfield networks to reduce spurious minima [58], and thus the general success of that method is promising for ours.

## 5 Discussion

The CHL learning rule was formalized in our surface and spatial multicomponent LLPS models to carve energy landscapes achieving several tasks. Encoding target compositions as stable minima of the surface-model Flory-Huggins energy connects our work to a broader literature of associative recall via energy minimization, first studied in abstract Hopfield neural networks, to the biologically-relevant process of multicomponent LLPS. Computation of low-dimensional input-output functions demonstrates that surface condensation may play an information-processing role even in systems with few unique components. Classification of the MNIST dataset in the surface model affirms the neural properties of multicomponent LLPS: many unique components aggregate information via relatively weak signals in order to establish an output. Finally, training of stable coexisting phases in the spatial Cahn-Hilliard framework provides validation beyond the surface model of phases with overlapping components, incorporating spatial physical considerations such as the phases' interfacial penalties. These tasks were achieved by the same learning framework: specification of a contrastive objective function $J$ that describes the difference in energy minima between more- and less-constrained systems. Our demonstration of these tasks across two distinct

multicomponent LLPS models illustrates how one specifies various target behaviors by contrastive objective functions $J$ and establishes CHL as a versatile learning rule, and illustrates that memory retrieval, classification, analog function computation, and phase coexistence design may all be treated as instances of energy-landscape sculpting.

The standard approach in modern machine learning is to specify and minimize a (potentially arbitrarily complex) loss function for specifying how well the system state or output matches a learning target. For example, backpropagation computes gradients of the loss function to send error signals through the network, which may be less biologically plausible [44]. A learning rule closer to CHL is equilibrium propagation [56], which is a local contrastive learning rule that compares the difference between a system's energy minima and the minima of an energy function "nudged" by the cost function. While these two methods may require different physical implementations for different learning rules, CHL requires only a physical implementation of clamping, shifting the burden to designing clamped and unclamped steps whose energy difference encodes the desired behavior.

The computational potential of multicomponent liquid mixtures demonstrated in our work inspires further questions. In the surface model, we show an example of memory storage and retrieval with five memories and 32 components. Empirically, storage capacity in Hopfield networks scales linearly with increasing number of units, while studies of the liquid systems suggest poorer scaling [37, 59, 12]. A feature lacking in prior design of coexisting phases is the ability to train interactions for hidden components, which may help stabilize target phases. Our demonstration of linear and nonlinear function computation also raises scaling questions. How does the number of required hidden components scale with the complexity of the decision boundaries? Zentner et al. [81] argue that, in the surface model, each extra hidden component adds an extra hidden linear decision boundary, and from this fact and other assumptions depict the potential for the model as a universal decision boundary approximator. Our one-input analog function tasks (the linear, sigmoidal, and inverted parabolic curves) present a new class of tasks in the surface model for which further analysis may explain hidden component scaling and potential universality of the model. Beyond these particular input-output tasks, a general theory that articulates the potential for sculpting arbitrary target energy landscapes via CHL may encompass memory storage, decision boundaries, function computation, and other behaviors in a unified framework.

The spatial model will enable investigation of pattern formation tasks beyond the phase coexistence we explored here. The design of a stable spatial pattern as a system equilibrium confers self-healing: equilibrium relaxation restores the pattern after perturbations. Although there is a rich literature on spatial pattern formation in reaction-diffusion systems [41, 43], which must be out-of-equilibrium, the formation of stable patterns at equilibrium requires interfacial penalties captured by the Cahn-Hilliard model. Interactions for complex stable spatial patterns beyond coexisting phases could be trained via CHL. We used a single constant $\kappa$ for the gradient penalty coefficient; since $\kappa$ is, physically, a pairwise term $\kappa_{ij}$, including it in the design space could potentially enable enriched spatial patterning. Our coexisting phases in the spatial model only loosely evoke Hopfield's original picture of memory storage and associative recall. Extension of classification and analog computation tasks illustrated in the surface model may extend naturally to the spatial model given a specification of spatial input and output representation. Questions related to scaling of function complexity with hidden component scaling and universality apply here just as well as the surface model case.

Finally, we discuss the prospects for physical implementation of our models in real molecular systems. Advances in molecular programming have greatly improved our ability to design molecular interactions with precision [27, 6]. Specific phase morphologies such as core-

shell structures have been shown to be rationally designable [49, 38, 2, 69, 63, 55]. Recently, multicomponent liquid condensates based on DNA nanostars have been demonstrated with up to nine components [15], approaching the regime where neural computation in liquid systems may become experimentally accessible. However, gaps remain between theory and experiment: concrete mappings between theoretical interaction parameters and real molecular design rules have yet to be established. Further, our model makes two key assumptions that potentially hinder physical implementation. First, our mean-field free energy assumes that the species are well-mixed in each volume. The degree to which this assumption may be violated can be checked post-training, e.g. via fine-grained lattice model simulations. Second, our single-compartment surface model assumes that the reservoir maintains chemical potentials $\mu_i$ for any learned interactions $\chi_{ij}$. Explicit construction of such a reservoir that itself does not internally phase separate is nontrivial: Teixeira et al. [12] propose $n$ chemical reservoirs, one for each component, and Zentner et al. [81] propose a composition for a single reservoir. An alternative approach towards physical realization of the surface model results is to adapt them to the Cahn-Hilliard framework discussed in Section 3, which reflects a simpler physical set-up without a reservoir. However, in physical implementations of the Cahn-Hilliard framework, the gradient penalty coefficient $\kappa$ will likely depend on microscopic interaction energies $G_{ij}$, and thus sufficient modeling may require the determination of pair-dependent $\kappa_{ij}$ parameters. Further, our model B dynamics do not capture other real physical phenomena, such as the bulk motion of droplets or fluids, nor large-scale Brownian effects. Our learning method requires only knowledge of equilibrium states, and is therefore compatible with non-deterministic dynamics. The Cahn-Hilliard equation may be modified to include noise terms (Langevin dynamics, [21]); this, alongside other extensions to the dynamics to improve physical realism, may help with translating from simulation parameters to real molecules.

The physical implementation of the CHL learning itself requires three key parts. First is the ability to clamp components' volume fractions. In the surface model, an example clamping mechanism would be a programmable semi-permeable membrane designed to allow passage of only a specified subset of the components, with the ability for that subset to be changed over time. Second is the ability to modify interaction parameters $\chi_{ij}$ and/or reservoir potentials $\mu_{i,res}$. The locality condition – that $d\chi_{ij}/dt$ depends only on statistics about components $i$ and $j$ – eases the requirements on physical implementation. For example, as suggested by Stern et al. [67], if an $i-j$ mediator species' concentration induces an effective interaction $\tilde{G}_{ij}$ between components $i$ and $j$, production of that mediator via proximity-based ligation under mass-action kinetics could produce a Hebbian update of the form $d\tilde{G}_{ij}/dt \propto \phi_i(t)\phi_j(t)$, where $\phi_i(t)$ is the volume fraction at time $t$. Third is the switching between more- and less-clamped conditions with the associated switch between Hebbian and anti-Hebbian parameter updates. An implicit-memory mechanism has been proposed to enable contrastive switching between Hebbian and anti-Hebbian phases in molecular systems [25], but a clear clamping mechanism for implementing this approach has yet to be proposed. Advancements in these three key tasks and their integration could yield a generalized framework for purely physical learning that would allow exploration of high-dimensional LLPS beyond the limited accuracy of *in silico* models, and would cement energy-landscape sculpting as a physical mechanism for biomolecular learning.

## References

1 David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. doi:10.1207/S15516709COG0901_7.

2 Siddharth Agarwal, Dino Osmanovic, Mahdi Dizani, Melissa A Klocke, and Elisa Franco. Dynamic control of DNA condensation. *Nature Communications*, 15(1):1915, 2024.

3 Blaise Agüera y Arcas, Jyrki Alakuijala, James Evans, Ben Laurie, Alexander Mordvintsev, Eyvind Niklasson, Ettore Randazzo, and Luca Versari. Computational life: How well-formed, self-replicating programs emerge from simple interaction. *arXiv:2406.19108*, 2024.

4 Daniel J Amit, Hanoch Gutfreund, and Haim Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55(14):1530, 1985.

5 Daniel J Amit, Hanoch Gutfreund, and Haim Sompolinsky. Statistical mechanics of neural networks near saturation. *Annals of Physics*, 173(1):30–67, 1987.

6 Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

7 Jack E Baldwin and Hans Krebs. The evolution of metabolic cycles. *Nature*, 291(5814):381–382, 1981.

8 Salman F Banani, Hyun O Lee, Anthony A Hyman, and Michael K Rosen. Biomolecular condensates: organizers of cellular biochemistry. *Nature Reviews Molecular Cell Biology*, 18(5):285–298, 2017.

9 Naama Barkai and Stan Leibler. Robustness in simple biochemical networks. *Nature*, 387(6636):913–917, 1997.

10 MA Belyaev and AA Velichko. Classification of handwritten digits using the Hopfield network. In *IOP Conference Series: Materials Science and Engineering*, volume 862, page 052048. IOP Publishing, 2020.

11 James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL: `http://github.com/jax-ml/jax`.

12 Rodrigo Braz Teixeira, Giorgio Carugno, Izaak Neri, and Pablo Sartori. Liquid Hopfield model: Retrieval and localization in multicomponent liquid mixtures. *Proceedings of the National Academy of Sciences*, 121(48):e2320504121, 2024.

13 Jehoshua Bruck and Vwani P Roychowdhury. On the number of spurious memories in the Hopfield model (neural network). *IEEE Transactions on Information Theory*, 36(2):393–397, 1990.

14 Miguel A Carreira-Perpinan and Geoffrey Hinton. On contrastive divergence learning. In *International Workshop on Artificial Intelligence and Statistics*, pages 33–40. PMLR, 2005.

15 Aria S Chaderjian, Sam Wilken, and Omar A Saleh. Diverse, distinct, and densely packed DNA nanostar droplets. *Proceedings of the National Academy of Sciences*, 123(7):e2523462123, 2026.

16 Cameron Chalk, Salvador Buse, Krishna Shrinivas, Arvind Murugan, and Erik Winfree. Learning and inference in a lattice model of multicomponent condensates. In *30th International Conference on DNA Computing and Molecular Programming (DNA 30)*, volume 314, pages 5:1–5:24, 2024. `doi:10.4230/LIPIcs.DNA.30.5`.

17 Fan Chen and William M Jacobs. Programmable phase behavior in fluids with designable interactions. *The Journal of Chemical Physics*, 158(21), 2023.

18 Long Qing Chen and Jie Shen. Applications of semi-implicit Fourier-spectral method to phase field equations. *Computer Physics Communications*, 108(2-3):147–158, 1998.

19 Kevin M Cherry and Lulu Qian. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature*, 559(7714):370–376, 2018.

20 Kevin M Cherry and Lulu Qian. Supervised learning in DNA neural networks. *Nature*, 645(8081):639–647, 2025.

21 HE Cook. Brownian motion in spinodal decomposition. *Acta Metallurgica*, 18(3):297–306, 1970.

22 David Doty, Jack H Lutz, Matthew J Patitz, Robert T Schweller, Scott M Summers, and Damien Woods. The tile assembly model is intrinsically universal. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 302–310. IEEE, 2012. `doi:10.1109/FOCS.2012.76`.

**23** Constantine G Evans and Erik Winfree. Physical principles for DNA tile self-assembly. *Chemical Society Reviews*, 46(12):3808–3829, 2017.

**24** Constantine Glen Evans, Jackson O'Brien, Erik Winfree, and Arvind Murugan. Pattern recognition in the nucleation kinetics of non-equilibrium self-assembly. *Nature*, 625(7995):500–507, 2024.

**25** Martin J Falk, Adam T Strupp, Benjamin Scellier, and Arvind Murugan. Temporal contrastive learning through implicit non-equilibrium memory. *Nature Communications*, 16(1):2163, 2025.

**26** Paul J Flory. Thermodynamics of high polymer solutions. *The Journal of Chemical Physics*, 10(1):51–61, 1942.

**27** Mark E Fornace, Jining Huang, Cody T Newman, Avinash Nanjundiah, Nicholas J Porubsky, Marshall B Pierce, and Niles A Pierce. NUPACK: Computational nucleic acid analysis and design. *ACS Synthetic Biology*, 15(4):1426–1441, 2026.

**28** Alireza Goudarzi, Matthew R Lakin, and Darko Stefanovic. DNA reservoir computing: a novel molecular computing approach. In *International Workshop on DNA-Based Computers*, pages 76–89. Springer, 2013. doi:10.1007/978-3-319-01928-4_6.

**29** Geoffrey E Hinton. Deterministic Boltzmann learning performs steepest descent in weight-space. *Neural Computation*, 1(1):143–150, 1989. doi:10.1162/NECO.1989.1.1.143.

**30** Pierre C Hohenberg and Bertrand I Halperin. Theory of dynamic critical phenomena. *Reviews of Modern Physics*, 49(3):435, 1977.

**31** John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

**32** John J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10):3088–3092, 1984.

**33** John J Hopfield and David W Tank. "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52(3):141–152, 1985.

**34** Maurice L Huggins. Solutions of long chain compounds. *The Journal of Chemical Physics*, 9(5):440–440, 1941.

**35** Anthony A Hyman, Christoph A Weber, and Frank Jülicher. Liquid-liquid phase separation in biology. *Annual Review of Cell and Developmental Biology*, 30:39–58, 2014.

**36** William M Jacobs and Daan Frenkel. Predicting phase behavior in multicomponent mixtures. *The Journal of Chemical Physics*, 139(2):024108, 2013.

**37** William M Jacobs and Daan Frenkel. Phase transitions in biological systems with many components. *Biophysical Journal*, 112(4):683–691, 2017.

**38** Byoung-jin Jeon, Dan T Nguyen, and Omar A Saleh. Sequence-controlled adhesion and microemulsification in a two-phase system of DNA liquid droplets. *The Journal of Physical Chemistry B*, 124(40):8888–8895, 2020.

**39** Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

**40** Patrick Kidger. *On Neural Differential Equations*. PhD thesis, University of Oxford, 2021.

**41** Shigeru Kondo and Takashi Miura. Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, 329(5999):1616–1620, 2010.

**42** Matthew R Lakin and Darko Stefanovic. Supervised learning in adaptive DNA strand displacement networks. *ACS Synthetic Biology*, 5(8):885–897, 2016.

**43** Inhoo Lee, Salvador Buse, and Erik Winfree. Differentiable programming of indexed chemical reaction networks and reaction-diffusion systems. In *31st International Conference on DNA Computing and Molecular Programming (DNA 31)*, pages 4:1–4:23, 2025. doi:10.4230/LIPIcs.DNA.31.4.

**44** Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

**45** Fritz Lipmann. The ATP–phosphate cycle. In *Current Topics in Cellular Regulation*, volume 18, pages 301–311. Elsevier, 1981.

**46** Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv:1711.05101*, 2017.

**47** Sheng Mao, Derek Kuldinow, Mikko P Haataja, and Andrej Košmrlj. Phase behavior and morphology of multicomponent liquid mixtures. *Soft Matter*, 15(6):1297–1311, 2019.

**48** Javier R Movellan. Contrastive Hebbian learning in the continuous Hopfield model. In *Connectionist Models: Proceedings of the 1990 Summer School*, pages 10–17. Morgan Kaufmann, 1991.

**49** Dan T Nguyen, Byoung-jin Jeon, Gabrielle R Abraham, and Omar A Saleh. Length-dependence and spatial structure of DNA partitioning into a DNA liquid. *Langmuir*, 35(46):14849–14854, 2019.

**50** Shu Okumura, Guillaume Gines, Nicolas Lobato-Dauzier, Alexandre Baccouche, Robin Deteix, Teruo Fujii, Yannick Rondelez, and Anthony J Genot. Nonlinear decision-making with enzymatic neural networks. *Nature*, 610(7932):496–501, 2022.

**51** Jacob Parres-Gold, Matthew Levine, Benjamin Emert, Andrew Stuart, and Michael B Elowitz. Contextual computation by competitive protein dimerization networks. *Cell*, 188(7):1984–2002, 2025.

**52** Lulu Qian, David Soloveichik, and Erik Winfree. Efficient Turing-universal computation with DNA polymers. In *International Workshop on DNA-Based Computers*, pages 123–140. Springer, 2010. doi:10.1007/978-3-642-18305-8_12.

**53** Lulu Qian, Erik Winfree, and Jehoshua Bruck. Neural network computation with DNA strand displacement cascades. *Nature*, 475(7356):368–372, 2011. doi:10.1038/NATURE10262.

**54** Yicheng Qiang, Chengjie Luo, and David Zwicker. Scaling laws for phase coexistence in multicomponent mixtures. *Physical Review Research*, 7(4):043008, 2025.

**55** Yusuke Sato, Tetsuro Sakamoto, and Masahiro Takinoue. Sequence-based engineering of dynamic functions of micrometer-sized DNA droplets. *Science Advances*, 6(23):eaba3471, 2020.

**56** Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in Computational Neuroscience*, 11:24, 2017. doi:10.3389/FNCOM.2017.00024.

**57** Nadrian C. Seeman. *Structural DNA Nanotechnology*. Cambridge University Press, Cambridge, United Kingdom, 2016.

**58** Ludovica Serricchio, Dario Bocchi, Claudio Chilin, Raffaele Marino, Matteo Negri, Chiara Cammarota, and Federico Ricci-Tersenghi. Daydreaming Hopfield networks and their surprising effectiveness on correlated data. *Neural Networks*, 186:107216, 2025. doi:10.1016/J.NEUNET.2025.107216.

**59** Krishna Shrinivas and Michael P Brenner. Multiphase coexistence capacity in complex fluids. *bioRxiv:2022.10.19.512909*, 2022.

**60** Krishna Shrinivas and Michael P Brenner. Phase separation in fluids with many interacting components. *Proceedings of the National Academy of Sciences*, 118(45):e2108551118, 2021.

**61** Krishna Shrinivas, Benjamin R Sabari, Eliot L Coffey, Isaac A Klein, Ann Boija, Alicia V Zamudio, Jurian Schuijers, Nancy M Hannett, Phillip A Sharp, Richard A Young, et al. Enhancer features that drive formation of transcriptional condensates. *Molecular Cell*, 75(3):549–561, 2019.

**62** Friedrich C Simmel, Bernard Yurke, and Hari R Singh. Principles and applications of nucleic acid strand displacement reactions. *Chemical Reviews*, 119(10):6326–6369, 2019.

**63** Karuna Skipper and Shelley FJ Wickham. DNA nanostars that self-assemble into core–shell condensate microdroplets. *Nanoscale Horizons*, 2026.

**64** David Soloveichik, Georg Seelig, and Erik Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398, 2010.

**65** Tianqi Song and Lulu Qian. Heat-rechargeable computation in DNA logic circuits and neural networks. *Nature*, 646(8084):315–322, 2025.

**66** Ulrich Stelzl, Uwe Worm, Maciej Lalowski, Christian Haenig, Felix H Brembeck, Heike Goehler, Martin Stroedicke, Martina Zenkner, Anke Schoenherr, Susanne Koeppen, et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122(6):957–968, 2005.

**67** Menachem Stern and Arvind Murugan. Learning without neurons in physical systems. *Annual Review of Condensed Matter Physics*, 14(1):417–441, 2023.

**68** Jonathan Stricker, Tobias Falzone, and Margaret L Gardel. Mechanics of the F-actin cytoskeleton. *Journal of Biomechanics*, 43(1):9–14, 2010.

**69** Diana A Tanase, Dino Osmanović, Roger Rubio-Sánchez, Layla Malouf, Elisa Franco, and Lorenzo Di Michele. Internal phase separation in synthetic DNA condensates. *Advanced Science*, 12(41):e06275, 2025.

**70** Rodrigo Braz Teixeira, Izaak Neri, and Pablo Sartori. Metastable phase separation and information retrieval in multicomponent mixtures. *arXiv:2509.10705*, 2025.

**71** Gašper Tkačik and William Bialek. Information processing in living systems. *Annual Review of Condensed Matter Physics*, 7(1):89–117, 2016.

**72** Hirotake Udono, Jing Gong, Yusuke Sato, and Masahiro Takinoue. DNA droplets: intelligent, dynamic fluid. *Advanced Biology*, 7(3):2200180, 2023.

**73** Jan van den Berg. The most general framework of continuous Hopfield neural networks. In *Proceedings of International Workshop on Neural Networks for Identification, Control, Robotics and Signal/Image Processing*, pages 92–100. IEEE, 1996.

**74** Jan van den Berg. Generalized Hopfield networks for constrained optimization. Technical Report EUR-FEW-CS-98-04, Erasmus School of Economics, Erasmus University Rotterdam, 1998. URL: `http://hdl.handle.net/1765/761`.

**75** Christian Waltermann and Edda Klipp. Information theory based approaches to cellular signaling. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1810(10):924–932, 2011.

**76** Erik Winfree. Algorithmic self-assembly of DNA: Theoretical motivations and 2D assembly experiments. *Journal of Biomolecular Structure and Dynamics*, 17(sup1):263–270, 2000.

**77** Erik Winfree, Xiaoping Yang, and Nadrian C Seeman. Universal computation via self-assembly of DNA: Some theory and experiments. In *DNA Based Computers II*, volume 44 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–213. American Mathematical Society, 1998.

**78** Damien Woods, David Doty, Cameron Myhrvold, Joy Hui, Felix Zhou, Peng Yin, and Erik Winfree. Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature*, 567(7748):366–372, 2019. `doi:10.1038/S41586-019-1014-9`.

**79** Xiaohui Xie and H Sebastian Seung. Equivalence of backpropagation and contrastive Hebbian learning in a layered network. *Neural Computation*, 15(2):441–454, 2003. `doi:10.1162/089976603762552988`.

**80** Wataru Yahiro, Nathanael Aubert-Kato, and Masami Hagiya. A reservoir computing approach for molecular computing. In *Artificial Life Conference Proceedings*, pages 31–38. MIT Press, 2018. `doi:10.1162/ISAL_A_00013`.

**81** Aidan Zentner, Ethan V Halingstad, Cameron Chalk, Michael P Brenner, Arvind Murugan, Erik Winfree, and Krishna Shrinivas. Information processing driven by multicomponent surface condensates. *arXiv:2509.08100*, 2025.

**82** David Yu Zhang and Georg Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature Chemistry*, 3(2):103–113, 2011.

**83** David Zwicker and Liedewij Laan. Evolved interactions stabilize many coexisting phases in multicomponent liquids. *Proceedings of the National Academy of Sciences*, 119(28):e2201250119, 2022.

# A Appendix

## A.1 Methods for numerical integration and training

Details of the numerical methods we used to integrate model A and model B dynamics and train the parameters are described below. Our simulations are written in JAX, use Diffrax solvers [11, 40], and follow integration schemes inspired by prior studies of multicomponent phase separation [47, 60, 81].

Our parameter initializations and training configurations are as follows. For Figure 2 (memory retrieval), the interaction matrix is initialized with a Hebbian product over target phase compositions, and the chemical potentials are initialized as 0. We train with a learning rate of 0.1 for 500-1000 epochs. For Figures 3 and 4 (classification), the interaction matrix is initialized by sampling uniformly in [-10, 10], with diagonal terms, interactions between the input components, and all the solvent interactions set to 0 and kept at 0 throughout. The chemical potentials are initialized at 0, and input and solvent chemical potentials are kept at 0 throughout. We use a learning rate of 0.02, with 300–500 samples per epoch for 500–2000 epochs. For Figure 5, we use a learning rate of 0.1 for 25–100 epochs. The interaction matrix is again initialized with a Hebbian rule.

In all cases, the parameters are updated according to the CHL rule derived in the main text. The actual parameter $\theta$ updates in each epoch are the average $\frac{\partial J}{\partial \theta}$ gradients over the samples in that epoch, multiplied by a discrete scaling factor (i.e., stochastic gradient descent). The AdamW optimizer from Optax [11] is used throughout to improve training stability [46], and $\chi_{ij}$ interactions are clipped to a maximum absolute value of 25.

All differential equations are solved numerically with finite difference methods, using discretized time, and also discretized space in the PDE case. Mobility $M = 1.0$ is used for all simulations. For the spatial simulations, we use $\kappa = 5.0$. The spatial simulations are determined to be converged and halted when the maximum step-to-step absolute change of all components at all positions is below $10^{-6}$, or the number of simulation steps exceeds $2 \times 10^5$. Source code and notebooks are available on GitHub.

## A.2 Model A dynamics in surface model with Lagrange multipliers

The single-compartment surface model treats the multicomponent liquid as a well-mixed system described by a composition vector $\vec{\phi}$ and the Flory-Huggins energy $f_{FH}(\vec{\phi})$. Standard model A dynamics [30] take the form $\frac{d\phi_i}{dt} = -M_i \frac{\partial f_{FH}}{\partial \phi_i}$, where $M_i$ is the mobility and for simplicity we choose $M_i = M$, a positive constant for all species. To enforce the constraint that $\sum_{i=1}^{n} \phi_i = 1$ while preserving the gradient-flow structure of the dynamics, we introduce a Lagrange multiplier $\lambda$ and the modified free energy $f_{\lambda FH} = f_{FH} + \lambda(\sum_{i=1}^{n} \phi_i - 1)$, with $\lambda$ determined dynamically to ensure the constraint (in which case $f_{\lambda FH} = f_{FH}$). The modified dynamics then become

$$\frac{d\phi_i}{dt} = -M \frac{\partial f_{\lambda FH}}{\partial \phi_i} = -M(\mu_{i,FH} + \lambda) \tag{14}$$

where we define $\mu_{i,FH} = \frac{\partial f_{FH}}{\partial \phi_i} = 1 + \log \phi_i + \sum_{j=1}^{n} \chi_{ij}\phi_j - \mu_{i,res}$ as the chemical potential of species $i$. Applying the constraint $\sum_{i=1}^{n} \frac{d\phi_i}{dt} = 0$, we can solve for $\lambda$:

$$\lambda = -\frac{1}{n} \sum_{i=1}^{n} \mu_{i,FH} = -\langle \mu_{FH} \rangle \ . \tag{15}$$

These dynamics guarantee a monotonic decrease of the free energy:

$$\begin{aligned}\frac{df_{\lambda FH}}{dt} &= \sum_{i=1}^{n} \frac{\partial f_{\lambda FH}}{\partial \phi_i} \cdot \frac{d\phi_i}{dt} = -\sum_{i=1}^{n} (\mu_{i,FH} - \langle \mu_{FH} \rangle) \cdot M(\mu_{i,FH} - \langle \mu_{FH} \rangle) \\ &= -M \sum_{i=1}^{n} (\mu_{i,FH} - \langle \mu_{FH} \rangle)^2 \leq 0 \ .\end{aligned} \tag{16}$$

The energy converges only when the system reaches the equilibrium condition $\frac{d\phi_i}{dt} = 0$, i.e., $\mu_{i,FH} = \langle \mu_{FH} \rangle$ or $\mu_{i,FH} + \lambda = 0$.

For numerical implementation, the state of the system at $t$ is captured by $\vec{\phi}^t = (\phi_1, \phi_2, \ldots, \phi_n)^t$, and the total system volume $v = 1$ in dimensionless units. Our model A dynamics occur in a space of transformed variables $x_i = \log \phi_i$, with $\phi_i = e^{x_i}$ recovered at each step, following Zentner et al. [81]. This logarithmic reparameterization enforces positivity and improves numerical stability near the boundary of the simplex. We perform numerical integration using the Dopri5 adaptive 4th/5th-order Runge-Kutta solver from Diffrax.

## A.3 CHL rule for reservoir potential in the single-compartment case

We derive the CHL update rule for the reservoir chemical potential $\mu_{i,res}$, following the approach used for learning the interaction matrix $\chi_{ij}$ described in the main text. Recall that the contrastive objective is $J = \check{f}_{FH}^{(+)} - \check{f}_{FH}^{(-)}$, and so we seek $\partial J / \partial \mu_{i,res}$ to learn $\mu_{i,res}$.

We begin by expanding the derivative of the modified free energy with respect to $\mu_{i,res}$:

$$\frac{\partial \check{f}_{FH}}{\partial \mu_{i,res}} = \frac{\partial (\sum_{k,l=1}^{n} \frac{1}{2} \check{\phi}_k \chi_{kl} \check{\phi}_l + \sum_{k=1}^{n} \check{\phi}_k \log \check{\phi}_k - \sum_{k=1}^{n} \mu_{k,res} \check{\phi}_k)}{\partial \mu_{i,res}} \ . \tag{17}$$

Differentiating the free energy density gives

$$\frac{\partial \check{f}_{FH}}{\partial \mu_{i,res}} = -\check{\phi}_i + \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \mu_{i,res}} (1 + \log \check{\phi}_k + \sum_{l=1}^{n} \chi_{kl} \check{\phi}_l - \mu_{k,res}) = -\check{\phi}_i + \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \mu_{i,res}} \mu_{k,FH} \ . \tag{18}$$

As before, we note that at equilibrium $\mu_{k,FH} = \langle \mu_{FH} \rangle$ and by differentiating the constraint $\sum_{k=1}^{n} \check{\phi}_k = 1$ to obtain $\sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \mu_{i,res}} = 0$, we can see that

$$\frac{\partial \check{f}_{FH}}{\partial \mu_{i,res}} = -\check{\phi}_i \ . \tag{19}$$

The gradient of the contrastive objective is then simply

$$\frac{\partial \check{J}}{\partial \mu_{i,res}} = \frac{\partial \check{f}_{FH}^{(+)}}{\partial \mu_{i,res}} - \frac{\partial \check{f}_{FH}^{(-)}}{\partial \mu_{i,res}} = -\check{\phi}_i^{(+)} + \check{\phi}_i^{(-)} \ , \tag{20}$$

and the CHL update rule for the reservoir chemical potential is $\Delta \mu_{i,res} \propto \check{\phi}_i^{(+)} - \check{\phi}_i^{(-)}$.

This has the same Hebbian structure as the $\chi_{ij}$ update rule: the reservoir potential for species $i$ is increased when the target (clamped) composition of that species exceeds its free equilibrium value, pulling the energy landscape toward the desired attractor.

## A.4 Spatial phase separation dynamics with Lagrange multipliers

Similar to the surface model case, a Lagrange multiplier $\lambda(\mathbf{x},t)$ is introduced to enforce the local normalization constraint $\sum_{i=1}^{n}\phi_i(\mathbf{x},t)=1$. The modified free-energy density is

$$\begin{aligned} f_{\lambda CH} &= f_{CH} + \lambda\left(\sum_{i=1}^{n}\phi_i - 1\right) \\ &= \frac{1}{2}\sum_{i,j=1}^{n}\phi_i\chi_{ij}\phi_j + \sum_{i=1}^{n}\phi_i\log\phi_i + \sum_{i=1}^{n}\frac{\kappa}{2}\left|\nabla\phi_i\right|^2 + \lambda\left(\sum_{i=1}^{n}\phi_i - 1\right)\ . \end{aligned} \tag{21}$$

The modified chemical potential of species $i$ is then

$$\mu_{i,\lambda CH} = \frac{\partial f_{\lambda CH}}{\partial\phi_i} = \frac{\partial f_{CH}}{\partial\phi_i} + \frac{\partial\lambda\left(\sum_{i=1}^{n}\phi_i - 1\right)}{\partial\phi_i} = \mu_{i,CH} + \lambda\ . \tag{22}$$

Applying the constraint $\sum_{i=1}^{n}\phi_i = 1$, which requires $\sum_{i=1}^{n}\frac{\partial\phi_i}{\partial t} = 0$, we obtain

$$\sum_{i=1}^{n}\frac{\partial\phi_i}{\partial t} = M\sum_{i=1}^{n}\nabla^2\mu_{i,\lambda CH} = M\sum_{i=1}^{n}\nabla^2(\mu_{i,CH}+\lambda) = M\nabla^2\left(\sum_{i=1}^{n}\mu_{i,CH} + n\lambda\right) = 0\ . \tag{23}$$

Under periodic or no-flux boundary conditions, this reduces to

$$\sum_{i=1}^{n}\mu_{i,CH} + n\lambda = \gamma\ . \tag{24}$$

Note that any choice of constant $\gamma$ yields the same dynamics, since it is spatially uniform and vanishes under the Laplacian $\nabla^2$. The choice $\gamma = 0$ is therefore made without loss of generality, allowing us to solve for $\lambda$:

$$\lambda = \frac{-\sum_{i=1}^{n}\mu_{i,CH}}{n} = -\langle\mu_{CH}\rangle\ . \tag{25}$$

Here $\lambda(\mathbf{x},t)$ is a local field, not a global constant. In the discretized implementation, $\lambda$ is computed cell-by-cell from the local mean chemical potential, ensuring the constraint holds pointwise throughout the simulation. Our modified model B dynamics become

$$\frac{\partial\phi_i}{\partial t} = M\nabla^2\mu_{i,\lambda CH} = M\nabla^2(\mu_{i,CH} - \langle\mu_{CH}\rangle)\ . \tag{26}$$

These dynamics also guarantee monotonic decrease of the total free energy:

$$\begin{aligned} \frac{dF_{\lambda CH}}{dt} &= \int_v\sum_{i=1}^{n}\frac{\delta F_{\lambda CH}}{\delta\phi_i}\frac{\partial\phi_i}{\partial t}d\mathbf{x} = \int_v\sum_{i=1}^{n}\mu_{i,\lambda CH}M\nabla^2\mu_{i,\lambda CH}d\mathbf{x} \\ &= -\int_v\sum_{i=1}^{n}\nabla\mu_{i,\lambda CH}M\nabla\mu_{i,\lambda CH}d\mathbf{x} + \int_{\partial\Omega}\sum_{i=1}^{n}\mu_{i,\lambda CH}M\nabla\mu_{i,\lambda CH}\cdot\mathbf{n}d\Omega\ . \end{aligned} \tag{27}$$

With periodic boundary conditions $\int_{\partial\Omega}\sum_{i=1}^{n}\mu_{i,\lambda CH}M\nabla\mu_{i,\lambda CH}\cdot\mathbf{n}d\Omega = 0$, and

$$\frac{dF_{\lambda CH}}{dt} = -\int_v\sum_{i=1}^{n}\nabla\mu_{i,\lambda CH}\ M\nabla\mu_{i,\lambda CH}d\mathbf{x} \le 0 \tag{28}$$

for all species with $\phi_i > 0$. At equilibrium, we obtain $\nabla\mu_{i,\lambda CH} = 0$, which means $\mu_{i,\lambda CH}$ is spatially uniform:

$$\mu_{i,\lambda CH} = \mu_{i,CH} - \langle\mu_{CH}\rangle = C_k\ , \tag{29}$$

where $C_k$ is a global constant, independent of position $\mathbf{x}$.

## A.5 CHL rule for spatial phase separation

Here we derive the CHL rule for the spatial case presented in Section 3. First, taking the derivative of $\check{F}_{CH}$, the total energy at equilibrium, we have

$$\begin{aligned}\frac{\partial \check{F}_{CH}}{\partial \chi_{ij}} &= \frac{\partial}{\partial \chi_{ij}} \int_v \left( \sum_{k,l=1}^{n} \frac{1}{2} \check{\phi}_k \chi_{kl} \check{\phi}_l + \sum_{k=1}^{n} \check{\phi}_k \log \check{\phi}_k + \sum_{k=1}^{n} \frac{\kappa}{2} \left| \nabla \check{\phi}_k \right|^2 \right) d\mathbf{x} \\ &= \int_v \left( \check{\phi}_i \check{\phi}_j + \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}} \mu_{k,CH} \right) d\mathbf{x}.\end{aligned} \tag{30}$$

At equilibrium, $\mu_{k,CH} - \langle \mu_{CH} \rangle = C_k$, giving us

$$\frac{\partial \check{F}_{CH}}{\partial \chi_{ij}} = \int_v \left( \check{\phi}_i \check{\phi}_j + \langle \mu_{CH} \rangle \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}} + \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}} C_k \right) d\mathbf{x}. \tag{31}$$

The term $\langle \mu_{CH} \rangle \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}}$ vanishes because of the constraint $\sum_{k=1}^{n} \partial \check{\phi}_k = 0$. Exchanging integral and sum, and taking $C_k$ outside the integral,

$$\int_v \sum_{k=1}^{n} \frac{\partial \check{\phi}_k}{\partial \chi_{ij}} C_k d\mathbf{x} = \sum_{k=1}^{n} C_k \int_v \frac{\partial \check{\phi}_k}{\partial \chi_{ij}} d\mathbf{x}. \tag{32}$$

Since $\int_v \check{\phi}_k d\mathbf{x}$ is conserved for each component, we have $\int_v \frac{\partial \check{\phi}_k}{\partial \chi_{ij}} d\mathbf{x} = 0$. The derivative becomes:

$$\frac{\partial \check{F}_{CH}}{\partial \chi_{ij}} = \int_v \check{\phi}_i \check{\phi}_j d\mathbf{x}. \tag{33}$$

The gradient of the contrastive objective is therefore

$$\frac{\partial \check{J}}{\partial \chi_{ij}} = \frac{\partial \check{F}^{(+)}_{CH}}{\partial \chi_{ij}} - \frac{\partial \check{F}^{(-)}_{CH}}{\partial \chi_{ij}} = \int_v \check{\phi}^{(+)}_i \check{\phi}^{(+)}_j d\mathbf{x} - \int_v \check{\phi}^{(-)}_i \check{\phi}^{(-)}_j d\mathbf{x}, \tag{34}$$

and the spatial CHL update rule is

$$\Delta \chi_{ij} \propto - \int_v \check{\phi}^{(+)}_i \check{\phi}^{(+)}_j d\mathbf{x} + \int_v \check{\phi}^{(-)}_i \check{\phi}^{(-)}_j d\mathbf{x}. \tag{35}$$

## A.6 Implicit-explicit integration of spatial phase separation

The spatial model tracks the composition field $\vec{\phi}(\mathbf{x}, t) = (\phi_1, \phi_2, \ldots, \phi_n)$ at every point on a 2D grid with periodic boundaries, with the local constraint $\sum_{i=1}^{n} \phi_i(\mathbf{x}, t) = 1$ enforced at each grid cell independently. No chemical reservoir is included. We use a square grid of size $(nx, ny)$ with grid spacing $(dx, dy)$, taking $nx = ny = 128$ and $dx = dy = 1.0$ here.

The total energy of the system is $F_{CH}$, and its variational derivative with respect to $\phi_i(\mathbf{x})$ gives the local chemical potential:

$$\mu_{i,CH}(\mathbf{x}, t) = \frac{\delta F_{CH}}{\delta \phi_i(\mathbf{x}, t)} = f'_{i,FH} - \kappa \nabla^2 \phi_i = \sum_{j=1}^{n} \chi_{ij} \phi_j + 1 + \log \phi_i - \kappa \nabla^2 \phi_i \ , \tag{36}$$

where $f'_{i,FH} = \partial f_{FH} / \partial \phi_i$ is the derivative of the bulk free energy density. The Laplacian term $-\kappa \nabla^2 \phi_i$ penalizes spatial curvature and gives rise to the interfacial energy.

The local sum-to-unity constraint is enforced by introducing the Lagrange multiplier, as described in the previous section. Recalling that our Lagrange-modified model B dynamics are $\frac{\partial \phi_i}{\partial t} = M\nabla^2 \mu_{i,\lambda CH} = M\nabla^2(\mu_{i,CH} - \langle \mu_{CH} \rangle)$, we can substitute the expression for $\mu_{i,CH}$, to obtain our PDE:

$$\frac{\partial \phi_i}{\partial t} = M\nabla^2 f'_{i,FH} - M\kappa\nabla^4\phi_i - M\nabla^2\langle \mu_{CH} \rangle \ . \tag{37}$$

We solve this PDE using a semi-implicit spectral method. Taking the Fourier transform, spatial derivatives map as $\nabla^2 \to -k^2$ and $\nabla^4 \to k^4$, where $k = |\mathbf{k}|$ is the wavenumber magnitude, we have

$$\frac{\partial \widehat{\phi}_i}{\partial t} = -Mk^2 \left[ \left( \widehat{f}'_{i,FH} + \kappa k^2 \widehat{\phi}_i \right) - \langle \widehat{\mu}_{CH} \rangle \right] = -Mk^2 \left( \widehat{f}'_{i,FH} - \langle \widehat{\mu}_{CH} \rangle \right) - M\kappa k^4 \widehat{\phi}_i \ , \tag{38}$$

where $\widehat{f}'_{i,FH}$ denotes the Fourier transform of the nonlinear bulk chemical potential. The $M\kappa k^4$ term is stiff at large wavenumbers – requiring prohibitively small time steps if treated explicitly – so it is treated implicitly, while the nonlinear terms are treated explicitly:

$$\frac{\widehat{\phi}_i^{t+1} - \widehat{\phi}_i^t}{\Delta t} = -Mk^2 \left( (\widehat{f}'_{i,FH})^t - \langle \widehat{\mu}_{CH} \rangle^t \right) - M\kappa k^4 \widehat{\phi}_i^{t+1} \ . \tag{39}$$

When the nonlinear bulk term $\widehat{f}'_{i,FH}$ varies rapidly, the basic semi-implicit scheme can still become unstable. To improve robustness, an r-stabilization scheme is employed [18]. A linear term $r\phi_i$ is simultaneously added and subtracted, shifting part of the nonlinear contribution into the implicit part without altering the equation:

$$\frac{\partial \phi_i}{\partial t} = M\nabla^2 \left( f'_{i,FH} - r\phi_i \right) + M\nabla^2(r\phi_i) - M\kappa\nabla^4\phi_i - M\nabla^2\langle \mu_{CH} \rangle \ . \tag{40}$$

The first term is treated explicitly and the remaining linear terms implicitly. In Fourier space, this becomes

$$\frac{\partial \widehat{\phi}_i}{\partial t} = -Mk^2 \left( \widehat{f}'_{i,FH} - r\widehat{\phi}_i - \langle \widehat{\mu}_{CH} \rangle \right) - M(rk^2\widehat{\phi}_i + \kappa k^4 \widehat{\phi}_i) \ . \tag{41}$$

The semi-implicit update is then

$$\widehat{\phi}_i^{t+1} = \frac{\widehat{\phi}_i^t - \Delta t M k^2 \left( (\widehat{f}'_{i,FH})^t - r\widehat{\phi}_i^t - \langle \widehat{\mu}_{CH} \rangle^t \right)}{1 + \Delta t M(rk^2 + \kappa k^4)} \ . \tag{42}$$

The stabilization parameter $r = 2.0$ is used in all spatial simulations. An adaptive time step $\Delta t$ is used to avoid simulation artifacts, guaranteeing monotonic decrease of the total energy and non-negative volume fractions throughout the simulation.

## A.7 Trained interaction parameters

Here we provide the parameters ($\chi_{ij}$, $\mu_i$) obtained by CHL for the low-dimensional input-output tasks described in Section 2.3, and the MNIST task of Section 2.4.

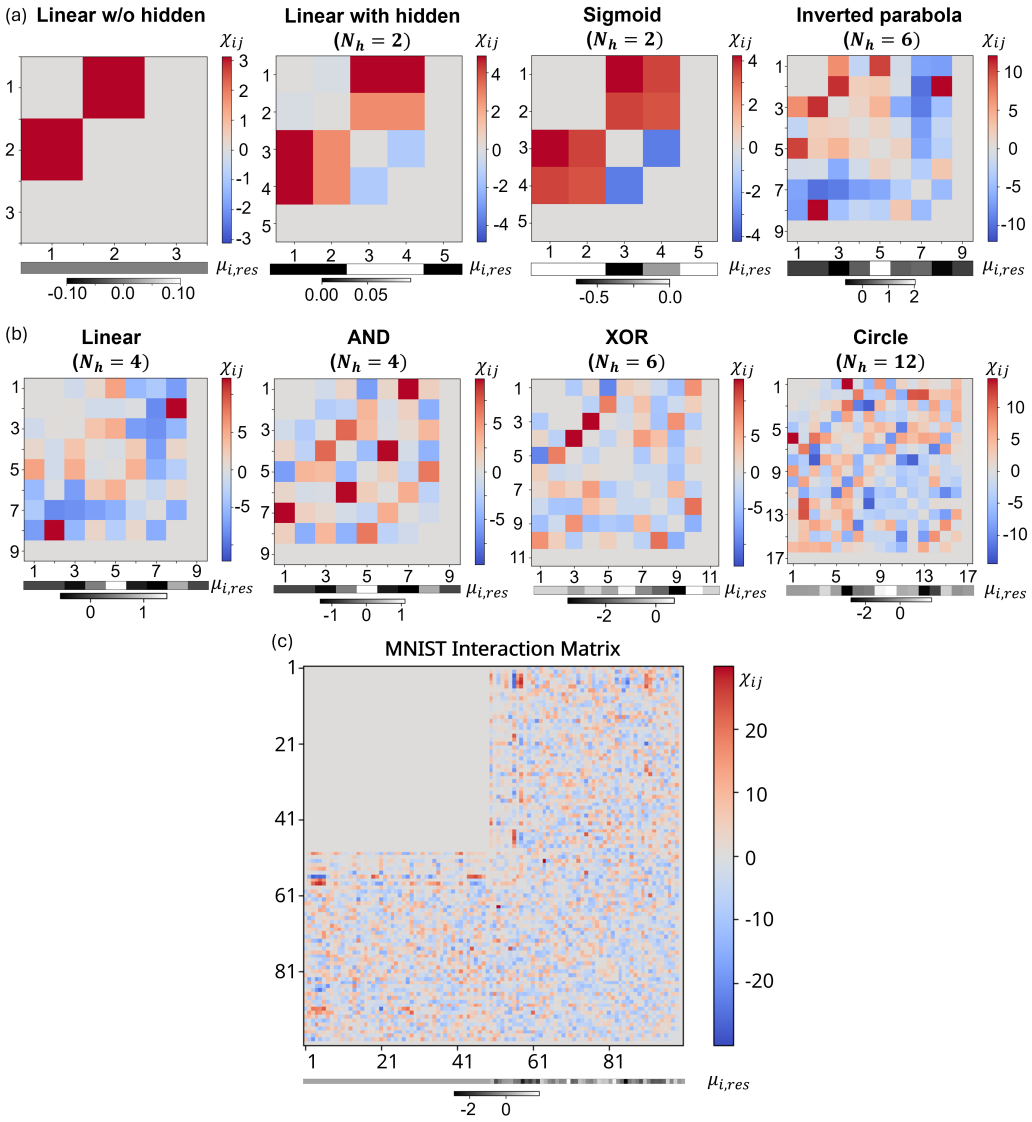


**Figure 7** Trained parameters for low-dimensional input-output functions in Figure 3. (a) Trained interaction matrices $\chi_{ij}$ and reservoir potentials $\mu_{i,res}$ for 1-input 1-output analog relations. Beyond linear cases, we need to add hidden components to expand the accessible energy landscape and solve these tasks. (b) Trained interaction matrices and reservoir potentials for 2-input 2-output classification tasks. The structure of $\chi_{ij}$ reflects the classification boundary: stronger repulsive interactions between input and output components encode the boundary, while hidden components mediate nonlinear separations. The last rows and columns indicate the solvent component. (c) Trained interaction matrix and reservoir potentials for the MNIST dataset.