PNAS

Supporting Information for

Combinatorial decision-making driven by multicomponent surface condensates.

Aidan Zentner, Ethan V. Halingstad, Cameron Chalk, Michael P. Brenner, Arvind Murugan, Erik Winfree, Krishna Shrinivas

Krishna Shrinivas
Email: krishna@northwestern.edu

This PDF file includes:

Supplementary Note 1: Model A Dynamics

Deriving Mean-Field Dynamics

In this section, we derive the mean-field dynamics for an effective interaction matrix χ and reservoir potential $\vec{\mu}_{\text{res}}$ used in the manuscript. For simplicity in deriving these dynamics, we make no distinction between input, output, and hidden species, and we assume that inputs can also exchange with the reservoir; we relax this assumption at the end of the derivation by setting their mobilities to 0. Accordingly, the surface exchanges with an infinite reservoir held at a chemical potential vector $\vec{\mu}''_{\text{res}}$. To accommodate indexing the solvent when required, we extend objects ($\vec{\phi}$, χ, $\vec{\mu}''_{\text{res}}$) to have an additional 0 index to denote the solvent, such that when the solvent is included as an explicit variable, we index from 0 to N rather than from 1 to N. We therefore define the length $N+1$ chemical potential vector $\vec{\mu}'_{\text{res}} \equiv 0 \circ \vec{\mu}^{(\text{in})}_{\text{res}} \circ \vec{\mu}_{\text{res}}$, where $\mu'_{\text{res},i} = \mu''_{\text{res},i} - \mu''_{\text{res},0}$ is the reservoir potential of species i relative to the solvent. Note that the input chemical potential $\vec{\mu}^{(\text{in})}_{\text{res}}$ is irrelevant to the dynamics since these species do not exchange with the reservoir. Written in this way, the vector $\vec{\mu}_{\text{res}}$ is the same as in the manuscript.

When the concentration vector $\vec{\phi}$ is treated as a function of space, the Landau-Ginzburg Hamiltonian describes the effective free energy of the surface as

$$\beta\mathcal{H} = \int dV \left[\Omega_{\text{G}}\left(\vec{\phi},\chi\right) + \frac{\kappa}{2}\left(\nabla\vec{\phi}\right)^2\right] \tag{S1}$$

where $\left(\nabla\vec{\phi}\right)^2 = \sum_{i=0}^{N}\sum_{n=1}^{d}\left(\partial_{x_n}\phi_i\right)^2$. The κ term penalizes spatial gradients in the homogeneous system. The grand-potential Ω_{G} describes an effective free-energy for the surface exchanging with an infinitely large reservoir and includes a Lagrange-multiplier term that enforces overall conservation of mass. For an open system at some initial composition, the relaxation to steady-state is driven by an exchange of species without conserving counts. Near equilibrium, Model A dynamics (1) characterizes these relaxation dynamics as purely downhill: the decrease in the overall free energy of the system is, to a first approximation, driven by linear gradients of the free energy with respect to the system's composition. Since we assume our surface remains well-mixed, we neglect the contributions from spatial gradients. We also neglect the interfacial energy between the surface volume and the reservoir, assuming that the free-energy is dominated by bulk contributions. Thus, the temporal evolution of the average volume fraction ϕ_i of species i within the system can be written as

$$\frac{\partial\phi_i(t)}{\partial t} = -\sum_{j=0}^{N} D'_{ij}\frac{\delta(\beta\mathcal{H})}{\delta\phi_j} + \eta_i(t) \tag{S2}$$

$$= -\sum_{j=0}^{N} D'_{ij}\frac{\partial\Omega_{\text{G}}}{\partial\phi_j} + \eta_i(t) \tag{S3}$$

where $\mathbf{D}'$ is the mobility matrix, again with index 0 corresponding to the solvent, that sets the rate of exchange between the system and reservoir. The above equation reflects the fact that, rather than purely decreasing energies, Model A dynamics also explicitly permits modeling the effect of temporally uncorrelated thermal fluctuations, described by η_i, such that

$$\langle\eta_i(t)\eta_j(t')\rangle = 2\beta D'_{ij}\delta_{ij}\delta^{(d)}(t-t'). \tag{S4}$$

We mention this term for completeness but focus on the purely deterministic limit in this paper. Thus, the effective dynamics of the system's composition as it exchanges with the reservoir are given by

$$\frac{d\phi_i}{dt} \approx -\sum_{j=0}^{N} D'_{ij}\frac{\partial\Omega_{\text{G}}}{\partial\phi_j} \tag{S5}$$

In writing our solute dynamics in the main manuscript, we treat the solvent implicitly. We first show below that this is tacit to assuming that the solvent molecules rearrange and equilibrate quickly to any small changes in solutes. The free energy is

$$\Omega_{\text{surface}} = \sum_{i=0}^{N}\phi_i\log\phi_i + \frac{1}{2}\sum_{i=0}^{N}\sum_{j=0}^{N}\phi_i\chi_{ij}\phi_j - \vec{\mu}''_{\text{res}}\cdot\vec{\phi} \tag{S6}$$

There is still a constraint $\sum_{i=0}^{N}\phi_i = 1$. We derive the dynamics of the system assuming Model A dynamics, where the mobility matrix $\mathbf{D}'$ is determined by first assuming that the mobility follows Fick's law of diffusion in the dilute limit, such

that $D'_{ij} = d_i\phi_i\delta_{ij}$ (2), and imposing the constraint via a Lagrange multiplier, so that $\Omega_{\mathsf{G}} = \Omega_{\mathsf{surface}} - \lambda\left(\sum_{i=0}^{N}\phi_i - 1\right)$. The dynamics from eq. S5 are therefore

$$\frac{\partial\phi_i}{\partial t} = -d_i\phi_i\left(\frac{\partial\Omega_{\mathsf{surface}}}{\partial\phi_i} - \lambda\right) = -d_i\phi_i\left[\beta\left(\mu''_i - \mu''_{\mathsf{res},i}\right) - \lambda\right] \tag{S7}$$

where

$$\beta\mu''_i = 1 + \log\phi_i + \sum_{j=0}^{N}\chi_{ij}\phi_j \tag{S8}$$

The constraint is given by $\frac{d\Omega_{\mathsf{G}}}{d\lambda} = 0$, which once differentiated is

$$\sum_{i=0}^{N}\frac{\partial\phi_i}{\partial t} = -\sum_{i=0}^{N}d_i\phi_i\left[\beta\left(\mu''_i - \mu''_{\mathsf{res},i}\right) - \lambda\right] = 0 \qquad \Longrightarrow \qquad \lambda = \frac{\sum_{j=0}^{N}d_j\phi_j\beta\left(\mu''_j - \mu''_{\mathsf{res},j}\right)}{\sum_{k=0}^{N}d_k\phi_k} \tag{S9}$$

Substituting λ and grouping the terms gives the dynamics

$$\frac{\partial\vec{\phi}}{\partial t} = -\mathbf{D}'\beta\left(\vec{\mu}'' - \vec{\mu}''_{\mathsf{res}}\right), \qquad D'_{ij} = d_i\phi_i\left(\delta_{ij} - \frac{d_j\phi_j}{\sum_{k=0}^{N}d_k\phi_k}\right) \tag{S10}$$

We now apply model assumptions. We assume first that $d_0 \gg d_i$ for $i > 0$ (implying that the solvent relaxes much faster than the solutes). Second, we assume that there are negligible effective solute-solvent interactions (taking $\chi_{0j} = 0$ for all j). This assumption implies that each solute is "well-solvated" and thus cannot phase separate on its own, i.e., when present in a binary mixture of only solute and solvent. In a fluid with N solutes and a solvent, this assumption maintains the same scaling of free parameters with increasing species ($\sim N^2$) but restricts the absolute number of such parameters with N fewer χ parameters to train. In this paper, we focus on this restricted limit where individual solutes are well-solvated, and in fact, are treated as effectively inert. Correspondingly, as explained in the lattice description (see SI Note 4), we enforce $\chi_{0j} = 0$ by setting both $\epsilon_{jj} = \epsilon_{0j} = 0$. Thus, the solutes we design exhibit multiphase behavior only through interaction with other types of solutes. This bears direct parallels with the restriction of individual neurons or cells forming associative connections with other cells but no self-connections as originally formulated in Hopfield models (3).

We note that this approximation poses challenges for experimental realization since it requires free-tuning of inter-species interactions while keeping self-interactions negligible. This poses issues when considering systems such as DNA nanostars that operate primarily through attractive, sequence-based interactions, where positive (or repulsive) χ_{ij} cannot be generically invoked without also allowing self interactions (i.e. pairs of species that have strong self-interaction but weak cross-interactions). In general, relaxing this assumption, allows N free parameters (χ_{0j}) to be freely tuned and could, in principle, increase expressivity of the solution. One such instantiation, in terms of the microscopic contact energies introduced in eq. 7, would stipulate that microscopic solvent interactions are zero ($\epsilon_{0j} = 0\ \forall j$) and solute self-interactions ϵ_{jj} are thus freely tuneable. In this paper, for simplicity, we focus on the theoretical limit of only inter-species interactions.

We can express the system dynamics in terms of the solute concentrations,

$$\frac{\partial\phi_0}{\partial t} \approx \sum_{j=1}^{N}\beta d_j\phi_j\left(\mu'_j - \mu'_{\mathsf{res},j}\right) \tag{S11}$$

$$\frac{\partial\phi_i}{\partial t} \approx -\beta d_i\phi_i\left(\mu'_i - \mu'_{\mathsf{res},i}\right), \qquad i > 0 \tag{S12}$$

or

$$\frac{\partial\vec{\phi}}{\partial t} \approx -\mathbf{D}'_{\mathsf{f}}\beta\left(\vec{\mu}' - \vec{\mu}'_{\mathsf{res}}\right), \qquad \left(\mathbf{D}'_{\mathsf{f}}\right)_{ij} = \begin{cases} d_j\phi_j(\delta_{0j} - 1), & i = 0 \\ d_i\phi_i\delta_{ij}, & i > 0 \end{cases} \tag{S13}$$

where

$$\beta\mu'_i = \beta\left(\mu''_i - \mu''_0\right) = \log\phi_i - \log\left(1 - \phi_T\right) + \sum_{j=1}^{N}\chi_{ij}\phi_j \tag{S14}$$

is the intrinsic (non-dimensionalized) chemical potential vector and $\phi_T = \sum_{i=1}^{N}\phi_i$ is the total solute volume fraction (i.e. omitting the solvent). The equations for $i > 0$ therefore form a matrix equation that is approximately diagonal in the limit of fast solvent dynamics.

Furthermore, this equation provides the solute dynamics used throughout this paper when we set $d_i = 0$ for $1 \leq i \leq N_{\text{in}}$ and $d_i = d$ for $i > N_{\text{in}}$, where d is a constant whose value does not affect the steady state. In this case, the input species are confined to the box, and the solute dynamics can be further simplified to be written only in terms of the $(N_{\text{out}} + N_{\text{h}}) \times (N_{\text{out}} + N_{\text{h}})$ lower block of the full mobility matrix $\mathbf{D}'_{\text{f}}$; this truncated matrix, which we label $\mathbf{D}$, is the mobility matrix used in eq. 8. Likewise, because the solvent is being treated implicitly and the inputs cannot exchange with the reservoir, the reservoir can be described by the length-$(N_{\text{out}} + N_{\text{h}})$ chemical potential vector $\vec{\mu}_{\text{res}}$ that includes only the output and hidden species—the convention used throughout the paper.

The solvent equation ($i = 0$) follows from—and implies—the constraint $\phi_0 = 1 - \phi_T$, since the solvent balances the flux of the solutes. Empirically, we also find that relaxing this assumption (by taking the solvent to have finite mobility compared to the solutes) essentially does not alter steady-state (or "performance") of trained multiphase fluids (Fig. S1).

Parameterizing the Dynamics

This section offers a parametrization for the case where we have fast solvent dynamics. Again, for the simplicity of the derivation, we consider all solutes (including input species) as mobile and therefore also take the reservoir potential vector to again be $\vec{\mu}'_{\text{res}}$, which is measured with respect to the solvent chemical potential.

Proposed Parametrization. In practice, the logarithmic terms in eq. S14 become unstable for $\phi_i \to 0$ or $\phi_T \to 1$, making even the simplified model in eq. S13 difficult to integrate. Eq. S14 therefore suggests the following parametrization:

$$x_i = \log\left(\frac{\phi_i}{1-\phi_T}\right) \qquad \Longleftrightarrow \qquad \phi_i = \frac{\exp(x_i)}{1+\sum_{j=1}^{N}\exp(x_j)} \tag{S15}$$

In this parametrization, the chemical potential vector simplifies to

$$\beta\vec{\mu}'(\vec{x},\chi) = \vec{x} + \chi\vec{\phi}(\vec{x}) \tag{S16}$$

This parametrization has the benefit that it spans all real numbers, thereby transforming the problem from a system of ODEs with constraints $\phi_i > 0$ and $\phi_T < 1$ to one that is unconstrained.

The inverse Jacobian of this transformation is

$$(\mathbf{J}^{-1})_{ij} = \frac{\partial x_i}{\partial \phi_j} = \frac{1}{\phi_j}\delta_{ij} + \frac{1}{1-\phi_T} \tag{S17}$$

As a result, the time-evolution of $\vec{x}$ is governed by

$$\frac{d\vec{x}}{dt} = \sum_{j=1}^{N}\frac{d\vec{x}}{d\phi_j}\frac{d\phi_j}{dt} = -\sum_{j,k=1}^{N}\frac{d\vec{x}}{d\phi_j}\left(\mathbf{D}'_{\text{f}}\right)_{jk}\frac{\partial\Omega_{\text{G}}}{\partial\phi_k} = -\beta\mathbf{J}^{-1}\mathbf{D}'_{\text{f}}\left[\vec{\mu}'(\vec{x},\chi) - \vec{\mu}'_{\text{res}}\right] \tag{S18}$$

where $\vec{\mu}'(\vec{x},\chi)$ is as defined in eq. S16. The transformed mobility matrix has components

$$\left(\mathbf{J}^{-1}\mathbf{D}'_{\text{f}}\right)_{ij} = \delta_{ij} + d_j\exp(x_j) \tag{S19}$$

In the case of fixed input concentrations, we take $d_i = 0$ for $1 \leq i \leq N_{\text{in}}$, such that $\phi'_i(t) = 0$ for the input species. Note that in the new parametrization, the input parameters $\vec{x}_{\text{in}} = \vec{x}_{\text{in}}(\vec{\phi})$ are no longer constant. However, the total input concentration of the surface is fixed at $\sum_{1\leq j\leq N_{\text{in}}}\phi_j \equiv \phi_{\text{in},T}$, and thus the non-input x_i can be evolved without needing to simultaneously evolve the input x_i, since

$$1 + \sum_{j=1}^{N}\exp(x_j) = \frac{1+\sum_{j>N_{\text{in}}}\exp(x_j)}{1-\phi_{\text{in},T}} \tag{S20}$$

This relation allows for the non-input components of $\vec{\phi}(\vec{x})$ in eq. S16 to be written independent of the input x_i coordinates. In turn, eq. S18 depends only on the (fixed) values of $\vec{\phi}_{\text{in}}$ and the non-input parameters x_i.

Supplementary Note 2: Training the model

Learning Rules

We optimize over both the χ matrix and the reservoir chemical potential $\vec{\mu}_{\text{res}}$, with the target being an enrichment in the desired output species for a given input concentration vector $\vec{\phi}_{\text{in}}$. Unlike in the case of an artificial neural network, where there are no physical constraints on the weights assigned to the hidden layers, a surface is constrained to have total volume fraction 1. Therefore, unlike the unconstrained problem, the cross-entropy of the output vector is not a favorable loss function, because imposing that the desired output concentration be as close as possible to 1 depletes the volume fraction available to the hidden species, thereby limiting their effectiveness. We require instead that the following criteria be captured by our loss function:

1. The final concentration of the desired output species should be above some threshold value $\phi_{\text{max}} = A/N$, where N is the total number of particle species and A is a value to be specified.
2. The final concentrations of the undesired output species should be below some threshold $\phi_{\text{min}} = B/N$, where B is a value to be specified.

These two criteria in turn enforce that the ratio of desired to undesired outputs should be above a set threshold A/B, and that this ratio is attained with a sufficiently enriched output species. In principle, for a steady-state concentration vector $\vec{\phi}$ where the j'th output is desired to be enriched, the above criteria could have been satisfied by using, instead of l_j in eq. 9, the function

$$\tilde{l}_j(\chi, \vec{\mu}_{\text{res}}) = -\sum_{k \neq j} \log\left[\frac{\min\left(1, \phi_{\text{out},j}/\phi_{\text{max}}\right)}{\max\left(1, \phi_{\text{out},k}/\phi_{\text{min}}\right)}\right] \tag{S21}$$

which enforces that the ratio in the argument of the log be as close as possible to 1, and therefore that $\phi_{\text{out},j}/\phi_{\text{out},k} > \phi_{\text{max}}/\phi_{\text{min}}$, while the numerator and denominator are independent of $\vec{\phi}_{\text{out}}$ when their values are above and below (respectively) their corresponding threshold values. This function indeed allows for successful decision boundaries to be sculpted. In practice, we further adjust this loss empirically to improve training. In particular, we use the fact that

$$-\log\left(\frac{\min(1,x)}{\max(1,y)}\right) = -\log\left(\min(1,x)\right) + \log\left(\max(1,y)\right) \tag{S22}$$

$$= -\log\left(1 - \max(0, 1-x)\right) + \log\left(1 + \max(0, y-1)\right) \tag{S23}$$

$$= \log\left(1 + \max(0, 1-x)\right) + \log\left(1 + \max(0, y-1)\right) + \mathcal{O}(\Delta x^2) \tag{S24}$$

where $\Delta x = 1 - x$ is small. Motivated by this expansion, and combined with empirical tests, we use the loss function

$$l_j(\chi, \vec{\mu}_{\text{res}}) = \log\left(1 + A\max\left(0, 1 - \frac{\phi_{\text{out},j}}{\phi_{\text{max}}}\right)\right) + \sum_{k \neq j} \log\left(1 + B\max\left(0, \frac{\phi_{\text{out},k}}{\phi_{\text{min}}} - 1\right)\right) \tag{S25}$$

We've deviated from the expansion of eq. S21 by dropping the factor of $(N-1)$ that would otherwise be on the i-dependent logarithm and also by introducing the hyperparameters $A = \phi_{\text{max}}N$ and $B = \phi_{\text{min}}N$ as prefactors in the logarithms. We find that this choice for the loss function gives strong results near decision boundaries, and we therefore favor l_j over $\tilde{l}_j$ for training our molecular networks. Substituting the expressions for A and B results in the form of the loss function in the main text. As discussed in the main text, we further constrain the magnitude of χ such that $|\chi_{ij}| < 15$. Following eq. 7 and the lattice definition of 18 nearest neighbors (SI Note 4), this constraint corresponds to magnitudes of $\epsilon_{ij} \sim k_{\text{B}}T$ for individual molecular bonds and is consistent with liquid-like mixtures where molecules can rearrange with thermal fluctuations.

Hyperparameter choices

We minimize $\mathcal{L}$ with respect to χ and $\vec{\mu}_{\text{res}}$ over several thousand training epochs using an RMSProp algorithm from the Optax library (4, 5) with an initial learning rate of 0.01, followed by several thousand more epochs with a learning rate of 0.001 to improve convergence. We use 5000 training points and a mini-batching scheme where $n_{\text{batch}} = 128$ randomly selected training points are evaluated at each epoch. Once trained, we construct a validation set of 500 data points to validate the classifier. For a given number of hidden species, we perform this optimization procedure over 15 initial guesses in the loss landscape. Using the definition of success S_{c} in eq. 13, the trained model performance is evaluated on the validation set and the best performing model is subsequently applied to an independent test set (of same size as the validation set) and depicted in figures.

Supplementary Note 3: Decision boundary

To understand the constraints on the shapes the decision boundary can encode in our model, we first provide insights with a simplified model only 2 inputs and 2 output species, and expand in later sections to explore the effect of adding more species. Without loss of generality, and consistent with our design articulated in the main text, we assume $\beta = 1$ below.

2 input + 2 output + 0 hidden species

In the case of mixtures with 2 input species (with species labels $i = 1, 2$) and 2 output species (with species labels $i = 3, 4$), the concentration vector is given by $\vec{\phi} = (\phi_{\text{in},1}, \phi_{\text{in},2}, \phi_{\text{out},1}, \phi_{\text{out},2})$. The decision boundary is defined as the manifold where output species are equally recruited, with $\phi_{\text{out},1} = \phi_{\text{out},2} \equiv \phi_o$. For a trained mixture with parameters $(\chi, \vec{\mu}_{\text{res}})$, the steady-state conditions of eq. S13 along this manifold are

$$\mu^{\text{res}}_{\text{out},1} = \log(\phi_o) - \log(1 - \phi_T) + \sum_{j=1}^{4} \chi_{3j}\phi_j \tag{S26}$$

$$\mu^{\text{res}}_{\text{out},2} = \log(\phi_o) - \log(1 - \phi_T) + \sum_{j=1}^{4} \chi_{4j}\phi_j \tag{S27}$$

Defining $\Delta\mu_{\text{res,out}} = \mu^{\text{res}}_{\text{out},1} - \mu^{\text{res}}_{\text{out},2}$, the difference of the above two equations is independent of the specific value of the output concentrations,

$$\Delta\mu_{\text{res,out}} = (\chi_{31} - \chi_{41})\phi_{\text{in},1} + (\chi_{32} - \chi_{42})\phi_{\text{in},2} \tag{S28}$$

and defines a decision manifold across which the recruited output species changes from species 1 to species 2. Recall that all diagonal elements of χ are 0 by definition, and the output-output interaction contributions cancel exactly thanks to the symmetry $\chi_{ij} = \chi_{ji}$. The decision boundary described in eq. S28 is therefore exactly linear in the inputs. Fig. S3A shows two theoretically computed linear boundaries using eq. S28.

N_{in} input + N_{out} output + 0 hidden species

Generalizing to N_{in} input species (with species labels $i = 1, \ldots, N_{\text{in}}$) and N_{out} output species (with species labels $i = N_{\text{in}} + 1, \cdots, N_{\text{in}} + N_{\text{out}}$, such that $\phi_{N_{\text{in}}+n} = \phi_{\text{out},n}$ for $n = 1, \ldots, N_{\text{out}}$), we see that the decision boundary between any two output species at equal concentrations ($\phi_{\text{out},n} \equiv \phi_i = \phi_o = \phi_j \equiv \phi_{\text{out},m}$) can similarly be written as

$$\Delta\mu^{(n,m)}_{\text{res,out}} = \sum_{k=1}^{N_{\text{in}}} (\chi_{ik} - \chi_{jk})\phi_{\text{in},k} + \sum_{\substack{k=N_{\text{in}}+1 \\ (k\neq i,j)}}^{N_{\text{in}}+N_{\text{out}}} (\chi_{ik} - \chi_{jk})\phi_k(\vec{\phi}_{\text{in}}, \vec{\mu}_{\text{res}}, \chi) \tag{S29}$$

where $\Delta\mu^{(n,m)}_{\text{res,out}} = \mu^{\text{res}}_{\text{out},n} - \mu^{\text{res}}_{\text{out},m}$. The first term is the generalization of eq. S28 to sum over all inputs, and second term is a sum over the remaining (non-boundary) output species. Unlike the previous case, here we treat the concentrations of non-boundary output species as nonlinear functions of input species, and as such they could encode more complex boundaries. Strictly speaking, as the energy landscape may have multiple local minima, the final output concentrations may not be uniquely determined by the input concentrations; however, in this work, training appears to avoid this situation for the cases we have tested, in part, because during training, the initial hidden/output concentrations are randomly assigned at different epochs. When training is successful, target surfaces typically enrich a single output species with the others being depleted, and the resulting output concentrations will have $\phi_k \ll 1$ for $k \neq i, j$. Since the χ matrix has components that are constrained to be $|\chi_{ij}| < \chi_{\text{max}}$, the second term in eq. S29 should therefore be negligible for solutions obeying the loss criterion. As a result, trained mixtures of surface condensing species form generalized linear boundaries as a function of input species concentrations:

$$\Delta\mu^{(n,m)}_{\text{res,out}} \approx \sum_{k=1}^{N_{\text{in}}} (\chi_{ik} - \chi_{jk})\phi_{\text{in},k} \tag{S30}$$

Fig. S3B shows two theoretically computed boundaries using eq. S30 for $N_{\text{in}} = 2$, $N_{\text{out}} = 3$, compared with numerical results. This linearity breaks down in the vicinity of points in the input space where multiple classes meet, in which case there are more than two relevant output species, and the second term in eq. S29 is no longer negligible. The decision boundaries therefore resemble hyperplanes far from regions of multiclass intersection, but can also potentially be nonlinear.

2 input + 2 output + 1 hidden species

In general, with the inclusion of hidden nodes, the equations become analytically intractable. Here we consider the inclusion of a single hidden species and show that this is sufficient for producing nonlinear decision boundaries. We consider a concentration vector as defined above, where indices 1 and 2 correspond to input species, 3 and 4 correspond to output species, and an additional component (species index 5) corresponds to the hidden species. The concentration vector then reads as $\vec{\phi} = (\phi_{\text{in},1}, \phi_{\text{in},2}, \phi_{\text{out},1}, \phi_{\text{out},2}, \phi_{\text{h}})$. When $\phi_{\text{out},1} = \phi_{\text{out},2} = \phi_o$, the decision boundary follows eq. S28 with the the modification

$$\Delta\mu_{\text{res,out}} = (\chi_{31} - \chi_{41})\phi_{\text{in},1} + (\chi_{32} - \chi_{42})\phi_{\text{in},2} + (\chi_{35} - \chi_{45})\phi_{\text{h}}(\vec{\phi}, \vec{\mu}_{\text{res}}, \chi) \tag{S31}$$

While the concentration of the hidden species is (assumed to be) an implicit function dependent on input concentrations, the shape of this boundary is generically hard to interpret. The chemical potential of the hidden species is given by

$$\mu_{\text{h}} = \log(\phi_{\text{h}}) - \log(1 - \phi_T) + \sum_{k=1}^{5} \chi_{5k}\phi_k$$

Since $\chi_{5,5} = 0$, the sum on the right hand side is independent of ϕ_{h}, and we can therefore isolate for ϕ_{h} as

$$e^{\mu_{\text{h}}} = \frac{\phi_{\text{h}}}{(1 - \phi_{\text{in},T} - 2\phi_o) - \phi_{\text{h}}} \exp\left(\sum_k \chi_{5k}\phi_k\right) \qquad \Longrightarrow \qquad \phi_{\text{h}} = \frac{1 - \phi_{\text{in},T} - 2\phi_o}{1 + \exp\left(-\mu_{\text{h}} + \sum_k \chi_{5k}\phi_k\right)} \tag{S32}$$

and the decision boundary in the input space thus obeys

$$\Delta\mu_{\text{res,out}} = (\chi_{31} - \chi_{41})\phi_{\text{in},1} + (\chi_{32} - \chi_{42})\phi_{\text{in},2} + \frac{(\chi_{35} - \chi_{45})\left((1 - 2\phi_o) - \phi_{\text{in},1} - \phi_{\text{in},2}\right)}{1 + \exp\left(-\mu_{\text{h}} + (\chi_{35} + \chi_{4,5})\phi_o\right)\exp(\chi_{51}\phi_{\text{in},1} + \chi_{52}\phi_{\text{in},2})} \tag{S33}$$

Fig. S3C shows two nonlinear decision boundaries for systems with a single hidden species, computed theoretically from eq. S33, against numerical results. To see how this rather complex equation permits nonlinear boundaries, it is instructive to look at the following limit when (a) input-output interactions are identical across species, (b) output-hidden interactions are non-zero and different, and (c) hidden-input interactions are strong and of opposing signs. Here, the boundary will primarily be defined by variances in the relative interaction of the hidden species with the two inputs. The features of such a decision boundary in the input plane can be computed as an implicit derivative from the decision boundary since output concentrations at the decision boundaries are low ($\phi_o \approx 0$) and hence eq. S33 is of the form $f(\phi_{\text{in},1}, \phi_{\text{in},2}) = 0$. As such,

$$\frac{d\phi_{\text{in},2}}{d\phi_{\text{in},1}} = -\frac{df/d\phi_{\text{in},1}}{df/d\phi_{\text{in},2}} = -\frac{1 + g(\mu_{\text{h}}, \vec{\phi}_{\text{in}})(1 + \chi_{51}(1 - \phi_{\text{in},1} - \phi_{\text{in},2}))}{1 + g(\mu_{\text{h}}, \vec{\phi}_{\text{in}})(1 + \chi_{52}(1 - \phi_{\text{in},1} - \phi_{\text{in},2}))} \tag{S34}$$

where

$$g(\mu_{\text{h}}, \vec{\phi}_{\text{in}}) = \exp(-\mu_{\text{h}})\exp(\chi_{51}\phi_{\text{in},1} + \chi_{52}\phi_{\text{in},2}) \tag{S35}$$

While the prefactor functions g are always positive, the second factor of the form $\chi_{5j}(1 - \phi_{in,1} - \phi_{in,2})$ for $j \in [1, 2]$ can change the sign and magnitude of the whole term depending on a particular choice of parameters. In general, this implies that not only is the decision boundary nonlinear (i.e. non-uniform slope for changing magnitude) but also capable of changing curvatures (i.e. changing of slope signs).

N_{in} input + N_{out} output + N_{h} hidden species

In general, since $\phi_{\text{out},n}$ and $\phi_{\text{h},m}$ are implicit functions of the input volume fractions $\phi_{\text{in},k}$, one cannot assume any particular shape of the decision boundary on the input space.

A potential route for a universal approximation construction

A hallmark of general-purpose machine learning architectures is that there is a well-defined sense in which they can approximate any target function of any complexity by scaling their size. The universal approximation theorem for multilayer sigmoidal feedforward networks used in early backpropagation algorithms is a canonical example of this kind of argument (6, 7). Motivated by this, and making the assumptions outlined below, we discuss a path towards showing that arbitrary continuous decision boundaries can be achieved by surface condensates by increasing the number of hidden species.

Linear decision boundaries and connection to winner-take-all dynamics. We first revisit the mixture considered above, consisting of N_{in} input species and N_{out} output species (and 0 hidden species), where all the output species are strongly mutually repulsive and thus form distinct condensates that are each enriched only in one output species. The decision boundary between a condensate of output species m and another with output species n is given by eq. S30 (where $\phi_{\text{out},m} \approx \phi_{\text{out},n} = \phi_o \gg \phi_{\text{out},i\neq m,n}$), rewritten for simplicity as

$$\sum_{k=1}^{N_{\text{in}}} (\chi_{mk} - \chi_{nk})\phi_{\text{in},k} - (\mu^{\text{res}}_{\text{out},m} - \mu^{\text{res}}_{\text{out},n}) = 0 \tag{S36}$$

As originally noted, the decision boundaries are linear planes in the input space. Then, the non-dimensionalized energy of a surface (as seen in eq. 5) of a condensate enriched in output species m ($\phi_{\text{out},m} \approx \phi_o$) and only with negligible amounts of other output species can be approximated as:

$$\Omega^m_{\text{surface}} \approx \sum_{i=1}^{N} \phi_i \log \phi_i + (1-\phi_T)\log(1-\phi_T) + (\sum_{i=1}^{N_{in}} \chi_{im}\phi_{\text{in},i} - \mu^{\text{res}}_m)\phi_o \tag{S37}$$

We define the corresponding score function

$$f_m(\vec{\phi}_{\text{in}}) = \sum_{k=1}^{N_{\text{in}}} \chi_{km}\phi_k - \mu^{res}_m \ = \vec{w}_m \cdot \vec{\phi}_{\text{in}} + b_m \ . \tag{S38}$$

Assuming the output condensate composition doesn't change majorly away from decision boundaries, the free energy difference between surface condensates enriched in output pairs (m, n) can be written as:

$$\Omega^m_{\text{surface}} - \Omega^n_{\text{surface}} = \phi_o \left(f_m(\vec{\phi}_{\text{in}}) - f_n(\vec{\phi}_{\text{in}}) \right) \tag{S39}$$

Since the mean-field model drives a steady-state composition that minimizes this free energy, we see that eq. S39 gives rise to a winner-take-all (WTA) form of dynamics. That is to say, output n dominates other output species in the condensate when

$$n = \text{argmin}_m \ f_m(\vec{\phi}_{\text{in}}) = \text{argmin}_m \ \vec{w}_m \cdot \vec{\phi}_{\text{in}} + b_m \ . \tag{S40}$$

In this sense, surface condensation behaves as a computational structure with N_{in} inputs that are linearly weighted in N_{out} ways, such that the winning output is chosen and the others suppressed. This exact computational structure has been well studied in the neural computation and machine learning literature, where it is it often described as a single linear layer followed by a winner-take-all (WTA) unit (7, 8). Such WTA circuits have been proposed theoretically and demonstrated experimentally as an architecture for biochemical computation (9–13). Notably, the linearly-weighted WTA is a powerful primitive, with a single such unit being able to perform classification that requires a quadratic number of classical linear threshold units (8).

However, a single linearly-weighted WTA remains computationally limited, in a way that can be visualized geometrically. Specifically, the region of input space that activates a given output must be a single convex polyhedron, i.e., bounded by planes that are the linear decision boundaries (as in the bottom left of Fig. S17). This suggests a close connection between linearly-weighted WTA and Voronoi diagrams (14) that is worth understanding here. Given a set of N_{out} reference points with centers $\vec{c}_m$, the Voronoi diagram partitions the input space into regions consisting of all points that are closest to a given reference point. Mathematically, input vector $\vec{\phi}_{\text{in}}$ is in region n if

$$n = \text{argmin}_m \ \|\vec{\phi}_{\text{in}} - \vec{c}_m\| = \text{argmin}_m \ (\vec{\phi}_{\text{in}} - \vec{c}_m)\cdot(\vec{\phi}_{\text{in}} - \vec{c}_m) = \text{argmin}_m \ \vec{w}_m \cdot \vec{\phi}_{\text{in}} + b_m \ . \tag{S41}$$

for $\vec{w}_m = -\vec{c}_m$ and $b_m = \frac{1}{2}\|\vec{c}_m\|^2$. So any Voronoi classification can be achieved by a single linearly-weighted WTA. In fact, linearly-weighted WTA can implement a slightly larger class of *generalized* Voronoi diagrams in which

$$n = \text{argmin}_m \ \|\vec{\phi}_{\text{in}} - \vec{c}_m\|^2 + d_m \tag{S42}$$

where each region m is augmented by an "importance" factor d_m. In the above construction, the linearly-weighted WTA is obtained simply by increasing b_m by d_m.

Thus, we claim that N_{in}-input, N_{output} surface condensates with no hidden species can perform any classification task in which each output class corresponds to a single cell in a generalized Voronoi diagram. However, it is fair to ask whether the above construction results in reasonable physical parameters χ_{km} and μ_m^{res}. For example, the training elsewhere in this paper was restricted to $\|\chi_{ij}\| < 15$ and used $\|\mu_m^{\text{res}}\| < 6$. In the Voronoi approach, we would choose cell centers $\vec{c}$ with $0 < c_i < 1$ since they represent composition fractions, so in fact $\|\chi_{km}\| < 1$ and $\|\mu_m^{\text{res}}\| < 1$, which ought to be physically feasible. Note that the decision boundaries are unaffected by scaling χ and μ^{res}, so if phase separation does not result in sharp enough classification, we can increase both by the same factor until either sharp enough performance is achieved or the limits of physicality are hit.

While this argument provides a mostly satisfactory characterization of the computational power of classification by surface condensation with no hidden species, it also illustrates limits to classification without hidden species. Specifically, non-convex decision boundaries cannot be achieved or even approximated well. Further, the extent to which computational power can be expressed primarily through a linear WTA mechanism, as derived from the approximate surface energy in eq S37, requires further characterization of the key assumption that condensates are essentially "one-hot" in output species recruitment far from the decision boundaries.

Decision boundary approximation construction. In the machine learning literature, the limitations of a single linearly-weighted WTA unit can be overcome in several ways, leading to universal approximation theorems for the enhanced computational architectures (7, 8). The most common approach is to allow multi-layer network computation, but perhaps the simplest generalization is known as the *piecewise linear machine* (15): rather than the WTA outputs being the class labels directly, with one class per WTA output, the piecewise linear machine has N_{in} input signals routed with linear weights to a single WTA unit with N_{h} lines, and winning output n is assigned a class label $l_n \in \{1 \dots N_{\text{out}}\}$. That is, for input vector $\vec{\phi}_{\text{in}}$ the classification decision is

$$l_n \quad \text{if and only if} \quad n = \text{argmin}_m \ \vec{w}_m \cdot \vec{\phi}_{\text{in}} + b_m \quad . \tag{S43}$$

Essentially, the computation is coloring a (generalized) Voronoi diagram with as many cells as needed to approximate the desired decision boundaries, and then coloring each cell with N_{out} colors (as in the bottom right of Fig. S17).

Let us now translate this approach into the language and mechanisms of surface condensation. First, let us informally note that if the surface classification result were indicated by a moiety ("red", "green", "blue", ...) of the output species, rather than the full species identity, then the above WTA construct suffices with almost no modification: just "color" each output species according to the desired class associated with its Voronoi cell. However, in this paper we require one of N_{out} distinct output species to be recruited. We will do this by using the above WTA construction to recruit one of N_{h} hidden species according to the relevant Voronoi cell, and then have that hidden species recruit a specific output species associated with that cell. To show that this construction will work, we must show that the hidden unit computation and output computation do not interfere with each other. Now the construction is given in detail.

We consider the following problem: Given N_{in} input species and N_{out} output species, we desire a target decision function $g(\vec{\phi}_{\text{in}}) \in \{1, \dots, N_{\text{out}}\}$. When $g(\vec{\phi}_{\text{in}}) = j$, as with the original model definition, we require the output species j to be selectively recruited at much higher concentrations over other output species, i.e., $\phi_{\text{out},j} = \phi_* \gg \phi_{\text{out},k\neq j}$. Note that, in this formulation, we don't require an **absolute** high value for ϕ_*, just that it is much more than other output species. Our goal is to design χ and $\vec{\mu}_{\text{res}}$ that can achieve this for arbitrary g, to any necessary degree of accuracy.

To achieve this, we suggest the following construction, outlined in Fig. S17. First, we consider a linear partition of the Euclidean input plane $\mathbb{R}^{N_{\text{in}}}$ into n_{p} cells. For example, a specific instantiation of this would be the Voronoi tesselation of n_p prototypical input concentrations, $\left(\vec{\phi}_{\text{in}}^{1}, \vec{\phi}_{\text{in}}^{2}, \dots, \vec{\phi}_{\text{in}}^{n_{\text{p}}}\right)$. Importantly, $n_{\text{p}} \gg N_{\text{out}}$ is a free parameter, and as it increases, one can achieve increasingly finer partitioning of the input space. For two-dimensional input, note that with this partitioning, pairs of cells share linear decision boundaries and a finite number of *vertices* where 3 or more decision boundaries meet.

Following our connection to linear partitions in eq. S39, we propose including 1 species for each of the n_p classes, which we label as *hidden* species ($N_{\text{h}} = n_{\text{p}}$). Despite the label name, hidden species are treated similar to outputs, in that interactions between any two hidden species is unfavorable $\chi_{ij} = (1 - \delta_{ij})\chi_{\text{pen}}$ for hidden species i and j, where $\chi_{\text{pen}} \gg 0$. At strong interactions, this is sufficient to encode for condensates that are each enriched in 1 hidden species and exclude all others. As described by eq. S30, the resulting molecular network encodes linear boundaries between condensates enriched in hidden species m versus hidden species n. From the score function eq. S38, we see that boundary slopes (weights) and intercepts (biases) are determined by the subset of tunable interactions $(\chi_{km}, \mu_m^{\text{res}})$ that are freely chosen for k being any input species and m being any hidden species. For a desired partitioning, these values can be assigned by the system of linear equations or gradient-descent based approaches.

With this construction, we return to the original objective of achieving a decision boundary of type $g(\vec{\phi}_{\text{in}})$ with N_{out} output species. Note that, with $n_p \gg N_{\text{out}}$, for a specific function g, we need to appropriately assign each of the n_p cells to the appropriate output. To accomplish this, we enforce that each hidden species has an attractive interaction with exactly one output species, and no interactions with the others, i.e., for hidden species n and output species m, we have $\chi_{nm} = \delta_{l_n,m}\chi_{\text{recruit}}$ where l_n is the class label as in eqn. S43 and $\chi_{\text{recruit}} \ll 0$. As before, all the N_{out} species also have unfavorable interactions between each other. Generically, since multiple cells can be colored with the same output, this provides a many-to-one attractive interaction from hidden species to outputs. We stipulate that the output species reservoir potentials are identical and cannot directly interact with inputs. This is, in principle, analogous to a "client" (output) and "scaffold" (hidden) relationship that has been proposed to study biological condensates (16). This construction justifies the assumptions that (a) except on decision boundaries, the surface will recruit at most one hidden species n and at most one output species m, with the other hidden and output species being negligible, and (b) the amount of solvent within the condensate will be negligible. To show that the decision boundaries established by the WTA construction remain intact, consider the boundary between some hidden/output pair n, m and a distinct hidden/output pair n', m'. Within a cell, the surface energy is

$$\begin{aligned}\Omega^{n,m}_{\text{surface}} &= \sum_{i=1}^{N} \phi_i \log\phi_i + (1-\phi_T)\log(1-\phi_T) + \frac{1}{2}\sum_{i,j}\phi_i\chi_{ij}\phi_j - \sum_i \mu_i\phi_i \\ &\approx \sum_{i=1}^{N_{\text{in}}} \phi_i\log\phi_i + \phi_n\log\phi_n + \phi_m\log\phi_m + \left(\sum_{i=1}^{N_{in}}\chi_{in}\phi_i - \mu^{\text{res}}_n\right)\phi_n + \left(\phi_n\chi_{\text{recruit}}\phi_m - \mu^{\text{res}}_{\text{output}}\phi_m\right)\end{aligned} \tag{S44}$$

At the classification decision boundary, $\phi_m = \phi_{m'}$ by definition. Following the "one-hot" assumption, since non-input species are negligible compared to relevant hidden species (n, n'), $\phi_n \approx \phi_{n'}$, so we assume $\phi_n = \phi_{n'} = \phi_h$. Thus when considering the difference in energy, most terms cancel out and what remains is

$$\Omega^{n,m}_{\text{surface}} - \Omega^{n',m'}_{\text{surface}} \approx \left(\sum_{i=1}^{N_{in}}\chi_{in}\phi_i - \mu^{\text{res}}_n\right)\phi_h - \left(\sum_{i=1}^{N_{in}}\chi_{in'}\phi_i - \mu^{\text{res}}_{n'}\right)\phi_h \tag{S45}$$

which gives the same linear decision boundary as before. A different decision boundary can be achieved by simply reassigning the hidden-output map of interactions as above. With sufficiently large n_{p}, this model should enable for increasingly complex (convex and non-convex) decision boundaries.

This construction provides sufficient basis for our universal function approximation claim subject to assumptions specified below. Specifically, the input plane is partitioned into linearly separable regions that each exhibit WTA classification, the number of regions can scale with the number of hidden species, and selective mapping of hidden-output species can approximate arbitrary decision functions—analogous to what is known as a piecewise linear machine for pattern recognition (7, 15). It is intriguing to note that although the universality of this construction relies on potentially using many hidden species, any given resulting condensate is dominated by just one hidden species and one output species; the molecular complexity of the system is not reflected in the simplicity of the outcome. Beyond existence, the surface condensation driven WTA regions are more flexible than the Voronoi-inspired construction above, so they might be able to achieve a given level of approximation with fewer species; similarly, gradient descent training may be able to find better approximations with fewer species by exploiting non-"one-hot" hidden representations, direct input-to-output interactions, and other nonlinearities.

We consider the above to only be a rough sketch for a universal approximation theorem for surface condensation. This framework makes several assumptions (below) that still require rigorous testing and investigation.

Key assumptions. First, note that this construction is only true away from parts of the input space where multiple decision boundaries meet, i.e., vertices of the decision planes, which we assume only excludes a finite number of points from an infinite plane. Second, within each decision region (a particular group of input surfaces as per our model definition), we assume that the "one-hot" condensate encoded by the complementary hidden species is always the only steady-state with negligible composition of other species i.e., even far away from the decision boundaries which, by construction, are the coexistence plane for two "one-hot" condensate compositions. While this steady-state should naturally exhibit WTA behavior in the mean-field limit (see eq. S39), this is unlikely in 3D liquids, where pockets of distinct phases may coexist within the same surface. Third, we assume the selective inclusion of hidden-output favorable links do not destabilize or change overall boundaries. Finally, since we require liquid-like condensates, the range of allowable χ_{pen} values are constrained. For increasingly complex decision boundaries that require many species, the entropic costs increase and may not be offset by the bounded value of χ_{ij} values. This potential failure mode can, in principle, be alleviated by considering polymeric molecules, where entropy costs of demixing are significantly reduced by the degree of polymerization.

Supplementary Note 4: Lattice liquid model

Lattice setup and boundary conditions

Simulations are performed on a three-dimensional cubic lattice of dimensions $24 \times 24 \times 24$, except where specified otherwise. We find that classifier performance saturates with bigger lattice sizes, saturating after a lattice width of $L \sim 16$, motivating our choice above (Fig. S13A). Each lattice position $\mathbf{p} = (z, y, x)$ can be occupied by a single species from the set $\{0, 1, \ldots, N\}$. Species 0 is treated as an inert solvent with zero chemical potential and inert interactions. The boundaries are *walls* meaning that no interactions wrap around from one lattice face to its opposite face. Consequently, any site on a boundary has fewer neighbors than an interior site. To capture interactions out to $\sqrt{2}$ in Euclidean distance, each lattice site has up to 18 neighbors. Note that this is consistent with short-range interactions like the standard Lennard-Jones potential that typically only has an impact 1-2 molecule lengths away. Specifically, if $\mathbf{p}_1$ and $\mathbf{p}_2$ differ by at most 1 in up to two of their three coordinates, then $\mathbf{p}_2$ is in the neighborhood of $\mathbf{p}_1$. Positions outside the lattice bounds are ignored.

Free energy model

Let ϵ_{ij} denote the pairwise interaction parameter between species i and j, and let γ_i denote the chemical potential of species i. We work at inverse temperature $\beta = 1/(k_B T)$. For a given configuration σ, the total interaction energy is computed by summing over all lattice sites. A configuration σ induces a count n_i for each molecule type i. Defining $\delta_{a,b}$ as the Kronecker delta, which is 1 if $a = b$ and 0 otherwise, the count n_i is given by

$$n_i \;=\; \sum_{\mathbf{p}} \delta_{\sigma(\mathbf{p}),\, i}, \tag{S46}$$

where the sum is taken over all lattice sites $\mathbf{p}$ in the system. To calculate the system energy, for each site $\mathbf{p}$ with species $i = \sigma(\mathbf{p})$ and each neighbor $\mathbf{q}$ with species $j = \sigma(\mathbf{q})$, we add $\beta \epsilon_{ij}$. To avoid double counting, we include a factor of $\frac{1}{2}$ in the total. The free energy, including chemical potentials, for a particular configuration may be written as:

$$\beta H(\sigma) \;=\; \tfrac{1}{2}\beta \sum_{\mathbf{p}} \sum_{\mathbf{q} \in \eta(\mathbf{p})} \epsilon_{\sigma(\mathbf{p}),\sigma(\mathbf{q})} \;+\; \beta \sum_{i=1}^{N} \gamma_i n_i, \tag{S47}$$

where $\eta(\mathbf{p})$ denotes the neighborhood of site $\mathbf{p}$, and species 0 (solvent) has $\mu_0 = 0$ by definition.

Parameter mapping from Model A

The interaction energies and chemical potentials used in these lattice Monte Carlo simulations are derived from Model A. Specifically, a mapping is applied to convert the Model A parameters (denoted $\chi, \vec{\mu}_{\text{res}}$) to lattice-gas (LG) parameters (ϵ_{ij}, γ_i). First, to convert from the mean field description (at $\beta = 1$) to our lattice gas formulation, the pairwise interaction coefficients are scaled by a factor of $\frac{1}{z}$, where the coordination number z indicates the number of neighbors for interior lattice sites. Thus we have

$$\epsilon_{ij} \;=\; \frac{1}{18}\chi_{ij} \tag{S48}$$

when we incorporate the assumptions that effective solute-solvent interactions are negligible (by setting $\epsilon_{i0} = \epsilon_{00} = 0$) and that solute monomers are non-self-interacting (by setting $\epsilon_{ii} = 0$), where 0 indexes the solvent. These assumptions are consistent with $\chi_{i0} = 0$, as seen more clearly in a later subsection. We set the solvent potential also to be $\gamma_0 = 0$, and under these assumptions

$$\gamma_i = -\mu_{\text{res},i} \quad \forall i \in (N_{in} + 1, N) \tag{S49}$$

Canonical vs. Grand Canonical moves

We implement two fundamental move types in each MC step:

(1) *Canonical (NVT) moves*, which exchange species between two lattice sites to conserve particle counts;

(2) *Grand Canonical (μVT) moves*, which insert or remove species at a single site, exchanging with an implicit infinite reservoir.

We treat the "input" species (for example, species 1 and 2) as *clamped*, meaning they cannot exchange position within the lattice or identity with the reservoir. For every move proposition, if an original site holds an *input* species, the replacement probability is zero (no replacement allowed), keeping the counts and positions of input species fixed. By contrast, all other species (including the solvent) can freely exchange within the remaining sites of the lattice and with the reservoir. These species are handled using both canonical and grand canonical moves. This mixed ensemble preserves the total amount and positions of each input species while allowing all other components to exchange with an infinite reservoir.

Canonical (NVT) moves. Starting from a selected collection of positions (described in more detail in GPU-Accelerated Implementation and Masking):

- Pair up any two sites $(\mathbf{p_1}, \mathbf{p_2})$ (global swaps).
- Propose swapping the species at $\mathbf{p_1}$ and $\mathbf{p_2}$.
- Compute the change ΔH in interaction energy $H(\sigma)$ induced by swapping the two species, and accept with the Metropolis-Hastings probability

$$P_{\text{accept}} = \min\{1, \exp[-\beta\,\Delta H]\}. \tag{S50}$$

- If accepted and neither position contains a clamped species, exchange the species; otherwise leave them unchanged.

Grand Canonical (μVT) moves. Starting from a selected collection of positions:

- For a position $\mathbf{p}$ with current species i, propose one of the free (unclamped) species as new species j.
- Compute the combined energy change ΔH that includes both interaction energies and the chemical-potential difference, then accept with probability

$$P_{\text{accept}} \;=\; \min\Big\{\,1,\; \exp\big[-\beta\,\Delta H\big]\,\Big\}. \tag{S51}$$

- If accepted and the original species is unclamped, update the site to species j; otherwise leave it unchanged.

Initialization and equilibrium

The lattice is initialized with only *input* species and solvent. Input species are assigned randomly to lattice sites according to a total sum of a fraction ϕ_i for each input species i. All remaining sites are filled with the inert solvent ($i = 0$).

Once initialized, the system is evolved via repeated MC moves (either NVT or μVT with equal probability) until the total free energy and the species counts remain stable over a sufficiently long period (on the order of 10^5 accepted moves). Asymptotic behavior was verified to be independent of sampling protocol (i.e. frequency of NVT vs μVT swaps or lattice size), as expected for equilibrium. We record 1000 equally spaced lattice configurations over the MC protocol and use the last 100 frames to estimate average species counts as the near-equilibrium state for analysis. Each simulation condition is repeated in triplicate, i.e. with 3 different random seeds but identical parameters, for averaging.

GPU-accelerated implementation and masking

All Monte Carlo sweeps are implemented in JAX and executed through the Accelerated Linear Algebra (XLA) compiler, combining just-in-time compilation (JIT), batched parallelism (via `vmap`), and functional key splitting for pseudorandom number generation (PRNG). To avoid race conditions involving calculation of the energy within the neighborhood of each site, we parallelize each step by considering lattice positions spaced modulo four and synchronize all accepted moves each step. Note that each step updates the entire lattice and each move within a step proposes an independent exchange (of position or identity).

PRNG and data-flow structure. At the start of each full step, a master PRNG key is split into subkeys to select the candidate positions of the 'reference grid', pick per-site swap directions or replacement species, and draw Metropolis acceptance variates. All arrays of positions, energy calculations, or acceptance evaluations are computed under a single JIT-decorated function.

Single-mask strategy with offsets. We generate a 'reference grid' by partitioning the cubic lattice into discrete modulo *4×4×4 blocks* of sites:

$$\text{ref\_grid} \;=\; \big\{\,4(k,l,m) \mid 0 \le l < L/4,\; 0 \le l < L/4,\; 0 \le l < L/4\big\}, \tag{S52}$$

so that no two reference points share an edge. From this 'reference grid' we can generate a shared 'offset grid' by applying a global offset to one of the 64 possible offset positions within each 4^3 cube.

Generating positions at each MC Step. At each step, we randomly select one of the 64 offset grids to parallelize the MC moves. During a μVT step, we propose replacements at each site in the offset grid (excluding input species). During an NVT step, one of 26 neighbor vectors (unit step in any x, y, and/or z) is independently applied to each offset grid point. From these shifted points, we then draw a random permutation, split the points into equal halves, and pair them, such that each point appears in at most one pair. This process yields non-overlapping global swap proposals across the whole lattice.

Acceptance or rejection is then computed according to the Metropolis-Hastings criterion described above, independently and in parallel for all proposed swaps, and all accepted moves are synchronized across the lattice. This parallelized procedure ensures that every lattice site has an opportunity to update while respecting the non-periodic boundary conditions and the mixed canonical-grand-canonical setup.

Correctness versus efficiency. By using one unified, randomized mask with per-site offsets and directions, our procedure guarantees:

- *Correctness:* By construction, any two candidate sites generated from different reference points are at least two lattice steps apart, so edges cannot be shared, and updates commute exactly. No two sites ever race to read or write the same neighbor; counts cannot drift or desynchronize. Boundary sites (hard walls) simply have fewer neighbor offsets.
- *Efficiency:* The entire nested-scan loop over all MC steps is traced once into a single XLA computation—there are no host-side Python loops or repeated JIT invocations. All random draws, vectorized grid updates, and other per-site operations are executed in one fused GPU kernel via `vmap`, giving efficient parallel throughput and minimizing host/device synchronization and overhead.

Together, this approach delivers robust statistical correctness (no hidden synchronization bugs or particle number errors) and optimized performance on modern hardware.

Deriving the mean-field model from the lattice liquid formulation

In this section, we establish a correspondence between the lattice model and the mean-field model discussed in the paper. Briefly, the lattice model defines an energy for each lattice configuration (or microstate). In the mean-field limit, we consider sets of configurations that share the same species counts (or macrostates). In what follows, we show that eq. 5 arises from the macrostate energies in the bulk limit, under certain assumptions. Furthermore, we relate the lattice model parameters ϵ_{ij} and γ_i to the mean field parameters χ_{ij} and $\mu_{\text{res},i}$. Except at the very end of this section, we make no assumptions constraining the possible values of ϵ_{ij} and γ_i. This is a standard treatment, included here with consistent terminology only as a convenience for the reader.

The lattice is a set R of positions, with $\|R\| = S$, such that the total volume of R is $S\nu$, where ν is the volume per position. For an $L \times L \times L$ cubic lattice, $S = L^3$. We use $\eta(\mathbf{p})$ to denote the neighbors of position $\mathbf{p}$, with $||\eta(\mathbf{p})|| = z$ being the effective valence of each particle. For a system with N distinct solute species and 1 solvent species, a microstate configuration is σ, where $\sigma(\mathbf{p}) \in \{0, \ldots, N\}$ is the species of the particle at position $\mathbf{p}$ and 0 indexes the solvent.

The energy of the lattice when in microstate σ is

$$H(\sigma) = \frac{1}{2} \sum_{\mathbf{p} \in R} \sum_{\mathbf{q} \in \eta(\mathbf{p})} \epsilon_{\sigma(\mathbf{p}),\sigma(\mathbf{q})} + \sum_{\mathbf{p} \in R} \gamma_{\sigma(\mathbf{p})} \tag{S53}$$

$$= \sum_{i=0}^{N} \sum_{j=i}^{N} n_{ij} \epsilon_{ij} + \sum_{i=0}^{N} n_i \gamma_i \tag{S54}$$

where $\epsilon_{ij} = \epsilon_{ji}$ is the microscopic nearest-neighbor contact energy between species i and species j, and γ_i relates to the reservoir chemical potential of species i. The number of i particles (a Delta function sum over all lattice sites) and $i:j$ interfaces are, respectively,

$$n_i = \sum_{\mathbf{p} \in R} \delta_{i,\sigma(\mathbf{p})} \tag{S55}$$

$$n_{ij} = \frac{1}{1+\delta_{ij}} \sum_{\mathbf{p} \in R} \sum_{\mathbf{q} \in \eta(\mathbf{p})} \delta_{i,\sigma(\mathbf{p})}\, \delta_{j,\sigma(\mathbf{q})} \quad . \tag{S56}$$

Note that the factor of $1/(1+\delta_{ij})$ in the second sum ensures interfaces are not double counted for each pair of *positions* when $i=j$. To compute the sums in $H(\sigma)$ symmetrically, we rewrite eq. (S54) as

$$H(\sigma) = \frac{1}{2}\sum_{i=0}^{N}\sum_{j=0}^{N} n_{ij}\epsilon_{ij}(1+\delta_{ij}) + \sum_{i=0}^{N} n_i\gamma_i \tag{S57}$$

$$= S\left(\frac{1}{2}\sum_{i\neq j}\frac{n_{ij}}{S}\epsilon_{ij} + \sum_i \frac{n_{ii}}{S}\epsilon_{ii} + \sum_i \frac{n_i}{S}\gamma_i\right) \quad . \tag{S58}$$

The sum above has a prefactor of 1/2 for the total pair contact energies since, unlike eq. S54, the sum double counts over all pairs of distinct species $i \neq j$.

Now consider a macrostate $M_{\vec{n}}$ consisting of all microstates whose counts of species i are n_i. As the Monte Carlo sampling satisfies detailed balance with respect to H and the state space is fully connected, at equilibrium the probabilities of microstates and macrostates will obey the Boltzmann distribution:

$$P(\sigma) = \frac{1}{Z}e^{-H(\sigma)/k_\mathsf{B}T} \qquad \text{where} \qquad Z = \sum_\sigma e^{-H(\sigma)/k_\mathsf{B}T} \tag{S59}$$

$$P(M_{\vec{n}}) = \sum_{\sigma\in M_{\vec{n}}} P(\sigma) = \frac{1}{Z}e^{-G(M_{\vec{n}})/k_\mathsf{B}T} \qquad \text{where} \qquad G(M_{\vec{n}}) = -k_\mathsf{B}T\ln\left(\sum_{\sigma\in M_{\vec{n}}} e^{-H(\sigma)/k_\mathsf{B}T}\right) \quad . \tag{S60}$$

We define $\phi_i = \frac{n_i}{S}$ to be the volume fraction of species i and note that for well-mixed states, $\frac{n_{ij}}{S} \approx z\frac{n_i}{S}\frac{n_j}{S} = z\phi_i\phi_j$ when $i \neq j$, and otherwise $\frac{n_{ii}}{S} \approx \frac{z}{2}\phi_i^2$. Such states σ all have similar energy

$$H(\sigma) \approx S\left(\frac{z}{2}\sum_{i\neq j}\phi_i\phi_j\epsilon_{ij} + \frac{z}{2}\sum_i \phi_i^2\epsilon_{ii} + \sum_i \phi_i\gamma_i\right) \tag{S61}$$

$$= S\left(\frac{z}{2}\sum_{i,j}\phi_i\phi_j\left(\epsilon_{ij} - \frac{\epsilon_{ii}+\epsilon_{jj}}{2}\right) + \frac{z}{2}\sum_i \phi_i\epsilon_{ii} + \sum_i \phi_i\gamma_i\right) \quad . \tag{S62}$$

In the mean-field limit, we assume that these well-mixed states dominate the free energy, and that the number of such states is approximately $\|M_{\vec{n}}\|$, which we can estimate using Stirling's approximation that $\ln n! \approx n\ln n/e$, so

$$\ln\|M_{\vec{n}}\| = \ln\binom{S}{\vec{n}} \tag{S63}$$

$$= \ln\frac{S!}{\prod_{i=0}^{N} n_i!} \tag{S64}$$

$$\approx S\ln S/e - \sum_i n_i\ln n_i/e \tag{S65}$$

$$= S\left(\ln S/e - \sum_i \phi_i\ln S\phi_i/e\right) \tag{S66}$$

$$= -S\sum_{i=0}^{N}\phi_i\ln\phi_i \quad . \tag{S67}$$

The free energy for this macrostate of the lattice is therefore

$$G(M_{\vec{n}}) \approx -k_\mathsf{B}T\ln\|M_{\vec{n}}\|e^{-H(\sigma)/k_\mathsf{B}T} \tag{S68}$$

$$\approx H(\sigma) + k_\mathsf{B}TS\sum_i \phi_i\ln\phi_i \tag{S69}$$

and the (non-dimensionalized) free energy of the lattice per unit volume is

$$\Omega_{\text{surface}} \equiv \beta\nu \frac{G(M_{\vec{n}})}{\nu S} \tag{S70}$$

$$\approx \beta \frac{H(\sigma)}{S} + \sum_{i=0}^{N} \phi_i \ln \phi_i \tag{S71}$$

$$= \beta \left(\frac{z}{2} \sum_{i=0}^{N} \sum_{j=0}^{N} \phi_i \phi_j \left(\epsilon_{ij} - \frac{\epsilon_{ii} + \epsilon_{jj}}{2} \right) + \frac{z}{2} \sum_{i=0}^{N} \phi_i \epsilon_{ii} + \sum_{i=0}^{N} \phi_i \gamma_i \right) + \sum_{i=0}^{N} \phi_i \ln \phi_i \tag{S72}$$

$$= \frac{1}{2} \sum_{i=0}^{N} \sum_{j=0}^{N} \phi_i \chi_{ij} \phi_j + \sum_{i=0}^{N} \phi_i \ln \phi_i - \beta \sum_{i=0}^{N} \phi_i \mu'_{\text{res},i} \tag{S73}$$

$$= \beta\nu f(\vec{\phi}, \chi) - \beta \vec{\mu}'_{\text{res}} \cdot \vec{\phi} \tag{S74}$$

where $\beta = 1/k_{\text{B}}T$ and $\chi_{ij} = \beta z(\epsilon_{ij} - \frac{1}{2}(\epsilon_{ii} + \epsilon_{jj}))$ and $\mu'_{\text{res},i} = -(\gamma_i + \frac{z}{2}\epsilon_{ii})$. We used the fact that $\phi_0 = 1 - \phi_T$ is the solvent volume fraction and $\chi_{i0} = 0$ by construction to equate the first term with eq. 6. Finally, since the input species are non-exchanging, and recalling that $\mu'_{\text{res},i}$ is the reservoir chemical potential of species i, with $\vec{\mu}'_{\text{res}} = 0 \circ \vec{\mu}^{(\text{in})}_{\text{res}} \circ \vec{\mu}_{\text{res}}$ defined in SI Note 1, we have that

$$\vec{\mu}'_{\text{res}} \cdot \vec{\phi} = \vec{\mu}_{\text{res}} \cdot \vec{\phi}_{\text{oh}} + \text{const.} \tag{S75}$$

and so, up to a constant,

$$\Omega_{\text{surface}} = \beta\nu f(\vec{\phi}, \chi) - \beta \vec{\mu}_{\text{res}} \cdot \vec{\phi}_{\text{oh}} \tag{S76}$$

is in agreement with eq. 5. Note that because elsewhere in this work we assume an inert solvent and non-self-interacting solute monomers, such that $\chi_{i0} = 0$, lattice simulations are run with $\epsilon_{ii} = 0 = \epsilon_{0i}$, so $\epsilon_{ij} = \frac{\chi_{ij}}{\beta z}$ for $z = 18$ and $\gamma_i = -\mu'_{\text{res},i}$.

Supplementary Note 5: Analyses

Phase number and composition calculation

The steady-state compositions of the n_{set} surfaces from the mean-field dynamics are gathered into a matrix $B = n_{\text{set}} \times (N_{\text{out}} + N_{\text{h}})$, and given the large number of surfaces, we generically assume $N_{\text{out}} + N_{\text{h}} \ll n_{\text{set}}$. Subsequently, the matrix is normalized (mean-centered and standard-deviation set to 1) and the covariance matrix's eigenvalues (i.e. eigenvalues of $\frac{B^T B}{N_{\text{out}}+N_{\text{h}}}$) is computed. If the normalized matrix was populated purely with i.i.d values from $N(0, \sigma = 1)$, the Marchenko-Pastur distribution (17) guarantees that the eigenvalues would be smaller than $\lambda = \left(1 + \sqrt{\frac{N_{\text{out}}+N_{\text{h}}}{n_{\text{set}}}}\right)^2$. Thus, eigenvalues larger than this are unlikely to arise from compositions sampled randomly around a typical composition (i.e. of a particular phase) and when no eigenvalues are significant, we assume that there is only 1 typical phase composition. Note that this is an approximation since the MP distribution does not generically guarantee that eigenvalues from "signal" cannot be less than the above λ, and only that the eigenmodes from "noise" cannot be larger—so the number of phases we estimate may be lower than actually present. With that caveat, we use the number of significant modes (larger than above threshold) to estimate number of phases as $n_{\text{phases}} = n(\text{eig} > \lambda) + 1$. With this estimate, we employ a hierarchical clustering method to group the n_{set} surfaces into n_{phases} compositions. The average composition of each phase is computed as the mean composition of all the surfaces clustered into the same phase and reported in Fig. 4.

Phase detection and composition analysis in lattice

To identify distinct phases and their compositions from equilibrium lattice configurations, we employ a composition-based clustering algorithm as described below. Conceptually, the process includes smoothing particle numbers into normalized compositions and performing unsupervised clustering to identify bulk phase compositions and volume fractions. In Fig. S14, this methodology is used to analyze that phase composition for the AND decision boundary with 2 hidden species at each surface.

Before clustering, we convert each discrete lattice snapshot into a matrix P (of size $N_{loc} \times N$ where N_{loc} is the number of *bulk* lattice positions as defined below and N is the number of species including solutes and the solvent) that describes composition at each bulk location as follows.

1. For each species i, we compute a smoothed density field $\rho_i(\mathbf{r})$ by applying a Gaussian filter with width $\sigma = 2.0$ lattice units to the binary indicator field (1 where species i is present, 0 elsewhere). The Gaussian convolution is implemented using `scipy.ndimage.gaussian_filter` specifically to handle non-periodic boundary conditions by extending edge values outward, appropriate for the closed boundary simulations.

2. To focus on phase identity rather than local density variations, we normalize the smoothed densities at each lattice position $\mathbf{r}$ to obtain composition fields:

$$c_i(\mathbf{r}) = \frac{\rho_i(\mathbf{r})}{\sum_j \rho_j(\mathbf{r}) + \epsilon} \tag{S77}$$

   where $\epsilon = 10^{-10}$ prevents division by zero and j runs from 0 to N . This normalization removes the magnitude axis from the feature space, ensuring that regions with identical composition but different total densities are recognized as the same phase, helping to identify small pockets of distinct phases within a larger bulk phase.

3. We identify bulk (low-gradient) regions suitable for phase analysis by computing the gradient magnitude of the composition field and applying a percentile-based threshold. For each species, we compute ∇c_i using central differences and define the total gradient magnitude as $g(\mathbf{r}) = \max_i \|\nabla c_i(\mathbf{r})\|^2$. Voxels are retained in the bulk mask in the bulk mask if $g(\mathbf{r}) < g_P$, where g_P is the Pth percentile of gradient values. In this analysis, we retain voxels below the 85th percentile of gradient values. Additionally, voxels with total density below 0.05 are excluded to avoid spurious phase detection in nearly empty regions. The included positions make up the N_{loc} that go into the matrix P.

From the matrix P, we next perform unsupervised clustering to identify the number, composition, and volume fraction of different coexisting phases as below. Note that this unsupervised clustering requires choices of hyperparameters that were found after benchmarking on lattices with known phase counts and compositions.

1. The number of phases K is determined via Bayesian Information Criterion (BIC) model selection. We fit Gaussian Mixture Models (GMM) to the bulk voxel composition vectors for $K = 1, 2, \ldots, 20$ using scikit-learn's `GaussianMixture`. For each K, the BIC is computed as $\text{BIC}(K) = -2\ln(L) + k\ln(n)$, where L is the maximum log-likelihood, k is the number of model parameters, and n is the number of data points. The optimal K is

identified using the elbow method: we compute the second derivative of the BIC curve and select the K where this quantity is maximal, indicating the point where additional clusters provide diminishing improvement in model fit.

2. With K determined, we cluster the bulk voxel composition vectors using GMM with the selected number of components. Each voxel is assigned to the cluster with highest posterior probability. The composition of each phase α is computed as the mean composition of all bulk voxels assigned to that cluster, normalized to sum to unity.

3. To prevent over-clustering due to noise or minor compositional fluctuations, we compute pairwise cosine similarities between all phase compositions: $\text{sim}(\alpha, \beta) = \mathbf{c}_\alpha \cdot \mathbf{c}_\beta / (\|\mathbf{c}_\alpha\| \|\mathbf{c}_\beta\|)$. Phase pairs with similarity exceeding 0.85 are merged, with the merged phase composition computed as the weighted average of the constituent phases. Merging continues iteratively until no pair exceeds the threshold.

Supplementary Note 6: Random fluids

Fluids with random collection of interactions

We explore whether fluids with a random interaction network (as reported in Fig. 5) can be trained to classify distinct decision boundaries by simply tuning concentrations. For this, we first initialize a system with 2 inputs, 2 outputs, and a large ensemble ($N_{\mathsf{h}} = 30$ in Fig. 5B) of hidden species. In a given trajectory, the relevant pairwise interaction (χ_{ij}) are directly sampled as follows: first a random variable x is sampled uniformly from $[-1, 1]$ and transformed to obtain $\chi_{ij} = \chi_{\mathsf{m}} \tanh(x)$, where $\chi_{\mathsf{m}} = 15$ is chosen to set a maximum strength of interactions $\chi_{\mathsf{max}} \approx 12$. This resulting transformed distribution is not perfectly uniform and is biased a bit towards higher values of χ. The output-output interactions are set to favor demixing as above. With this initial *frozen* interaction matrix, we perform training as described above, except over 3000 epochs, to minimize the loss by only changing the reservoir potential $\vec{\mu}_{\mathsf{res}}$. Since the interaction matrix is sampled randomly and frozen, we repeat this training across 30 replicates and for distinct decision boundaries. The results of these tests are presented in the manuscript.

Supplementary Note 7: Sharp edges of the model

Reservoir

A central assumption of the model is that the trained reservoir potential ($\vec{\mu}_{\text{res}}$) will be maintained by the cellular milieu, likely through out-of-equilibrium mechanisms. Note that this assumption does not directly posit any further requirements of such a reservoir. That is, it could exist as a single or multiple coexisting phases, and either be dense or dilute—as long as the reservoir potential remains constant ($\vec{\mu}_{\text{res}}$) and unaffected by the exchange with surfaces. While not explicitly modeled in our study, we briefly discuss potential considerations in *designing* biological/physical models of reservoirs.

Biological reservoirs: The cellular millieu typically contains the same molecular repertoire but is generically coupled to various active processes. For example, molecules are routinely created and destroyed through active reactions, and cytoplasm/nucleoplasm resident molecules like chaperones and disaggregases (18, 19) contribute to partial solubilization of the reservoir. Thus, explicit models of the chemical potential remain challenging to describe.

Physically realizable reservoirs: In physical or synthetic systems, particularly those at equilibrium, one pertinent question relates to properties of the reservoir. In particular, what are its corresponding composition and stability? This requires a *specific* model of the reservoir. For example, one could allow for the *same* mean-field like treatment of the entropy/interactions for the reservoir as was used for the surface, except it could exist at a different, larger volume V_{res}. If we further assume that the reservoir is input-free—comprised of only hidden, output, and solvent species—one can invert the $\vec{0}$ input surface composition to get a reservoir composition from the Model A dynamics.

We discuss next how this inferred composition is guaranteed to be thermodynamically stable i.e., outside of the spinodal, and as such, will not spontaneously phase separate. This emerges because the criteria for the thermodynamic stability of the surface is $\frac{d^2\Omega_{\text{surface}}}{d\phi_i\phi_j} = \frac{\delta_{ij}}{\phi_i} + \frac{1}{\phi_T} + \chi_{ij}$ is positive semi-definite ($i,j \forall$ non-input species). This is guaranteed by construction, since the gradient descent procedure employed in the mean-field model finds local minima of Ω_{surface} that must satisfy this constraint. Importantly, the input-associated terms and linear reservoir terms do not explicitly show up in this Hessian. The above term is identical to the Hessian of the free energy that describes a box of finite volume comprising non-input only species at the identified steady-state composition. Physically, this can be interpreted as the stability of non-input species in a canonical ensemble, or in other words, following the $\beta\nu f$ like-term that we describe in eq. 6 only for the pertinent species. The lack of input-related terms, despite their contribution in the free-energy, stems from their constraints in the model. Since inputs are both clamped in space and position, $\phi_{\text{in},i}$ is not a free parameter that is capable of fluctuations. Thus, their interactions with non-input species can be re-interpreted as an (linear) "internal" chemical potential coupling, i.e., $(\chi_{ik}\phi_{in,i})\phi_k \equiv \mu_{ik}^{int}\phi_k \;\; \forall k \in (N_{\text{h}}, N_{\text{out}}), \; \forall i \in N_{\text{in}}$. An important caveat to note is that this Hessian does not guarantee stability of a mixture where inputs *also* contain translational entropy i.e. the ability to move in space. Although their counts are fixed, input species can now undergo spatial fluctuations, and thus can change the stability of the surface. Evaluating the stability of the whole surface requires determining: $\frac{d^2\Omega_{\text{surface}}}{d\phi_i d\phi_j}, \; i,j \forall (N_{\text{in}}, N_{\text{out}}, N_{\text{h}})$ - inherently not possible directly in the spatially unresolved mean-field model described here but could be studied by incorporating spatial gradient terms (as in eq. S1) leading to a Cahn-Hilliard type formulation or through lattice models. Note that this stability would also depend on input-associated parameters like input-input interactions, that are not directly learned or modified in our model. Preliminary investigations of lattice simulations with mean-field parameters, but with inputs no longer immobilized, suggest a loss in classifier performance as well as stronger intra-surface demixing. In such cases, since inputs strongly prefer distinct outputs and are still constrained to remain in the box, they demix to form pockets of coexisting phases with distinct outputs and compositions. These suggest the possibility of novel, or only partially overlapping, class of (microscopic) solutions may be discovered in a model where inputs are free to move within the surface but still incapable of exchanging with the reservoir—bearing resemblance to the model explored for MNIST classification by (20).

However, generally such an effective composition requires the multiple assumptions stated above. More generally, it may be experimentally advantageous to directly specify a desired reservoir composition ($\vec{\phi}^*_{\text{res}}$)—for instance, an equimolar, dilute reservoir. One could incorporate this constraint by suitably modifying our formulation to instead require that as the molecular interaction parameters χ evolve in the optimization procedure, the reservoir potential is implicitly derived as $\vec{\mu}_{\text{res}} = \vec{\mu}(\vec{\phi}^*_{\text{res}}, \chi)$ from the mean-field model. This will need to include an additional constraint that the Hessian matrix of the free-energy $H_{ij} = \frac{d\mu_{\text{res},i}}{d\phi_j}$ be positive semi-definite at $\vec{\phi}^*_{\text{res}}$ (21, 22). However, in both methods outlined here, there is still no guarantee that the reservoir composition is stable to fluctuation-driven nucleation.

Stability and properties of surface phases

In the model formulation, the surface is treated in the well-mixed mean-field limit. Thus, we don't explicitly consider whether the surface itself can demix *within* the volume that it occupies. In this section, we discuss the assumptions that underlie this model and where they may break down.

Biological motivation for mean-field treatment: We begin with the context presented in the paper i.e., DNA-bound transcription factors (TFs) as input species on genetic loci and mobile species (polymerases, cofactors etc.) that exchange with the nucleoplasm. DNA-bound TFs (inputs) are treated as fixed in position and space within our framework. This is motivated by the fact that the time-scales of free diffusion and exchange from the nucleoplasm of mobile molecules is significantly faster than for DNA-bound TFs. For example, the diffusion coefficients of chromatin, and thus molecules stably bound to it, are typically 2-3 orders of magnitude slower than those of nucleoplasmic proteins. We ignore any internal organization of the inputs within the surface that may emerge from the 3D DNA conformation and treat it as uniform, i.e., well-mixed. Thus the *mobile* species (hidden and outputs) in our model framework effectively live in a mean-field environment created by the well-mixed inputs. More generally, there may exist other active mechanisms like, for example, chromatin associated remodeler proteins that stir DNA, that may further contribute to keeping the input species well-mixed.

Stability of a surface: With the above assumption that inputs are effectively randomly well-mixed in the surface, the composition of the exchanging species (as queried by the Model A dynamics) is found as a minimum of the effective free-energy of the surface. This means the surface will not spontaneously phase separate but may still form multiple phases from nucleation. As described in the next section, we generally find that the explicit 3D lattice model shows a single phase in most regions except for the region adjacent to the decision boundary—see SI Notes 4 and 5 as well as Fig. S14.

Remarks from the lattice liquid model: In the lattice liquid (see SI Note 4), for each trajectory both the overall composition of the input species as well as their positions are held fixed to mimic immobile, non-exchanging TFs on short timescales. Note that the initial positions of the inputs are randomly assigned in the lattice. With this implementation, we find that parameters trained from the mean-field model successfully *translate* to 3D lattice fluids as measured by the classifier performance. This supports the idea that under the assumption of immobile, localized input species, the lattice model generally predicts a major, single phase within the surface. When closer to the decision boundary, we see that the lattice models deviate from the mean-field predictions (Fig. 7B). At these points, we empirically find that multiple phases can form *within* the surface that are enriched in the two distinct outputs.

Input-free surfaces: Biologically, the no-input surface is explicitly considered as a finite volume DNA loci that has *no* binding sites for any of the input molecules. Thus, the "output function" of a (0,0) surface is ascribed by condensing the appropriate output ('green' in AND, 'pink' in XOR, and so on). More generally, a surface absent of input species is nonetheless described by a fixed volume V that can freely exchange with the reservoir. As discussed above, in a (non-biological) physical reservoir that is not actively disaggregated and is constrained to a finite volume (i.e. a canonical ensemble), we expect the same condensed phase to emerge in the reservoir as in the (0,0) surface volume. More generally, if an input-free surface or "low-input" surface is desired to not recruit any outputs, a suitably altered loss will aid in discovery of parameters that achieve this. Fig S16A represents an AND gate where the system is explicitly trained (with an alternate loss) to not recruit any output in regions that are not high in either inputs, and thus, near the (0,0) region as well. Thus, the composition of the input-free surface would not recruit any outputs.

1. P. C. Hohenberg and B. I. Halperin. Theory of dynamic critical phenomena. *Reviews of Modern Physics*, 49(3):435–479, July 1977. doi: 10.1103/RevModPhys.49.435.
2. Krishna Shrinivas and Michael P. Brenner. Multiphase coexistence capacity in complex fluids. *bioRxiv*, October 2022. doi: 10.1101/2022.10.19.512909.
3. J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, April 1982. doi: 10.1073/pnas.79.8.2554.
4. James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, 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/google/jax.
5. DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem. 2020. URL: http://github.com/google-deepmind.
6. Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8.
7. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, November 2012. ISBN 978-1-118-58600-6.
8. Wolfgang Maass. On the computational power of winner-take-all. *Neural Computation*, 12(11):2519–2535, 2000.
9. Jongmin Kim, John Hopfield, and Erik Winfree. Neural Network Computation by In Vitro Transcriptional Circuits. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.
10. Anthony J. Genot, Teruo Fujii, and Yannick Rondelez. Computing with Competition in Biochemical Networks. *Physical Review Letters*, 109(20):208102, November 2012. doi: 10.1103/PhysRevLett.109.208102.
11. Anthony J. Genot, Teruo Fujii, and Yannick Rondelez. Scaling down DNA circuits with competitive neural networks. *Journal of The Royal Society Interface*, 10(85): 20130212, August 2013. doi: 10.1098/rsif.2013.0212.
12. Kevin M. Cherry and Lulu Qian. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature*, 559(7714):370–376, July 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0289-6.
13. Zibo Chen, James M. Linton, Shiyu Xia, Xinwen Fan, Dingchen Yu, Jinglin Wang, Ronghui Zhu, and Michael B. Elowitz. A synthetic protein-level neural network in mammalian cells. *Science*, 386(6727):1243–1250, December 2024. doi: 10.1126/science.add8468.
14. Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991. ISSN 0360-0300. doi: 10.1145/116873.116880.
15. R. O. Duda and H. Fossum. Pattern Classification by Iteratively Determined Linear and Piecewise Linear Discriminant Functions. *IEEE Transactions on Electronic Computers*, EC-15(2):220–232, April 1966. ISSN 0367-7508. doi: 10.1109/PGEC.1966.264302.
16. 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, May 2017. ISSN 1471-0080. doi: 10.1038/nrm.2017.7.
17. V. A. Marčenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, April 1967. ISSN 0025-5734. doi: 10.1070/SM1967v001n04ABEH001994.
18. Jared A. M. Bard and D. Allan Drummond. Chaperone regulation of biomolecular condensates. *Frontiers in Biophysics*, 2, May 2024. ISSN 2813-7183. doi: 10.3389/frbis.2024.1342506.
19. Arjun Narayanan, Anatoli Meriin, J Owen Andrews, Jan-Hendrik Spille, Michael Y Sherman, and Ibrahim I Cisse. A first order phase transition mechanism underlies protein aggregation in mammalian cells. *eLife*, 8:e39695, February 2019. ISSN 2050-084X. doi: 10.7554/eLife.39695.
20. 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 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:24, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-344-7. doi: 10.4230/LIPIcs.DNA.30.5.
21. Josiah Willard Gibbs. On the Equilibrium of Heterogeneous Substances. Other, 1879.
22. Richard P. Sear and José A. Cuesta. Instabilities in Complex Mixtures with a Large Number of Components. *Physical Review Letters*, 91(24):245701, December 2003. doi: 10.1103/PhysRevLett.91.245701.

Supplementary Figures: Combinatorial decision-making driven by multicomponent surface condensates

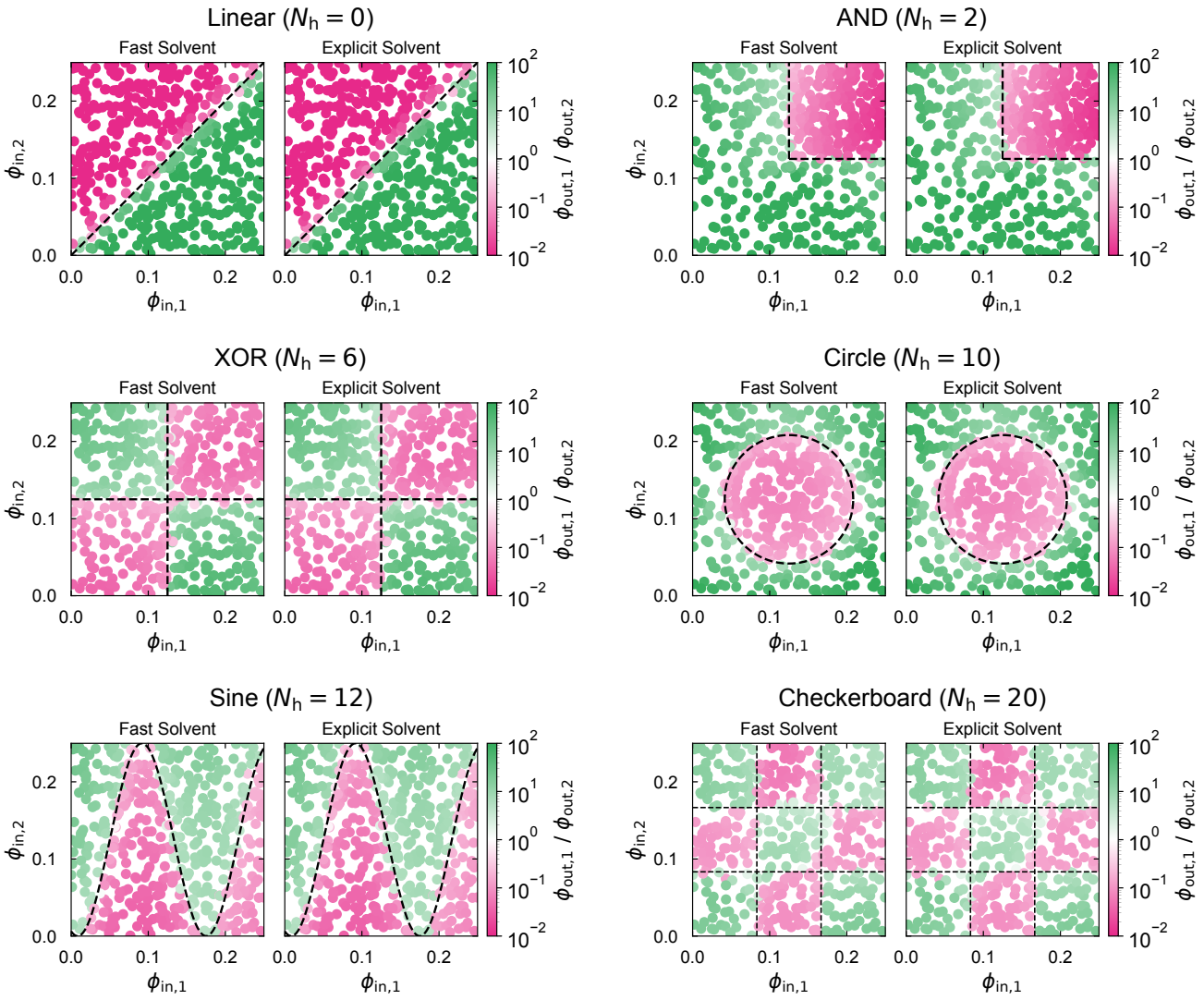

Fig. S1. In training and testing the classifiers in the manuscript, we assume that the solvent has fast dynamics and can therefore be treated implicitly according to the mass constraint of the system. However, the steady states of surfaces are largely insensitive to the choice of solvent dynamics, as shown above. For each of the decision boundaries tested in Figs 1-3 (reproduced here for ease of comparison as the "fast solvent" panels), we produce the same plot using dynamics in which the solvent is treated explicitly in the dynamics and is given the same mobility as the solutes (presented as the "explicit solvent" panels). The result is a decision boundary that looks nearly identical for all cases.

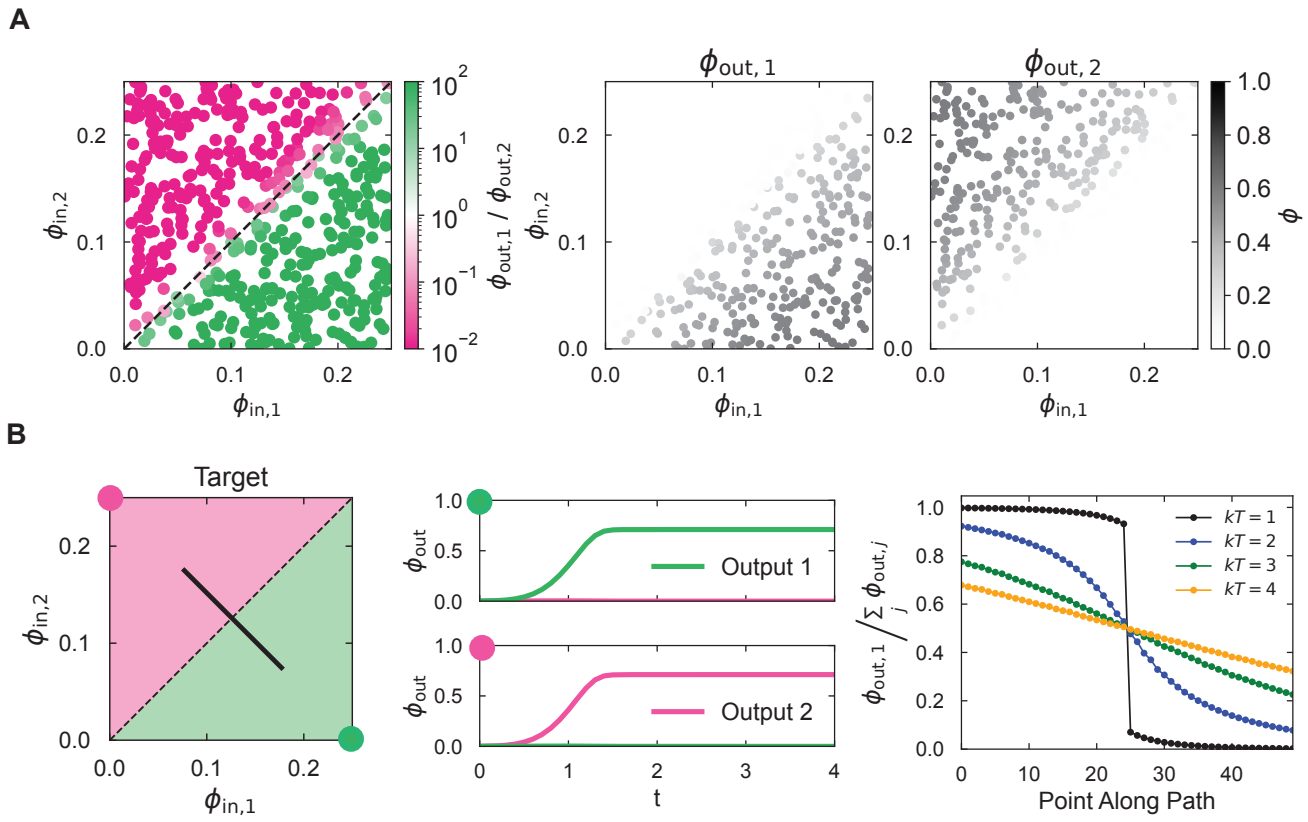

Fig. S2. **(A)** The absolute concentrations of the two output species, with the test prediction in the Fig. 2A reproduced on the left for comparison. **(B)** The middle panel shows the dynamics of the mean-field composition at two points (labelled by the green and pink dots) far away from the decision boundary. As an extension of Fig. 2B, the right panel depicts how the mean-field composition changes across the decision boundary at multiple temperatures.

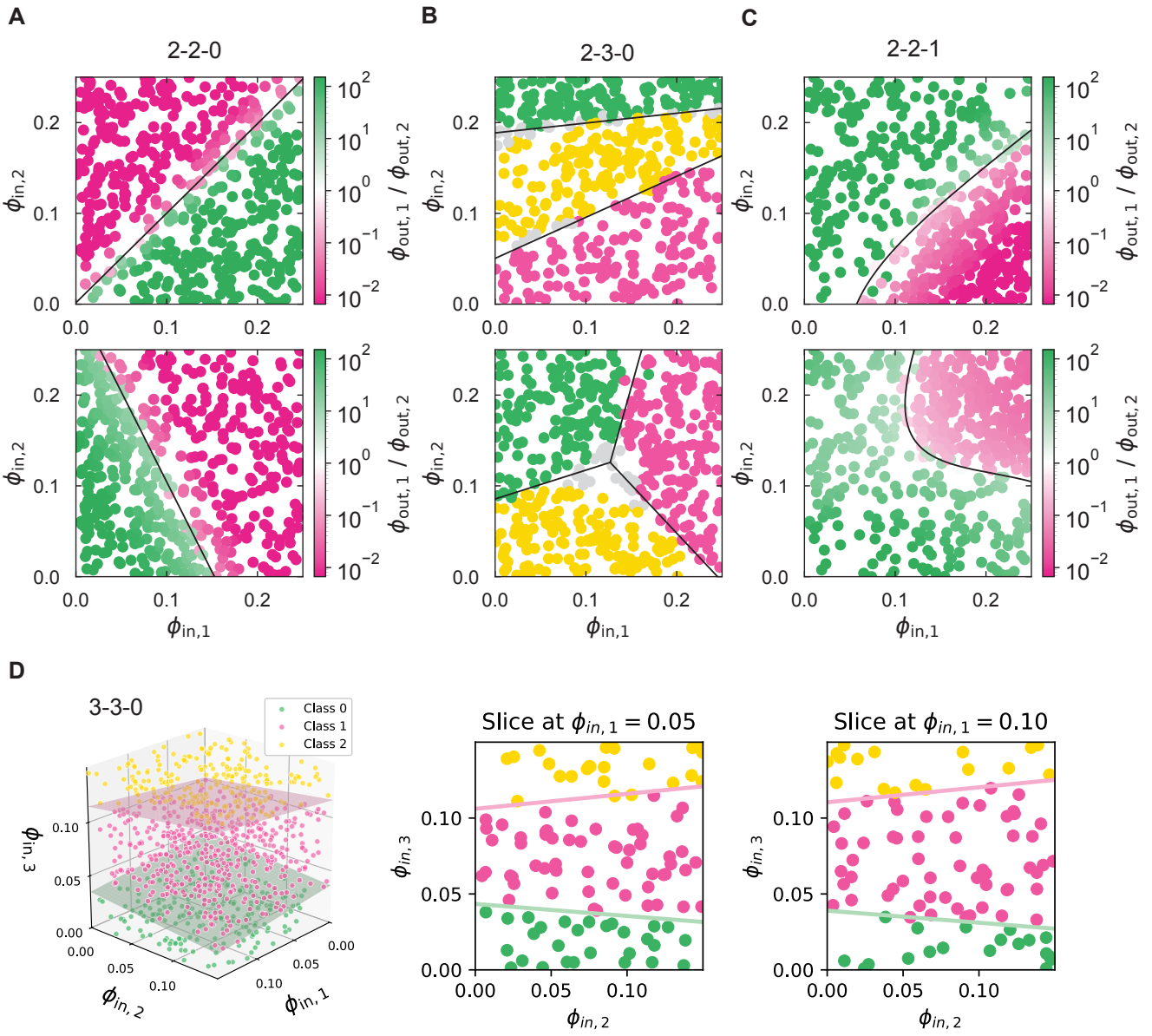

Fig. S3. **(A)** Two theoretical boundary solutions to systems with only 2 input and 2 output species (computed using eq. S28). Only linear boundaries are possible. **(B)** Two theoretical boundary solutions to systems with only 2 input and 3 output species (computed using eq. S30); gray points indicate surfaces for which none of the output species were recruited by a significant amount above others. Away from intersection points between more than two classes, the decision boundaries remain linear. **(C)** Two theoretical boundary solutions to systems with only 2 input, 2 output and 1 hidden species (computed using eq. S33), enabling the construction of nonlinear boundaries. **(D)** An example of a 3-input, 3-output, 0-hidden system, with classes separated by 2D planes. The 2D slices compare two different planes of constant concentration of species 1; as before, the solid lines are predictions of the decision boundary from theory, and each point is false-colored by its true class.

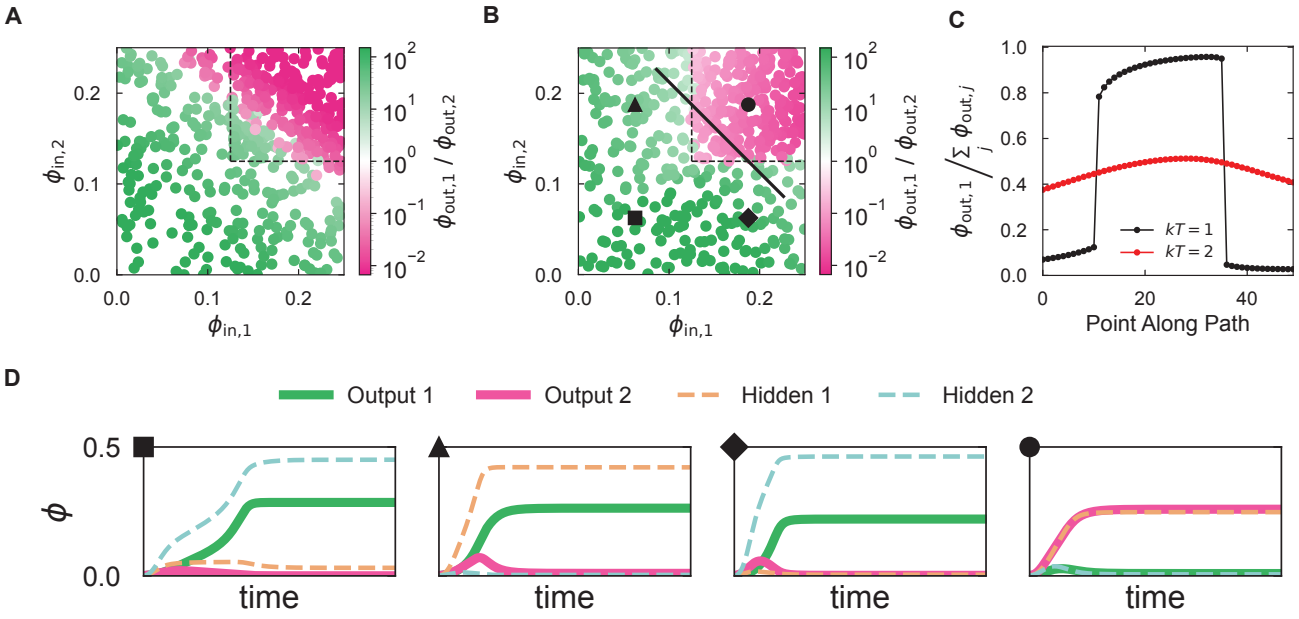

Fig. S4. (A) The optimal solution to an AND-like upper quadrant decision boundary with 0 hidden species, which is a best-fit linear cut of the boundary. **(B)** The 0-hidden solution in (A) can be compared with the solution using 2 hidden species, reproduced from Fig. 3C for convenience. **(C)** Moving along the solid black line and across the decision boundary in (B), we find that the system undergoes an abrupt transition in the recruited output species that is destroyed at higher temperatures. **(D)** The mean-field dynamics for the AND-like upper quadrant solution from a point in each of the quadrants of the input space.

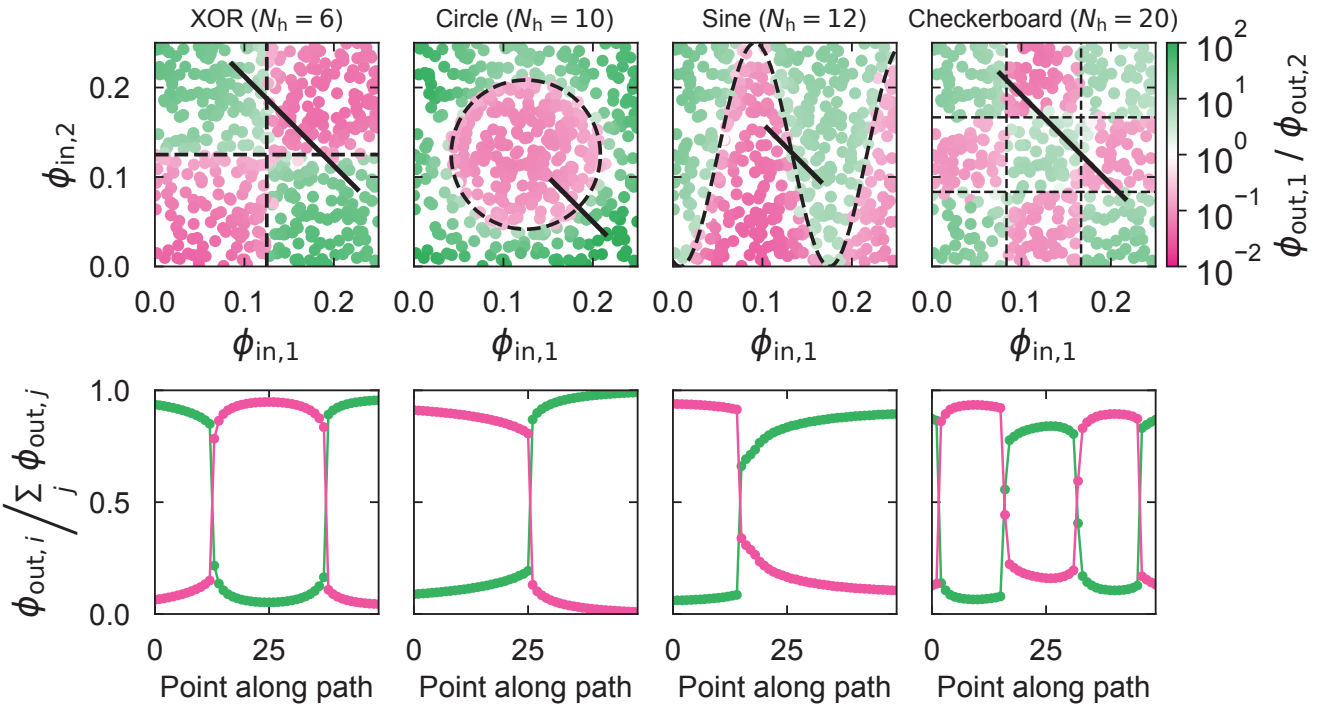

Fig. S5. The analogous simulations for the remaining 4 decision boundaries studied in Fig 3D. All show abrupt transitions in the order parameter across a decision boundary.

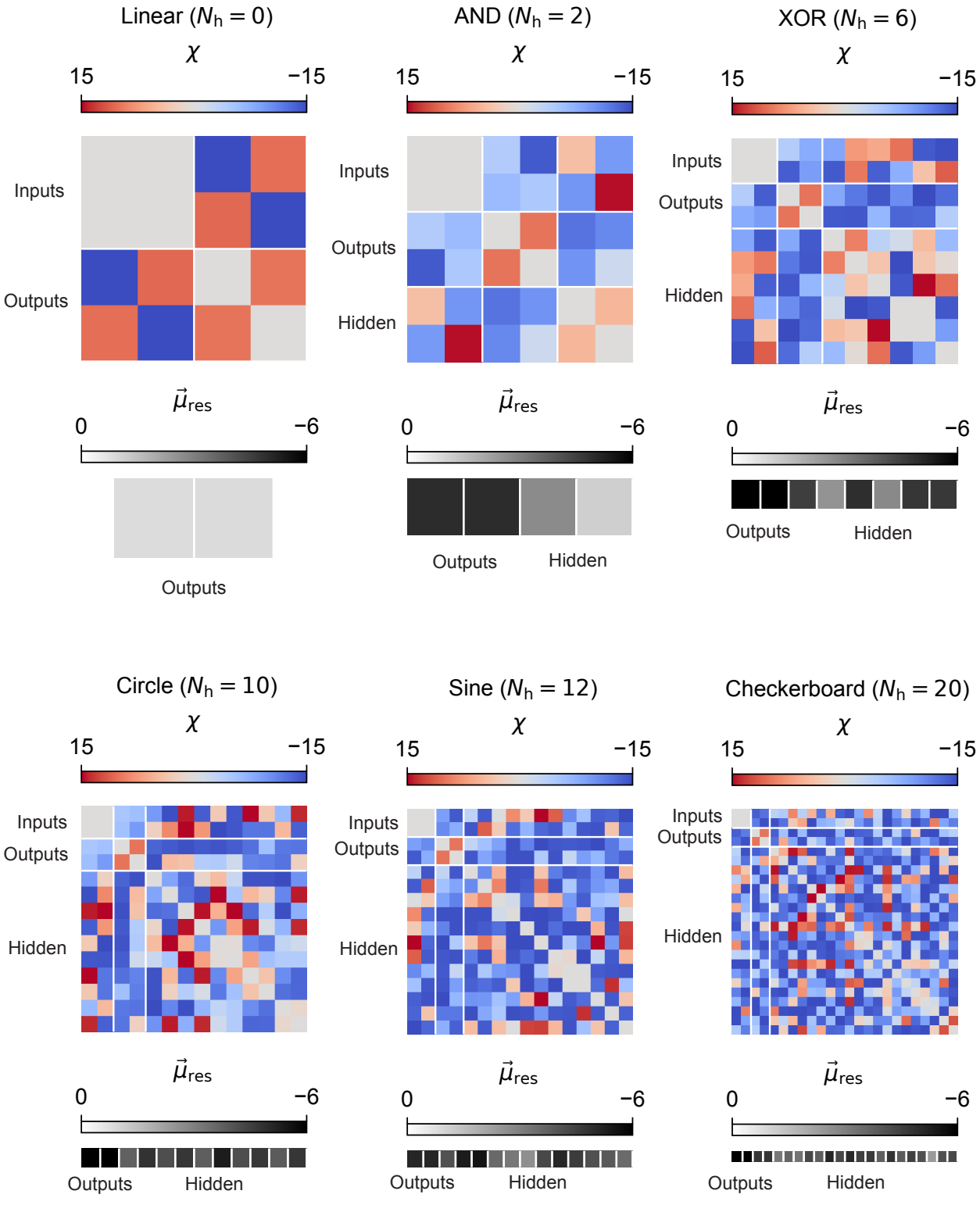

Fig. S6. The parameters for the trained networks that were used on the test data shown in Figs. 2 and 3.

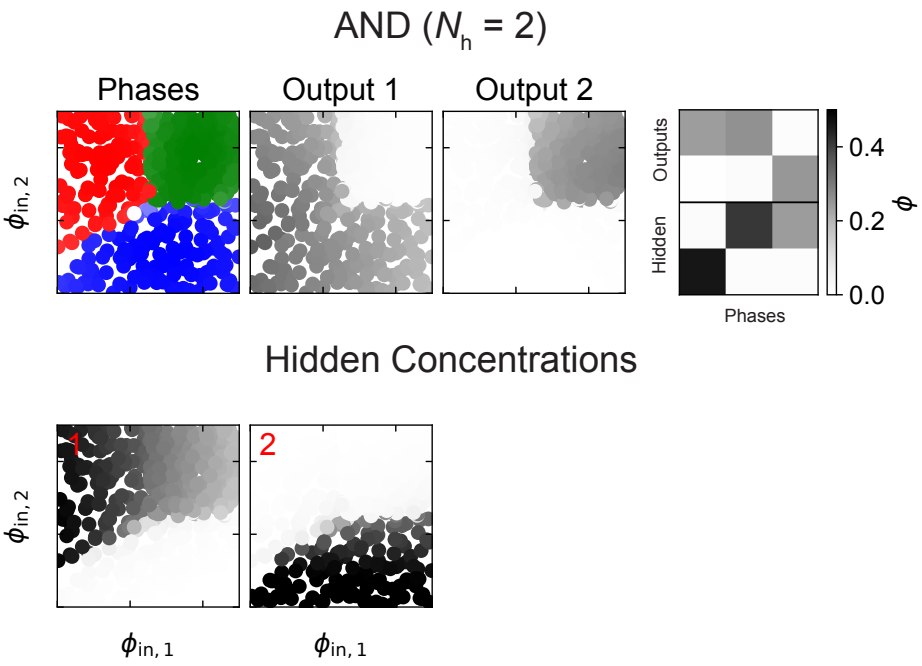

Fig. S7. The encrypted phases in the AND classifier and the volume fraction of each hidden species in the input space. The phase decomposition aligns with the decision boundary shown in Fig. 3C. Following the convention in Fig. 4, the color of a point in the phase panel indicates the phase that the surface exhibits, while the color intensity of the point indicates the cosine similarity between the surface's phase vector and the mean concentration vector of all surfaces within the phase.

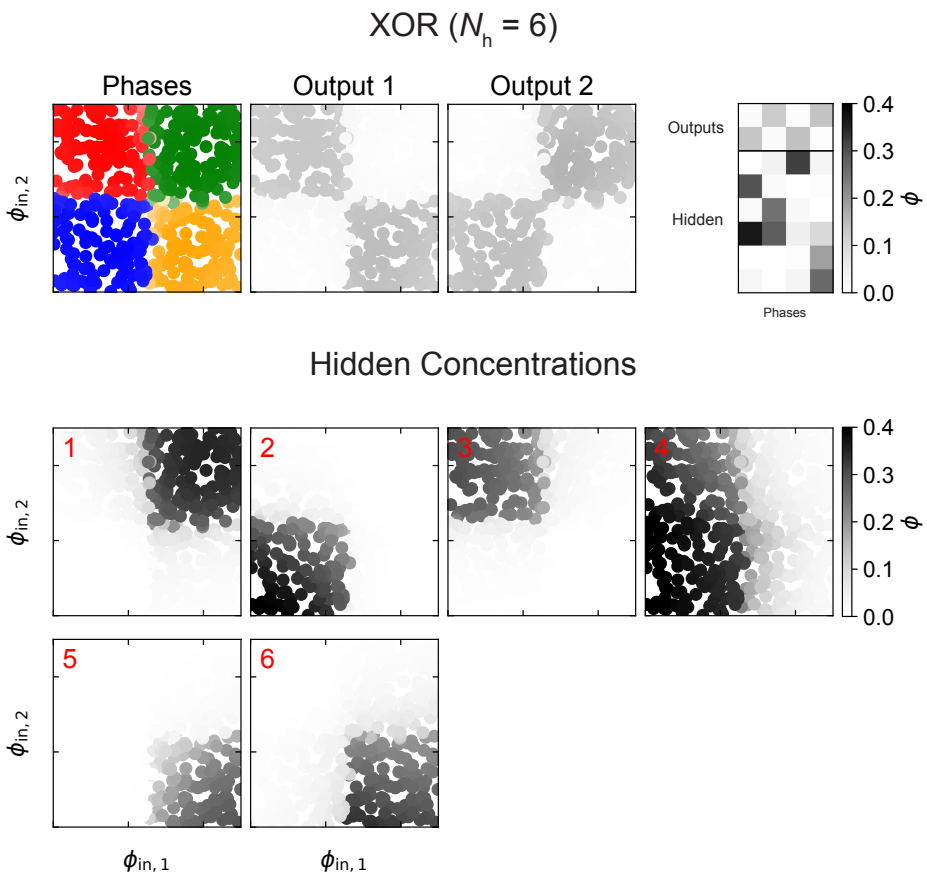

Fig. S8. The encrypted phases in the XOR classifier and the volume fraction of each hidden species in the input space. The phase decomposition aligns with the decision boundary shown in Fig. 3D. Following the convention in Fig. 4, the color of a point in the phase panel indicates the phase that the surface exhibits, while the color intensity of the point indicates the cosine similarity between the surface's phase vector and the mean concentration vector of all surfaces within the phase.

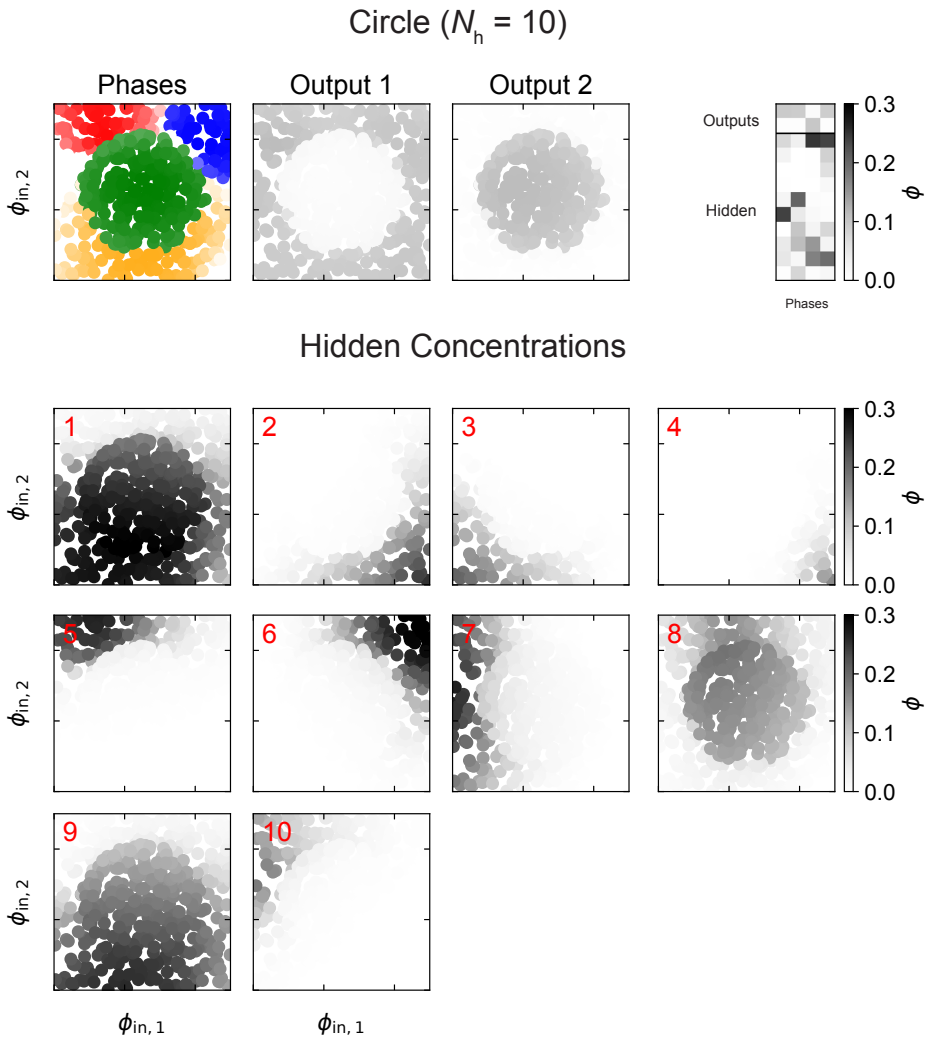

Fig. S9. The encrypted phases in the Circle classifier and the volume fraction of each hidden species in the input space. The phase decomposition aligns with the decision boundary shown in Fig. 3D. Following the convention in Fig. 4, the color of a point in the phase panel indicates the phase that the surface exhibits, while the color intensity of the point indicates the cosine similarity between the surface's phase vector and the mean concentration vector of all surfaces within the phase. Note that in this case, there is a mismatch in the phase decomposition and the concentration of constituent species in the yellow phase. This disagreement likely results from our choice of clustering algorithm—the Marchenko-Pastur distribution does not guarantee that there are no significant signals below its cutoff threshold, and can therefore underestimate the number of phases in the input space.

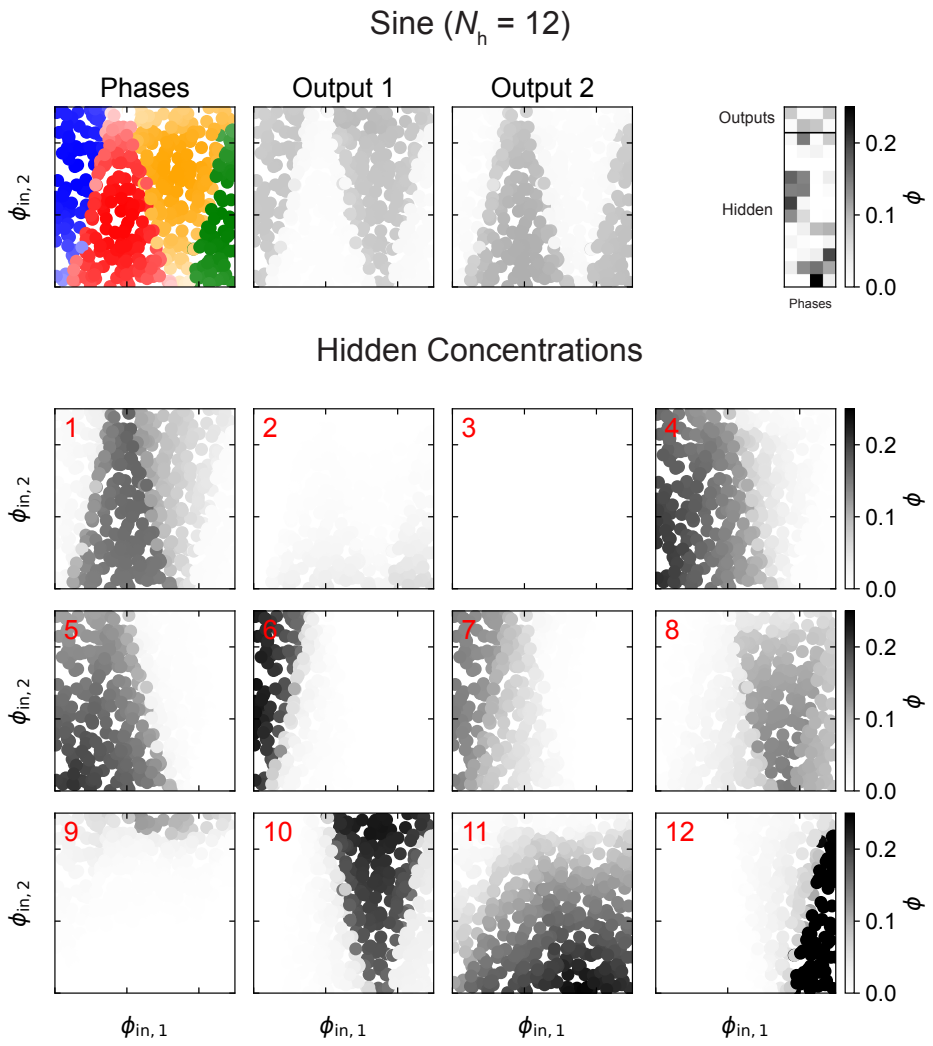

Fig. S10. The encrypted phases in the Sine classifier and the volume fraction of each hidden species in the input space. The phase decomposition aligns with the decision boundary shown in Fig. 3D. Following the convention in Fig. 4, the color of a point in the phase panel indicates the phase that the surface exhibits, while the color intensity of the point indicates the cosine similarity between the surface's phase vector and the mean concentration vector of all surfaces within the phase.

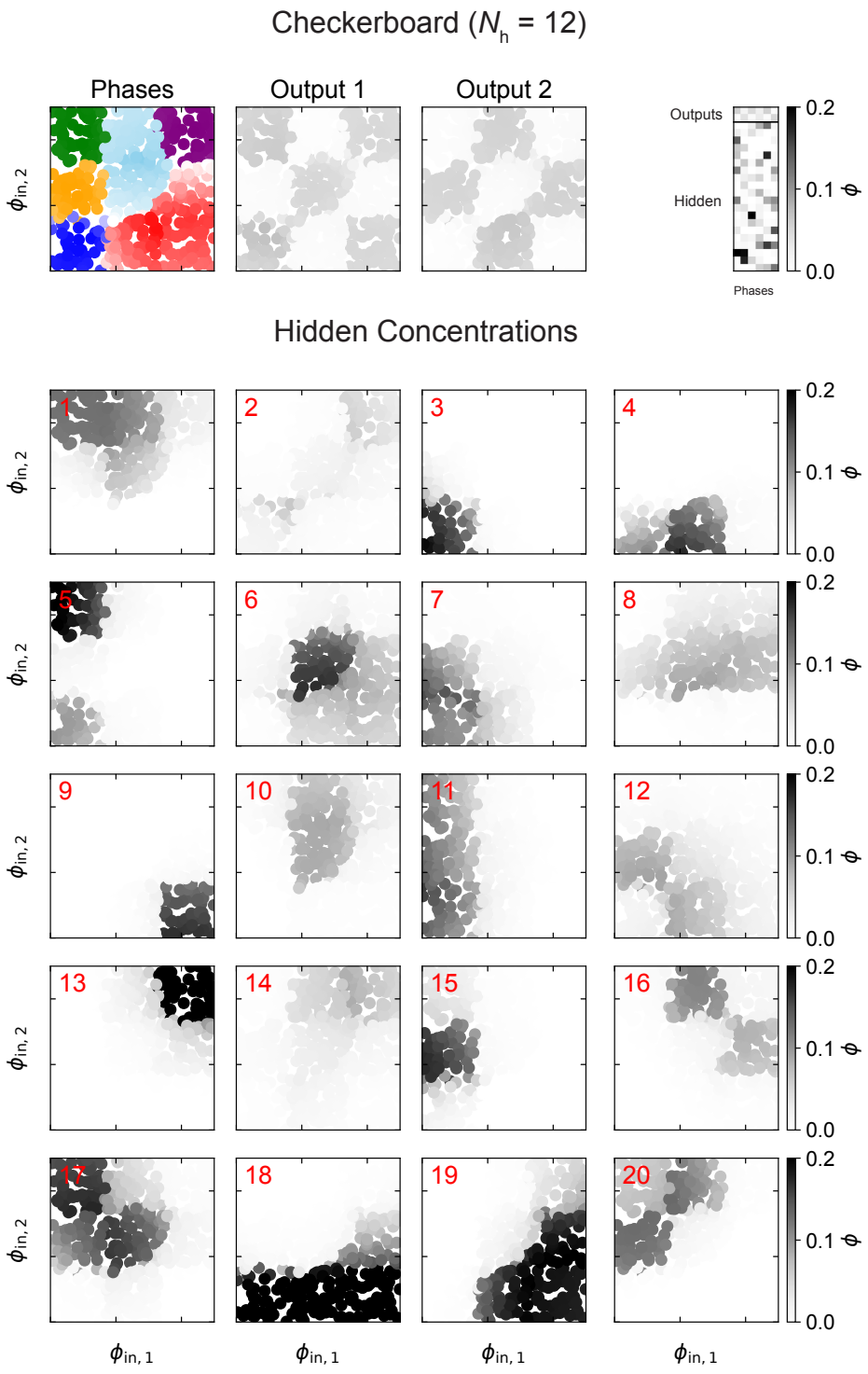

Fig. S11. The encrypted phases in Checkerboard classifier and the volume fraction of each hidden species in the input space. The input space has distinct phase compositions in its combination of hidden species. Following the convention in Fig. 4, the color of a point in the phase panel indicates the phase that the surface exhibits, while the color intensity of the point indicates the cosine similarity between the surface's phase vector and the mean concentration vector of all surfaces within the phase. The discrepancy between the phase decomposition and the decision boundary in Fig. 3D likely results from our choice of clustering algorithm—the Marchenko-Pastur distribution does not guarantee that there are no significant signals below its cutoff threshold, and can therefore underestimate the number of phases in the input space.

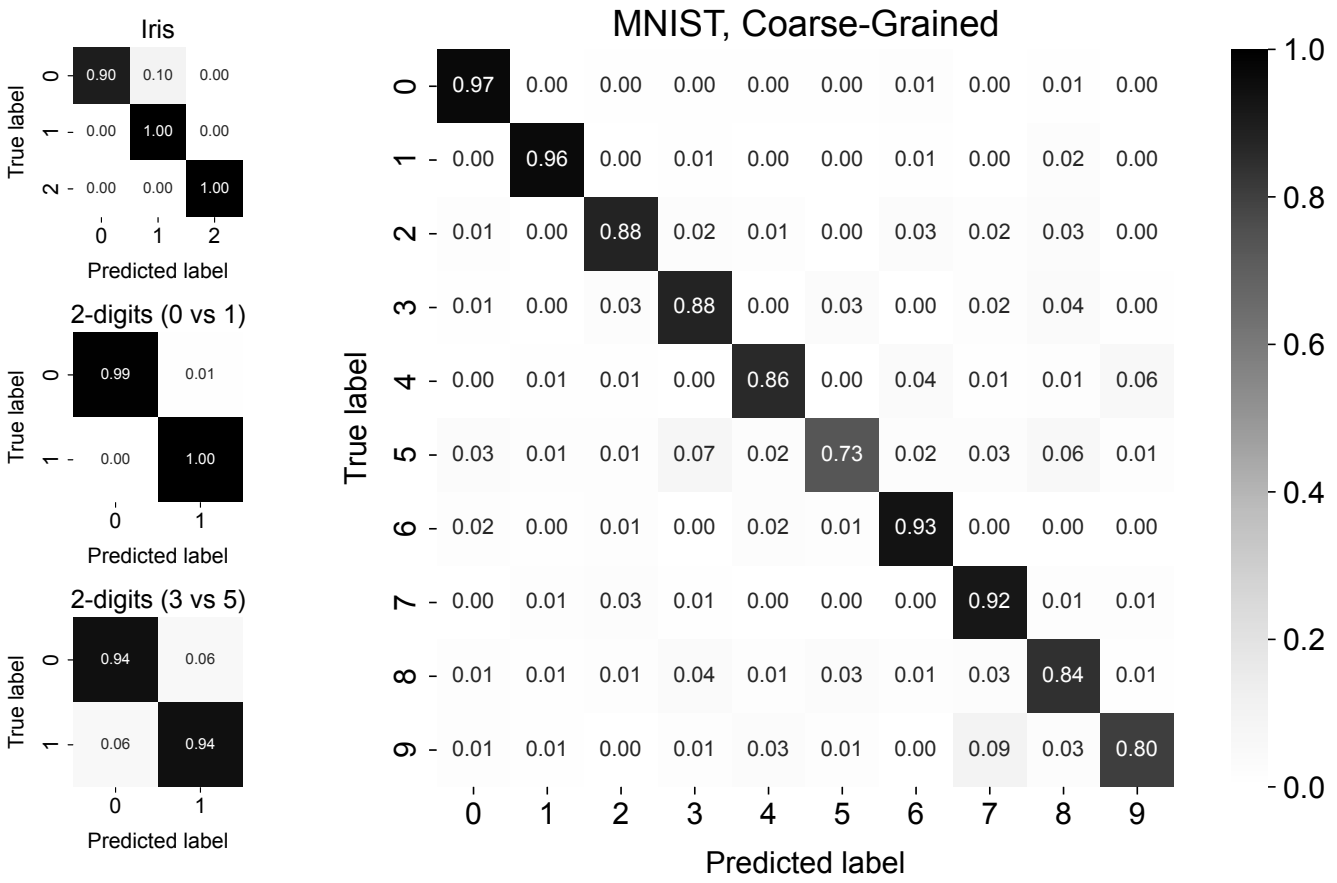

Fig. S12. Confusion matrices for traditional classification problems, where rows are the true class and columns are the predicted class. The confusion matrices show the proportion of correct predictions (diagonal entries) vs. incorrect predictions (off-diagonal entries) made by the molecular network for each of these cases. Left column, top to bottom: the Iris dataset (0 hidden species—classes 0, 1, and 2 denote setosa, versicolor and virginica, respectively), 2-digit 0 vs 1 MNIST (2 hidden species), 2-digit 3 vs 5 MNIST (10 hidden species), and full 10-digit MNIST (20 hidden species). Here, the success criterion is made more lenient as in Fig. 6. In the case of 10-digit MNIST, this success criterion yields an average correct percentage of 87%, the worst-performing digit being 5 at 73% and the best-performing digit being 0 at 97%.

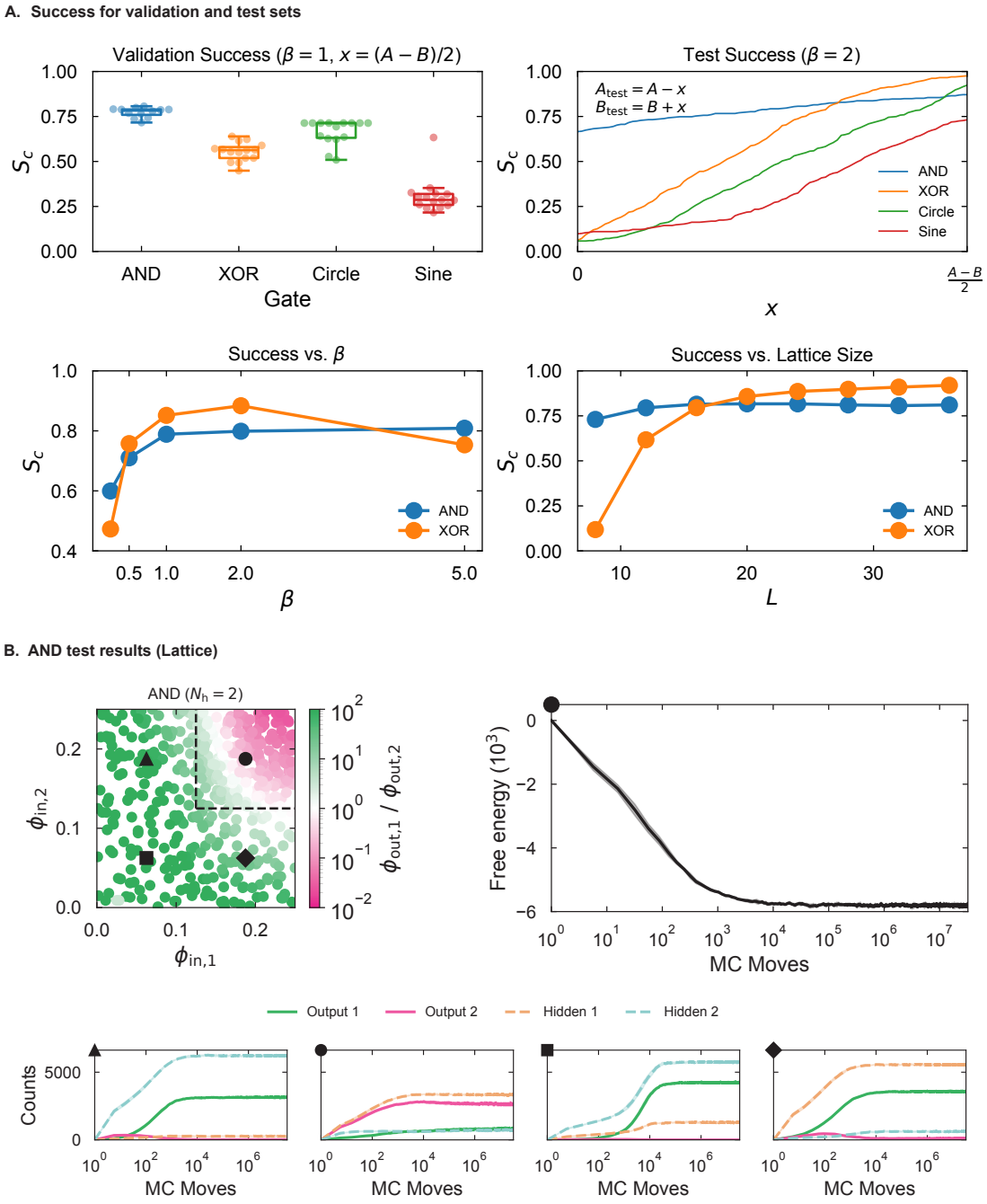

Fig. S13. (A) The success score S_c over the validation set for 15 separate mean-field solutions for AND, XOR, Circle, and Sine decision boundaries evaluated using the lattice liquid at $\beta = 1$ (top left). In plotting the validation success, we used the maximally lenient (or asymptotic) values of $A_{\text{test}} = B_{\text{test}} = (A+B)/2$, which allows for a clearer assessment of boundary correctness in blurry regions. The classification success on the test set improves with increasing threshold leniency x, to above $\sim 75\%$ (with several of the boundaries reaching $\sim 90\%$) when x is maximally lenient (top right); this test success uses $\beta = 2$, based on the temperature scan performed on the AND and XOR boundaries (bottom left), where we found that using lower temperatures (or, higher β) generally improved performance after translating the mean-field parameters to lattice parameters. We also confirm that the lattice size L does not meaningfully alter S_c for $L \gtrsim 24$. **(B)** The test results of the AND decision boundary at $\beta = 2$. The plot from Fig. 7 is reproduced, with the test point nearest to the center of that quadrant for which the dynamics of the corresponding surface are shown. For the point in the upper-right quadrant, we also show the free energy of the surface as the simulation progresses, confirming that the surface reaches a free energy minimum.

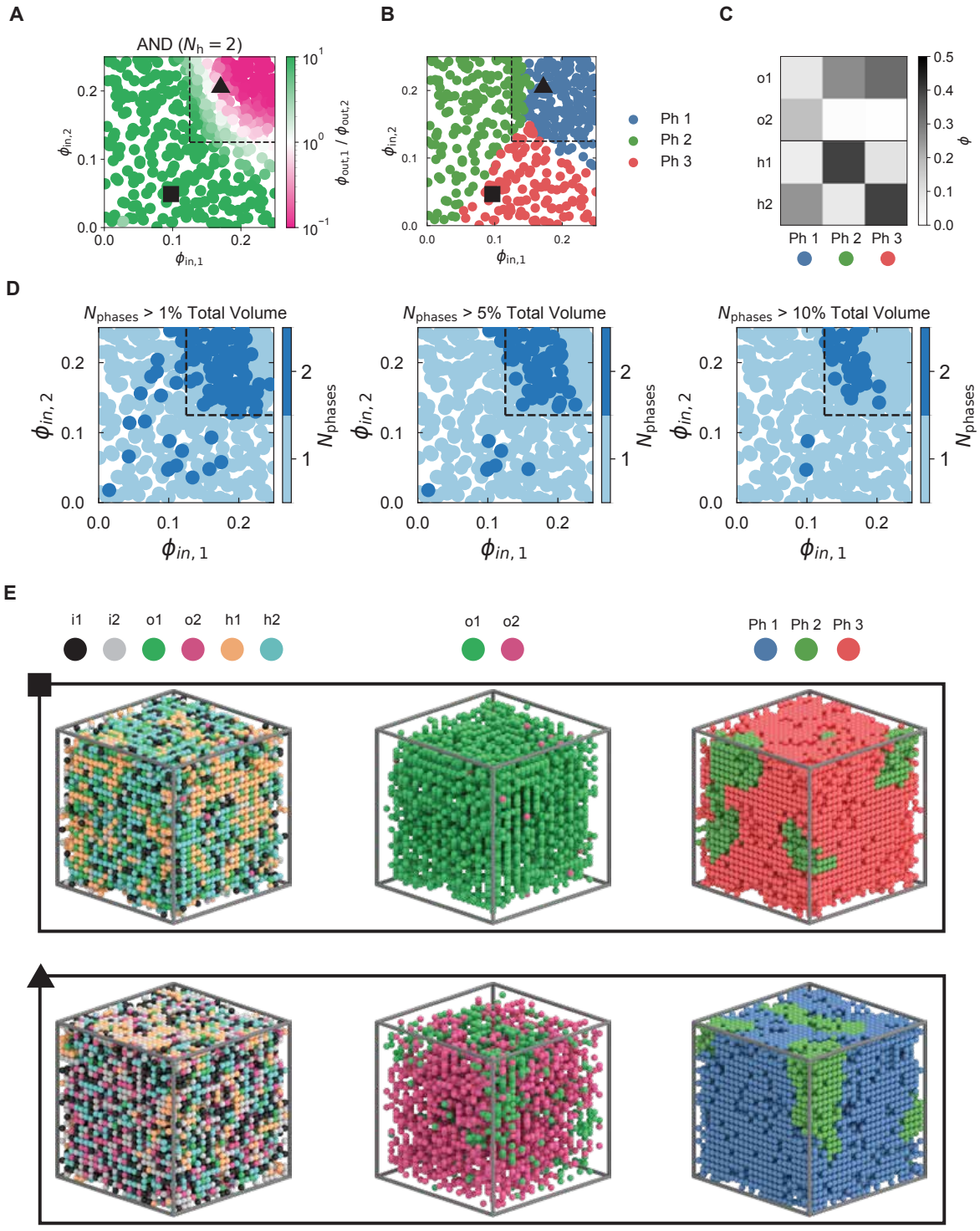

Fig. S14. The structure within the lattice for the molecular network trained to solve the AND boundary. **(A)** The trained decision boundary for Fig. 7, reproduced for clarity. **(B)** The phase decomposition of the input space. Using the methodology presented in SI Note 5, three distinct phases are predicted, consistent with the phase analysis in the mean-field limit (see Fig. S7). **(C)** The concentration vectors defining the 3 discovered phases. **(D)** Near the phase boundaries, the number of phases within the simulation box grows from 1 to 2. **(E)** This change indicates phase separation within the box: at the points indicated in panel (A), which are near phase boundaries, the surface separates into phases 2 and 3 (square) and phases 1 and 2 (triangle).

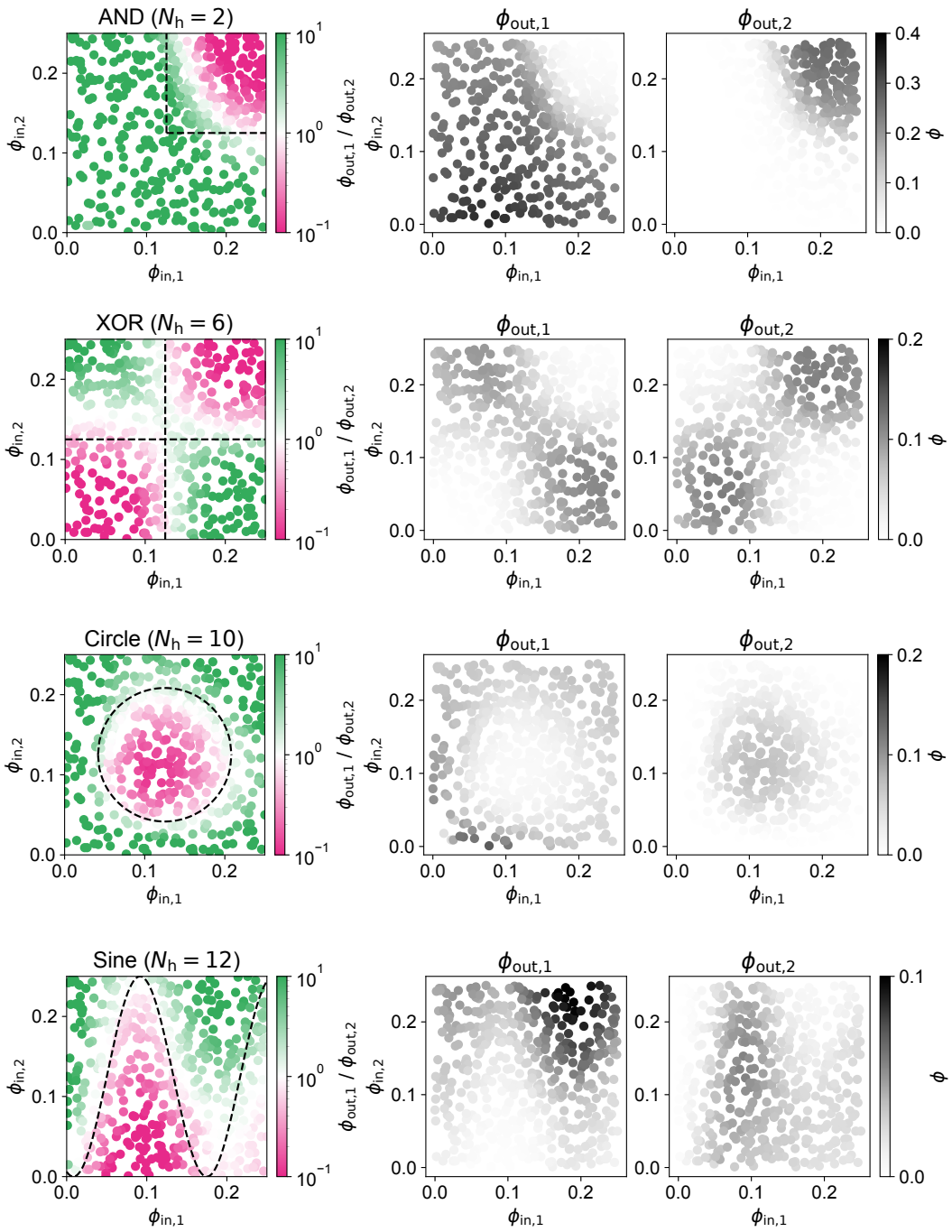

Fig. S15. In the left column are the test predictions from Fig. 7 reproduced with a truncated colorbar ranging from 0.1 to 10 for greater visual clarity. Next to each test prediction are the absolute concentrations of the two output species across the input space.

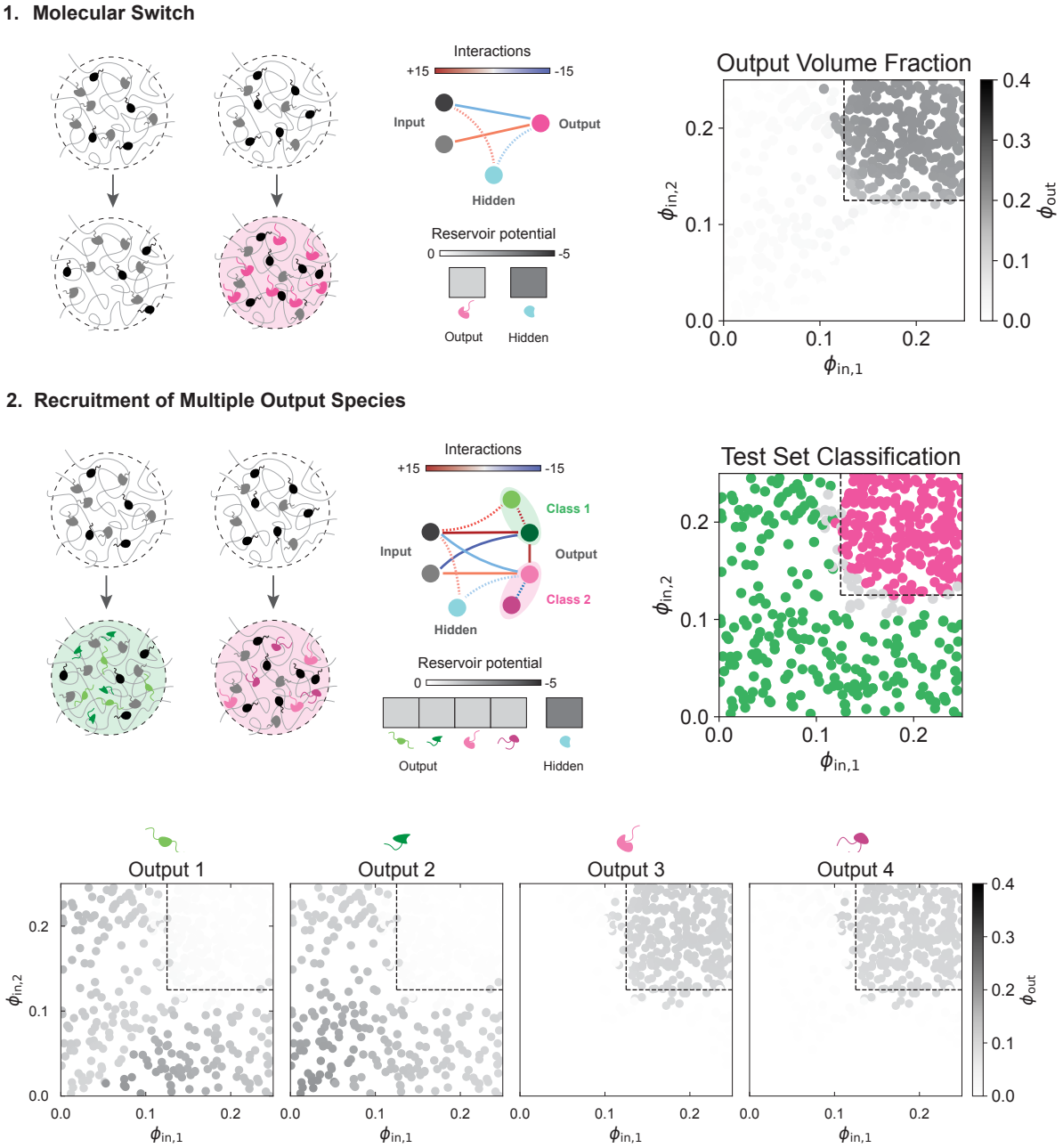

Fig. S16. Examples of additional possibilities for the proposed system. Rather than recruit one output or another, it may be desirable to recruit, for instance, no outputs vs one output (the "molecular switch"—see the top row), or multiple distinct outputs per class (bottom two rows). Many other variations are possible.

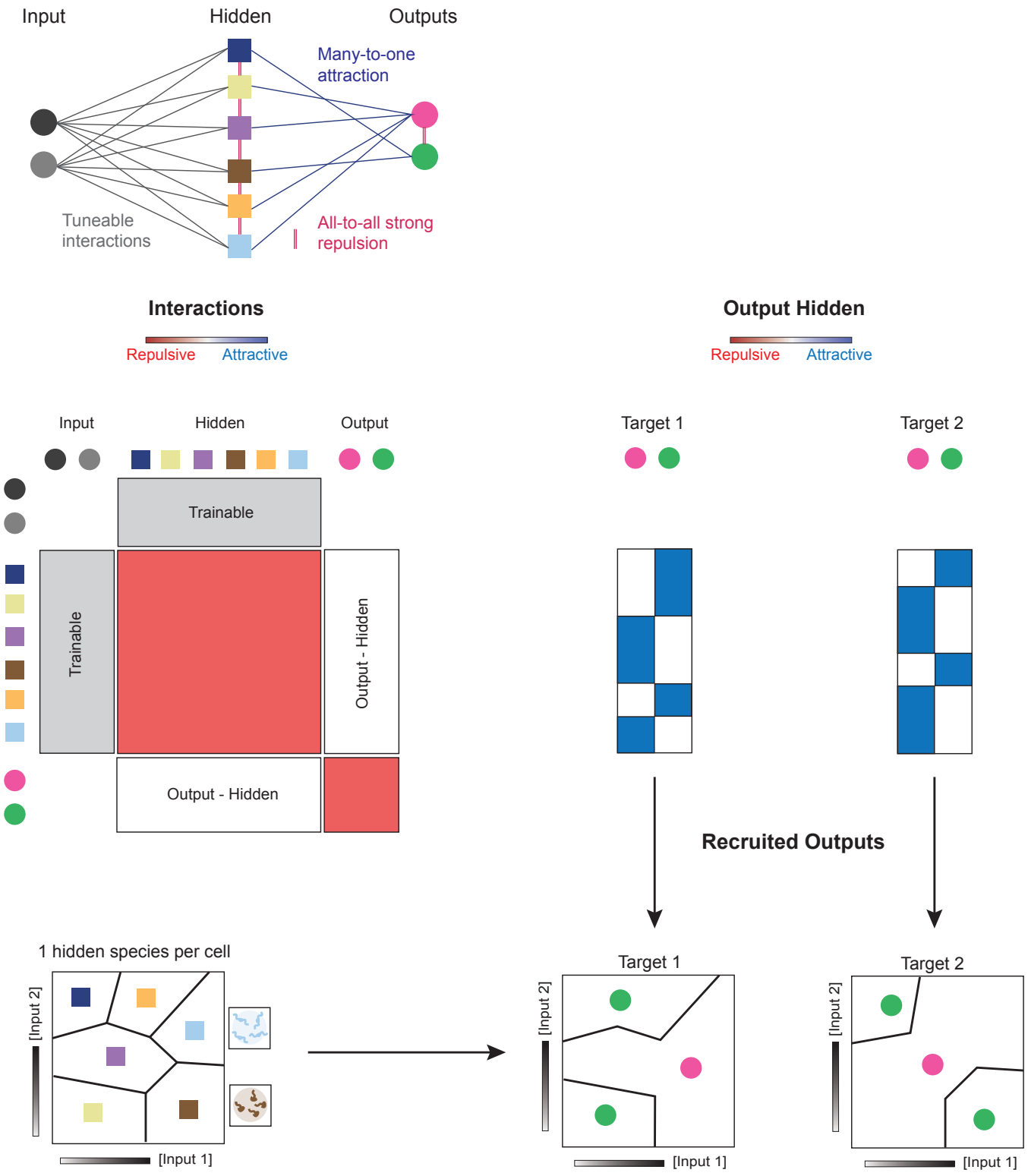

Fig. S17. A schematic illustrating the proposed route towards proving that multiphase systems can perform universal approximation. The input-hidden interactions are tuned while the hidden-hidden and output-output interactions are taken to be repulsive. After training, the input space is partitioned into regions in which condensates are enriched in a single hidden species. Subsequently, the output-hidden interactions can be chosen such that each cell of the input space partition recruits the target output species, in line with the decision boundary being approximated. Increasing the number of hidden species could allow for a finer partitioning of the input space, leading to a better approximation of the target decision boundary.