Supervised task learning via stimulation-induced plasticity in rate-based neural networks

Francesco Borra*, Simona Cocco, Rémi Monasson

*postdoc, Laboratoire de Physique de l'Ecole Normale Supèrieure, CNRS Paris

Neuchip project

Département de Physique
École normale supérieure

The NEU-ChiP project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement Nº:964877

Is it possible to create a chip with biological neurons? → The chip should be able to "do something": a task

Computation with biological neurons

Exploit pre-existing dynamical features of a (possibly structured) network

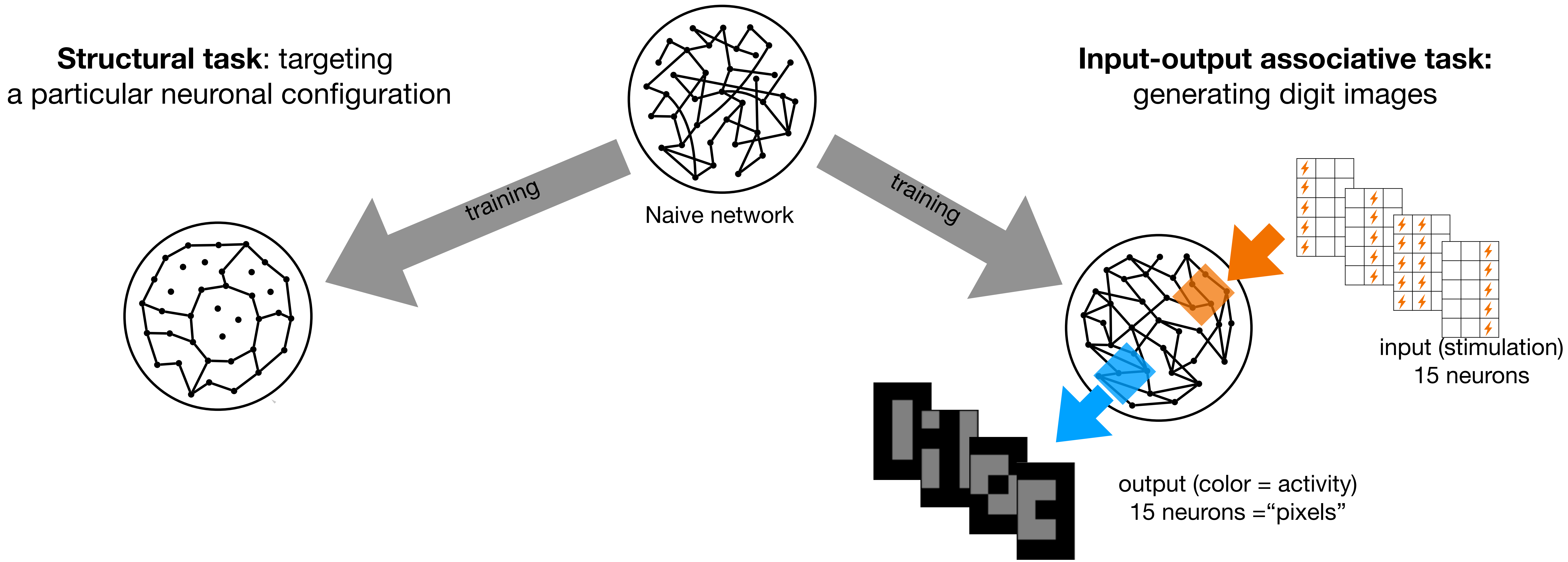
Reservoir computing

Reshaping the network: "Training or encoding" by exploiting **plasticity**

See e.g. KAGAN, Brett J., et al. *Neuron*, 2022, 110.23: 3952-3969. e8.

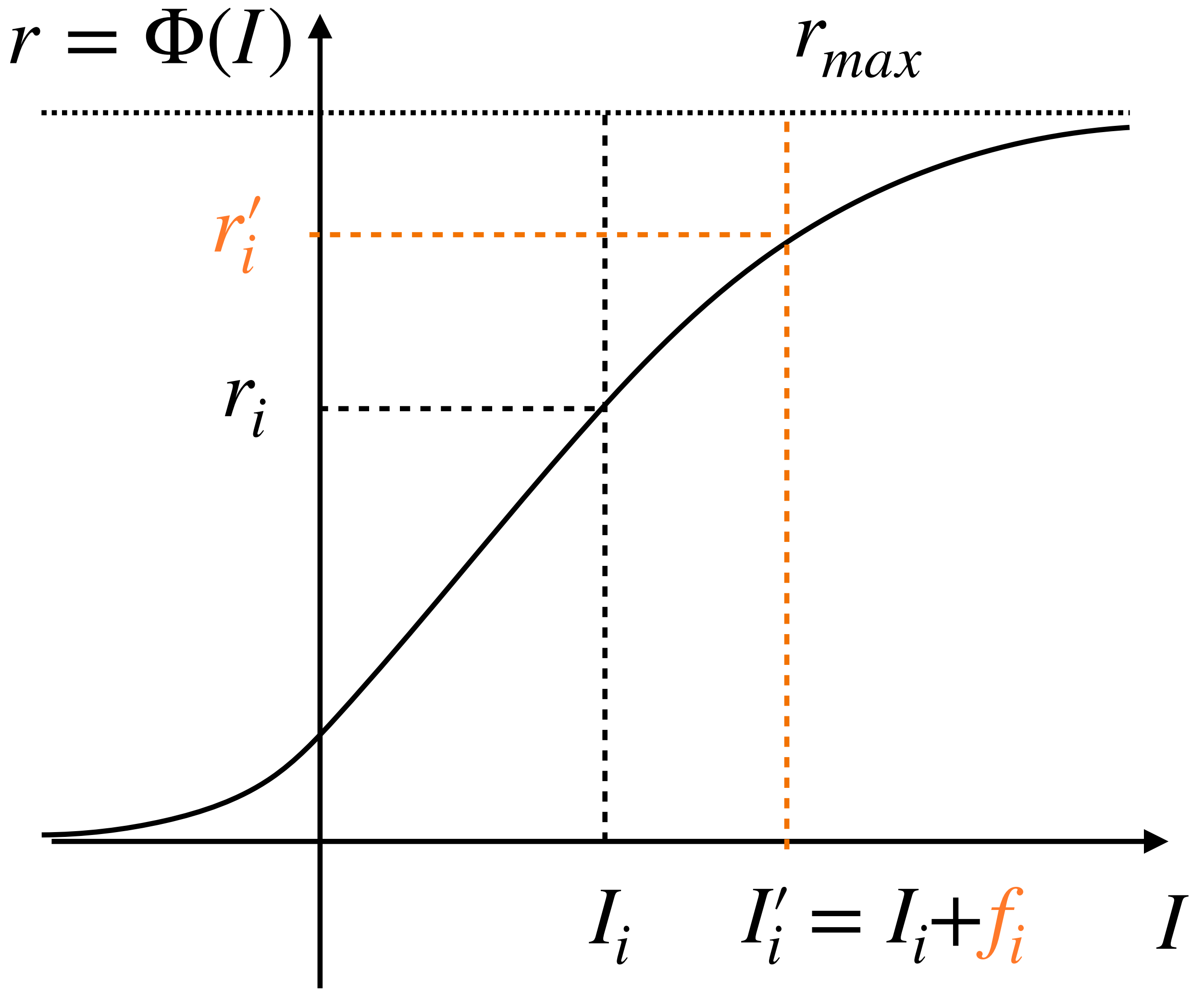
Task-oriented reshaping of a neural network

Two strategies



The model

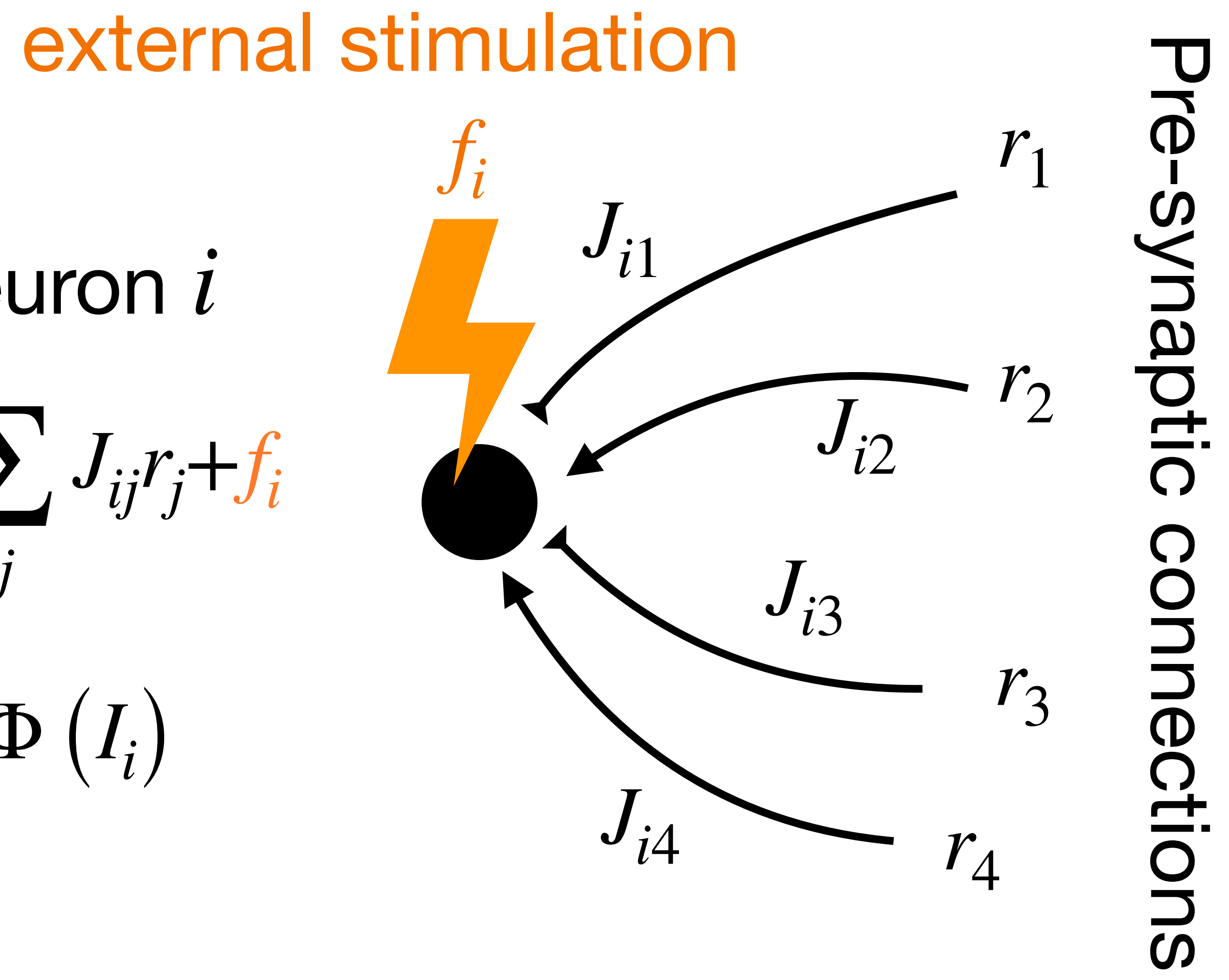
firing rate equation:
relation between firing rates, stimulation, connections



neuron i

average current $I_i = \sum_j J_{ij} r_j + f_i$

firing rate $r_i = \Phi\left(I_i\right)$



By stimulating, we can change the activity of individual neurons.
However, due to connections, effects of stimulation are non-local

Dynamical equation

$$\tau_n \frac{dr_i}{dt}(t) = -r_i(t) + \Phi\left(\sum_j J_{ij}(t)\, r_j(t) + f_i(t)\right)$$

Modelling plasticty

Plasticity equation: how connectins change depending on the activity

$$\tau_s \frac{dJ_{ij}}{dt}(t) = \underbrace{\eta(\epsilon_j)\,(r_i - \theta(\epsilon_j))\,r_j}_{\text{hebbian}} - \underbrace{\beta_1\,J_{ij}\,(r_i^2 - \theta_0(\epsilon_j)^2)}_{\text{homeostasis 1}} - \underbrace{\beta_2\,\text{ReLU}\left(|J_{ij}| - \bar{J}\right)^2}_{\text{homeostasis 2}}$$

homeostasis 1: Activity reverts Towards baseline

homeostasis 2: Synaptic strength Cannot increase Indefinitely

$f =$ control/exernal input

$J =$ connection matrix

$r =$ neuron firing rate

$\epsilon_i = E/I$

$\Phi =$ activation function: soft ReLU with maximum rate

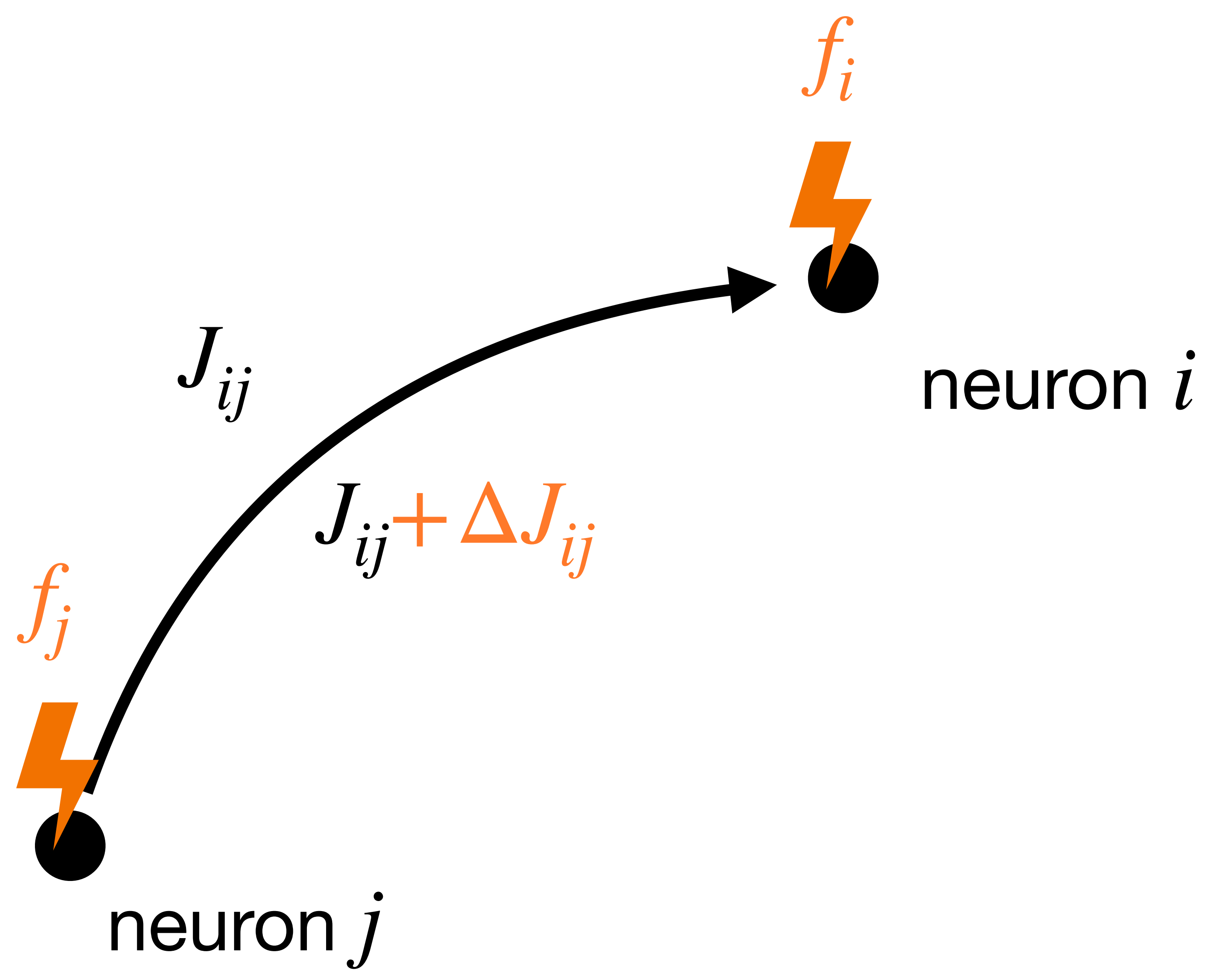
Timescale separation assumption $\tau_s \gg \tau_n$

Neural dynamics is faster than plasticity

$$\tau_n \frac{dr_i}{dt}(t) = -r_i(t) + \Phi\left(\sum_j J_{ij}(t)\,r_j(t) + f_i(t)\right) \longrightarrow r_i = \Phi\left(\sum_j J_{ij}\,r_j + f_i\right)$$

Modelling plasticty

$$\tau_s \frac{dJ_{ij}}{dt}(t) = \underbrace{\eta(\epsilon_j)\,(r_i - \theta(\epsilon_j))\,r_j}_{\text{hebbian}} - \underbrace{\beta_1\,J_{ij}\,(r_i^2 - \theta_0(\epsilon_j)^2)}_{\text{homeostasis 1}} - \underbrace{\beta_2\,\text{ReLU}\big(|J_{ij}| - \bar{J}\big)^2}_{\text{homeostasis 2}}$$



By controlling the activities r_i and r_j via stimulation,
we can in principle control plasticity.

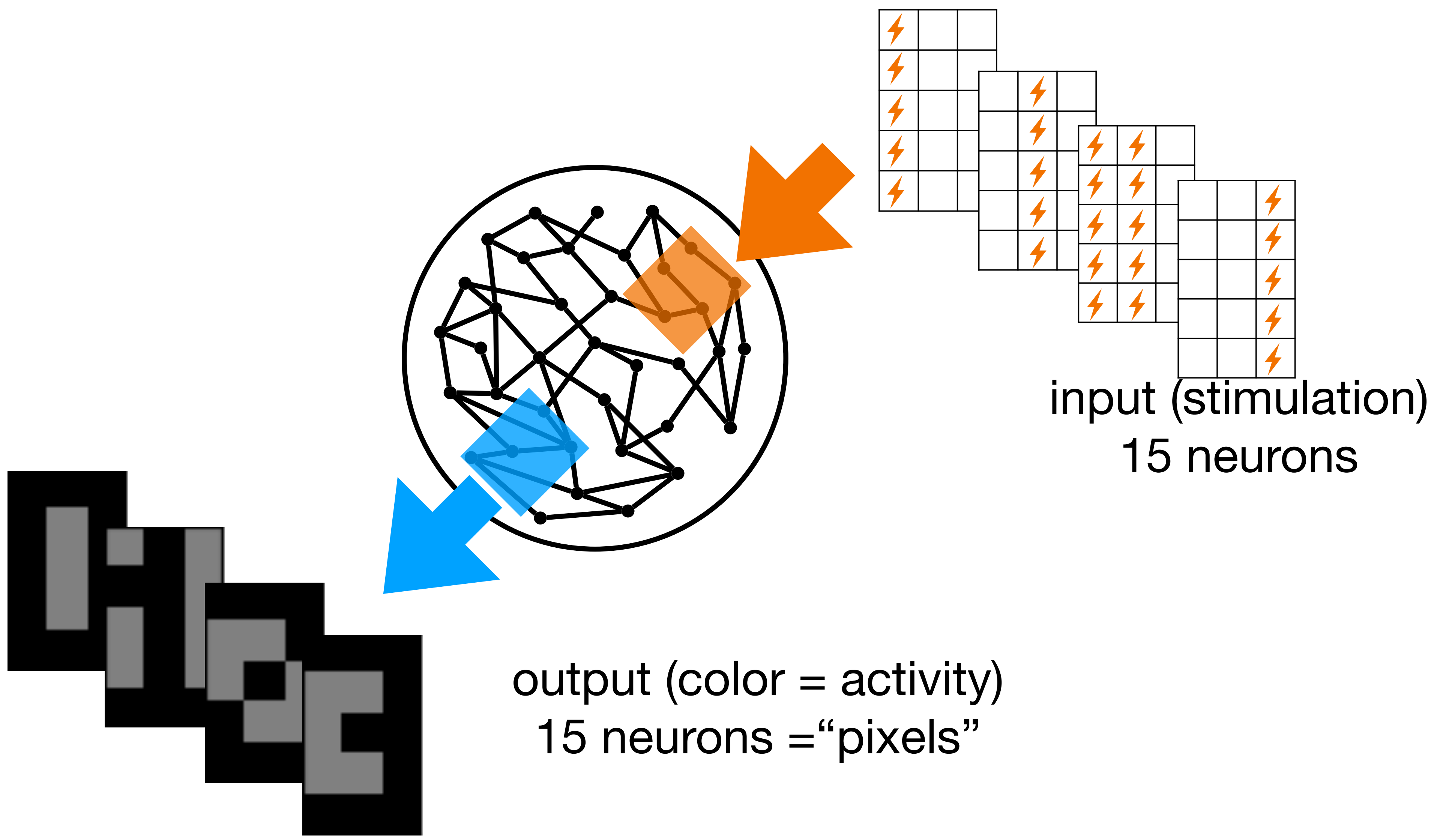
BUT

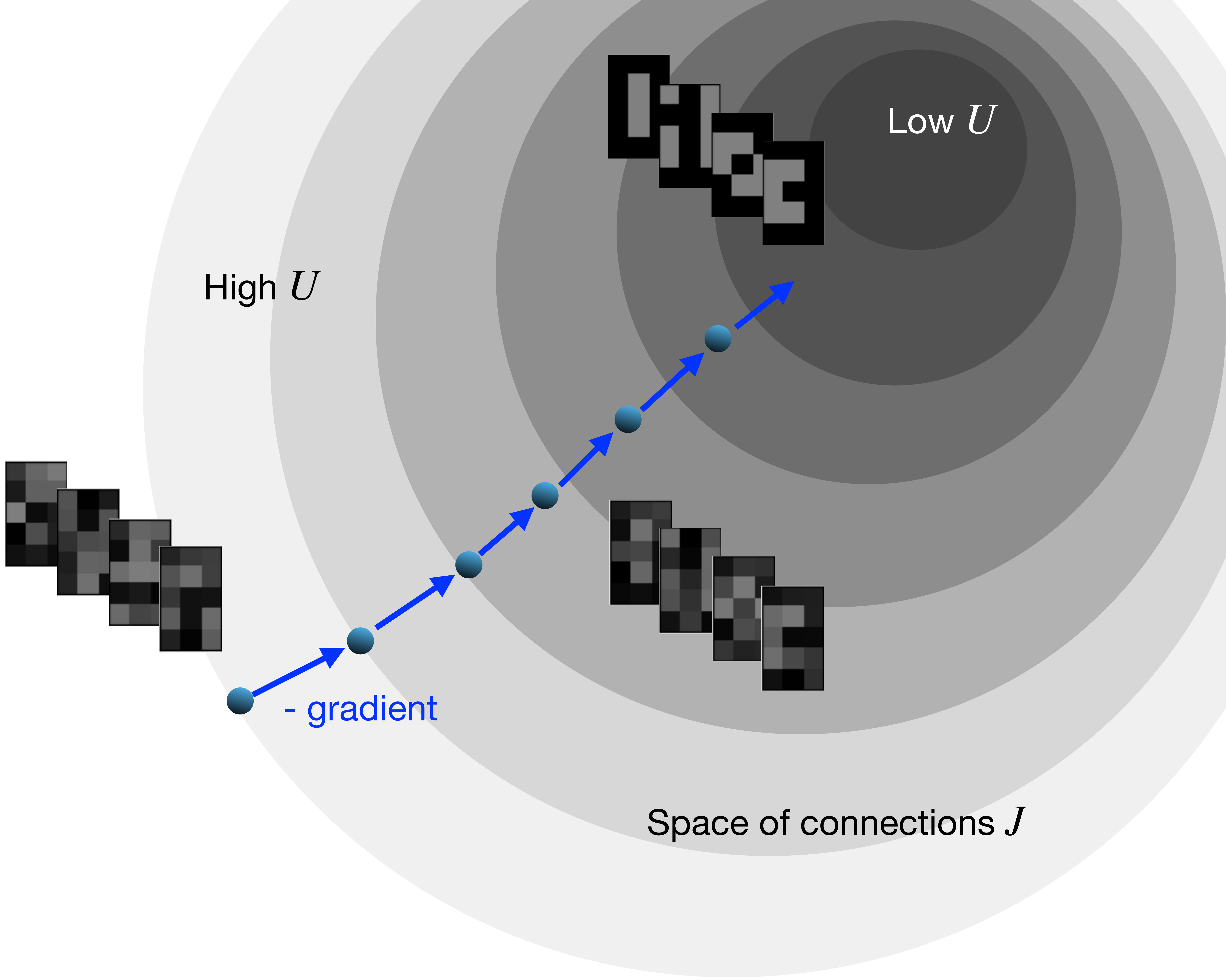
1) We do not have necessariliy full control of the network: connected activity $r_i = \Phi\left(\sum_j J_{ij}\,r_j + f_i\right)$

2) Even so, by changing activity of neuron i, in principle we affect all connections to and from neuron i:
If we have N neurons, we have $\approx N^2$ connection and we can only control N neurons: hard control problem

No direct control: implications

Cost function $U(\mathbf{J})$ = how well the connectivity performs a task: e.g. average square error





Local best synaptic modification: (minus) the gradient, i.e. direction along which the cost decreases the most

$$\Delta \mathbf{J} \approx -\eta \, \nabla U(\mathbf{J})$$

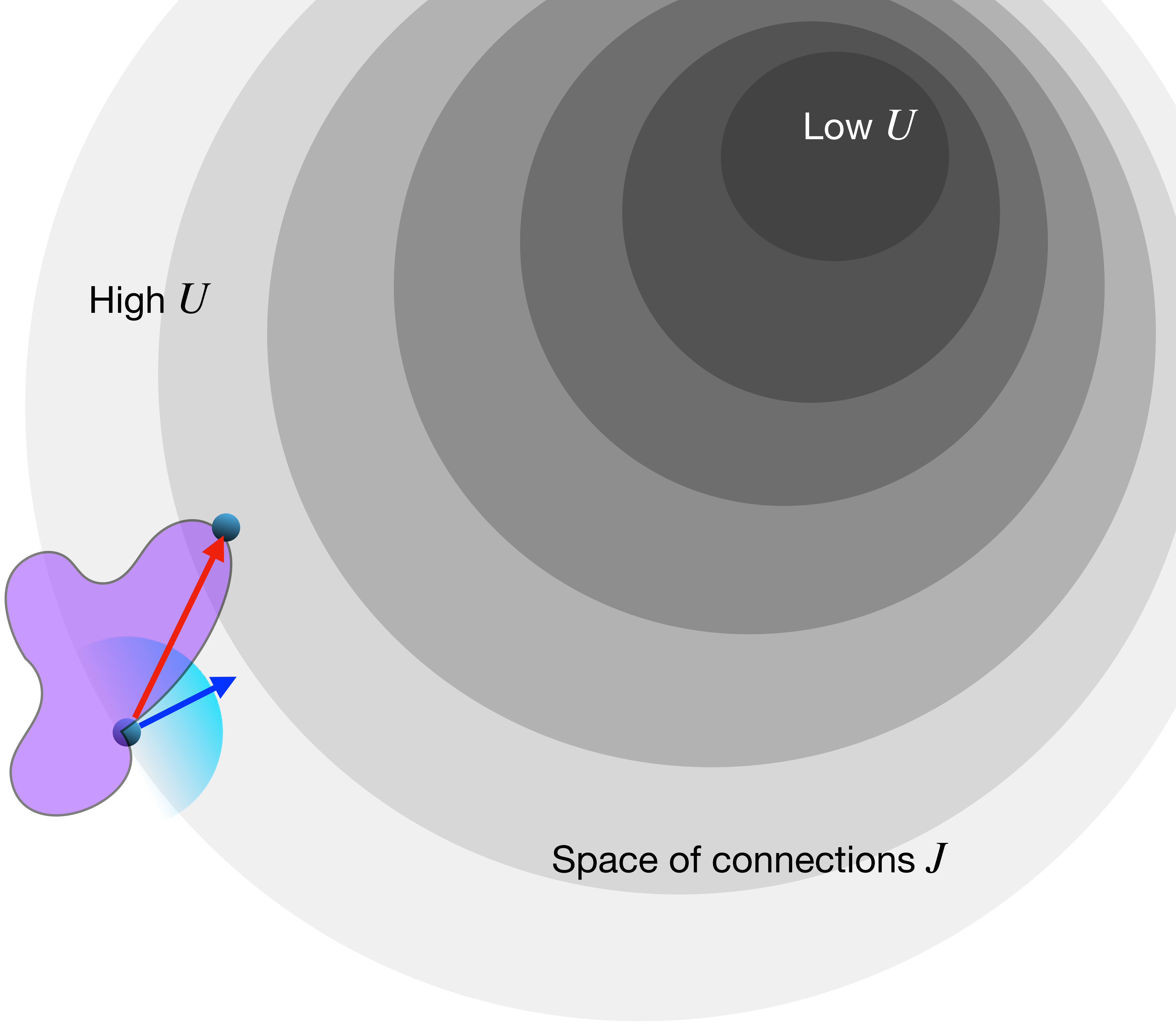
We are not free to implement this!

Not all directions in the space of connections
are allowed by the dynamics of the synapses

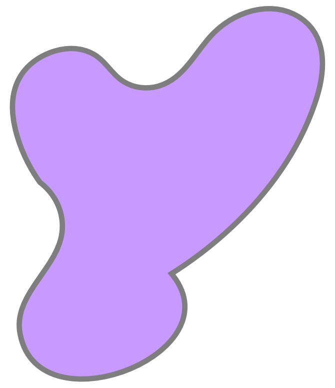
1) <u>**$\approx N^2$ connections, but $\approx N$ controllable units!**</u>

2) No vanishing leaerning rates: no infinitesimal updates

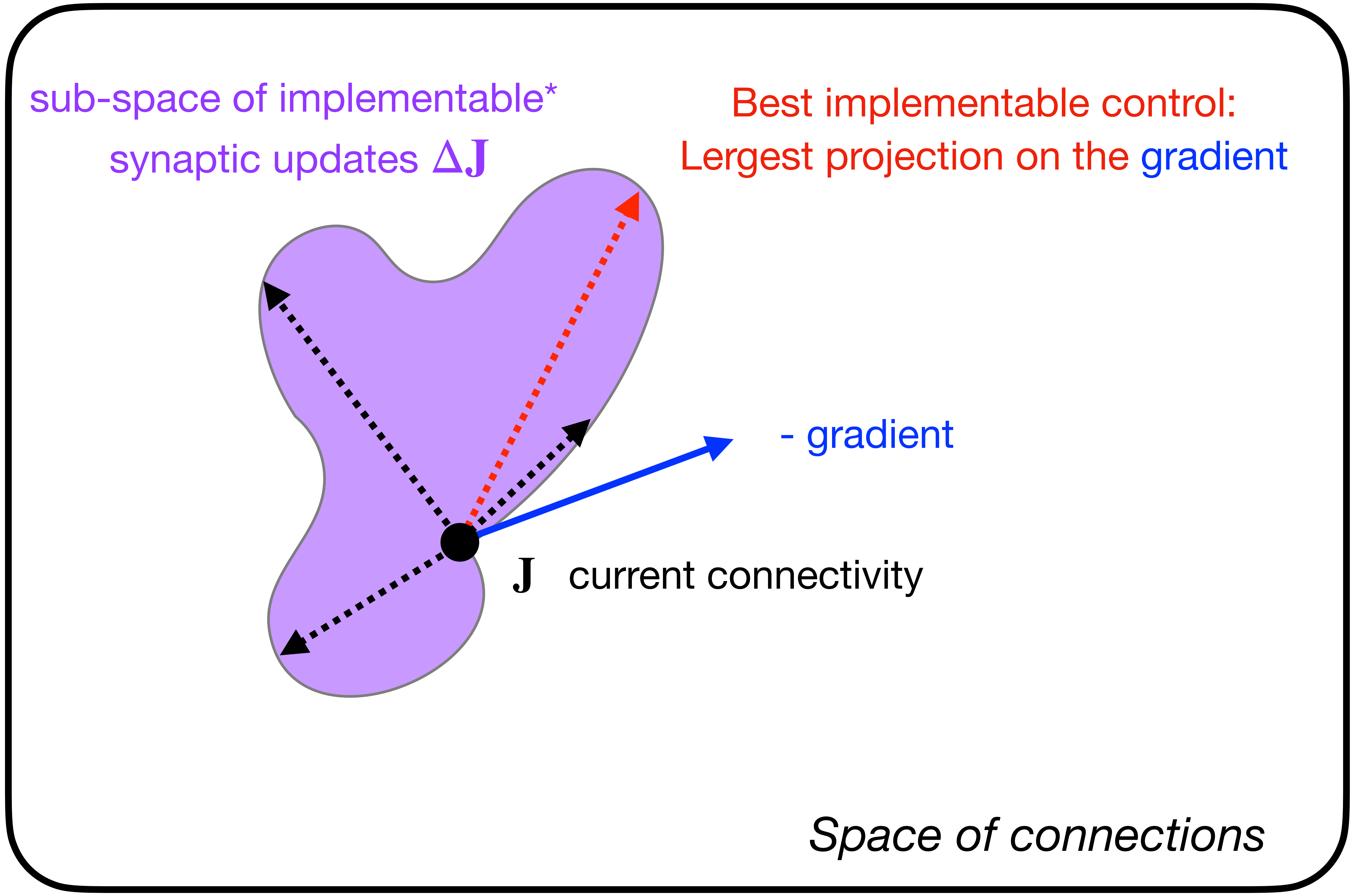
How do we find the control to
implement the best possible direction?



Favorable directions
(U decreases)

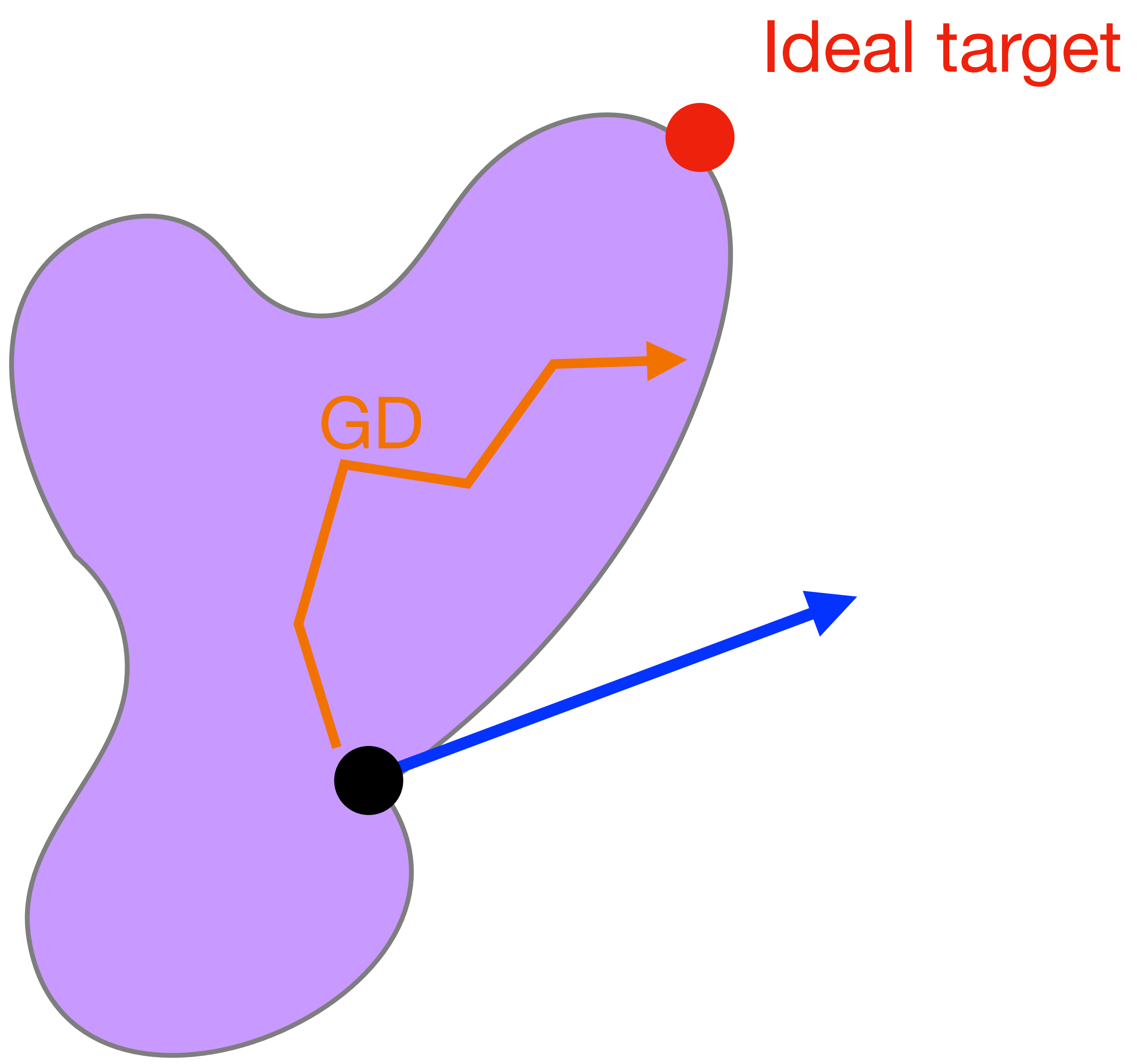


Variations allowed by
plasticity constraints



How do we find the best control?
A gradient descent in the space of controls

$$\mathbf{f} \to \mathbf{f} - \eta \nabla_{\mathbf{f}}(\Delta U)$$



A control ΔJ_{ij} ($N \times N$ matrix) is implementable if there is
a control f_i (N-dimensional array) which induces it

Inferring the structure

Inferring connectivity with a model:
Many possibilities*: here we use an idealized but consistent procedure

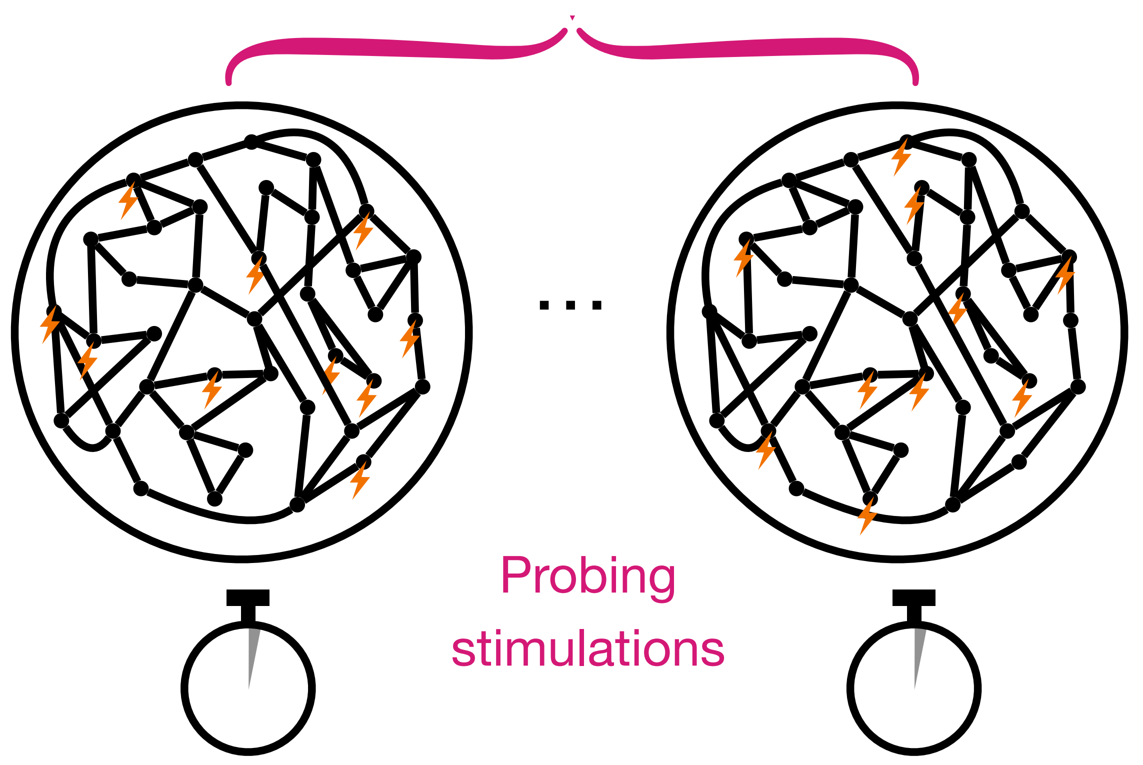
$$\Phi^{-1}(r) = Jr + f$$

N (=number of neuron) equations

With n different stimulations f_μ we have Nm equations

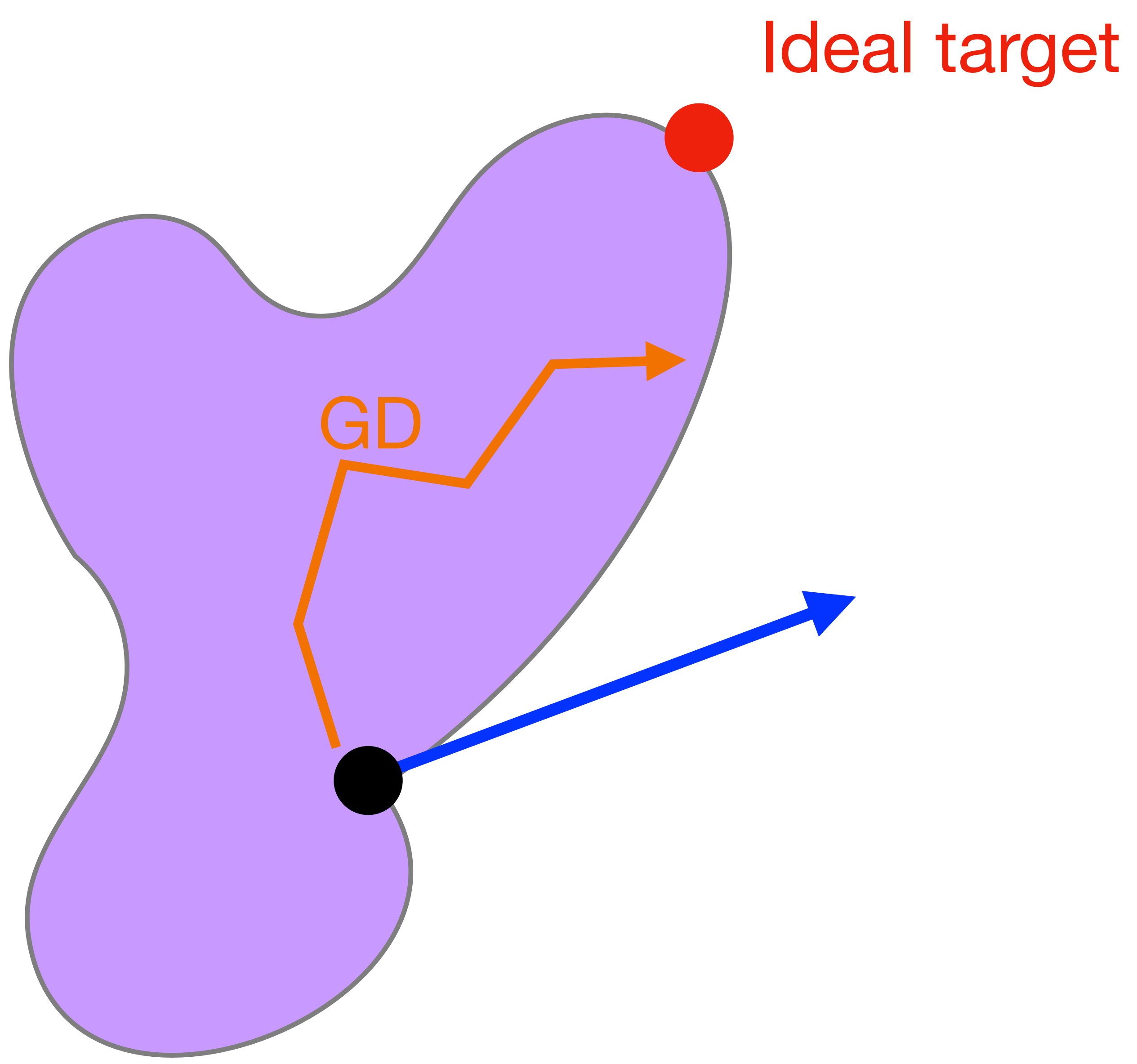
$$\Phi^{-1}(r_\mu) = Jr_\mu + f_\mu$$

With n different stimulations f_μ we have Nm equations. If $m > c$ (connectivity), we can infer J

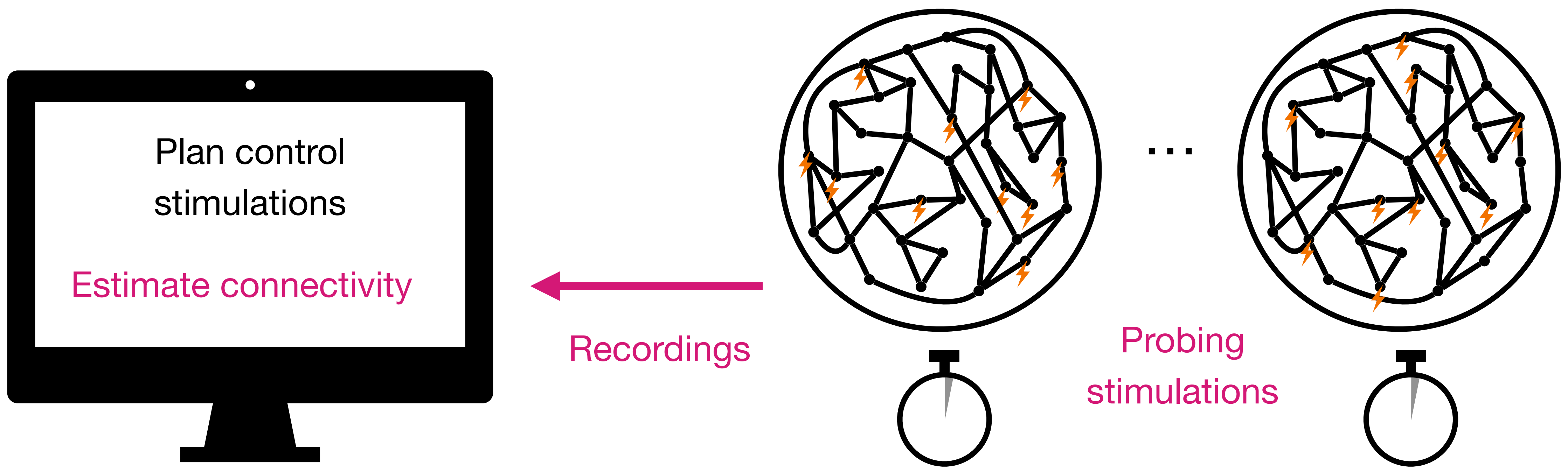


Computing optimal (or good) stimulation

$$\mathbf{f} \rightarrow \mathbf{f} - \eta \nabla_{\mathbf{f}}(\Delta U)$$



We compute an **array** $\mathbf{f}$ which describes the stimulation we should apply in each site

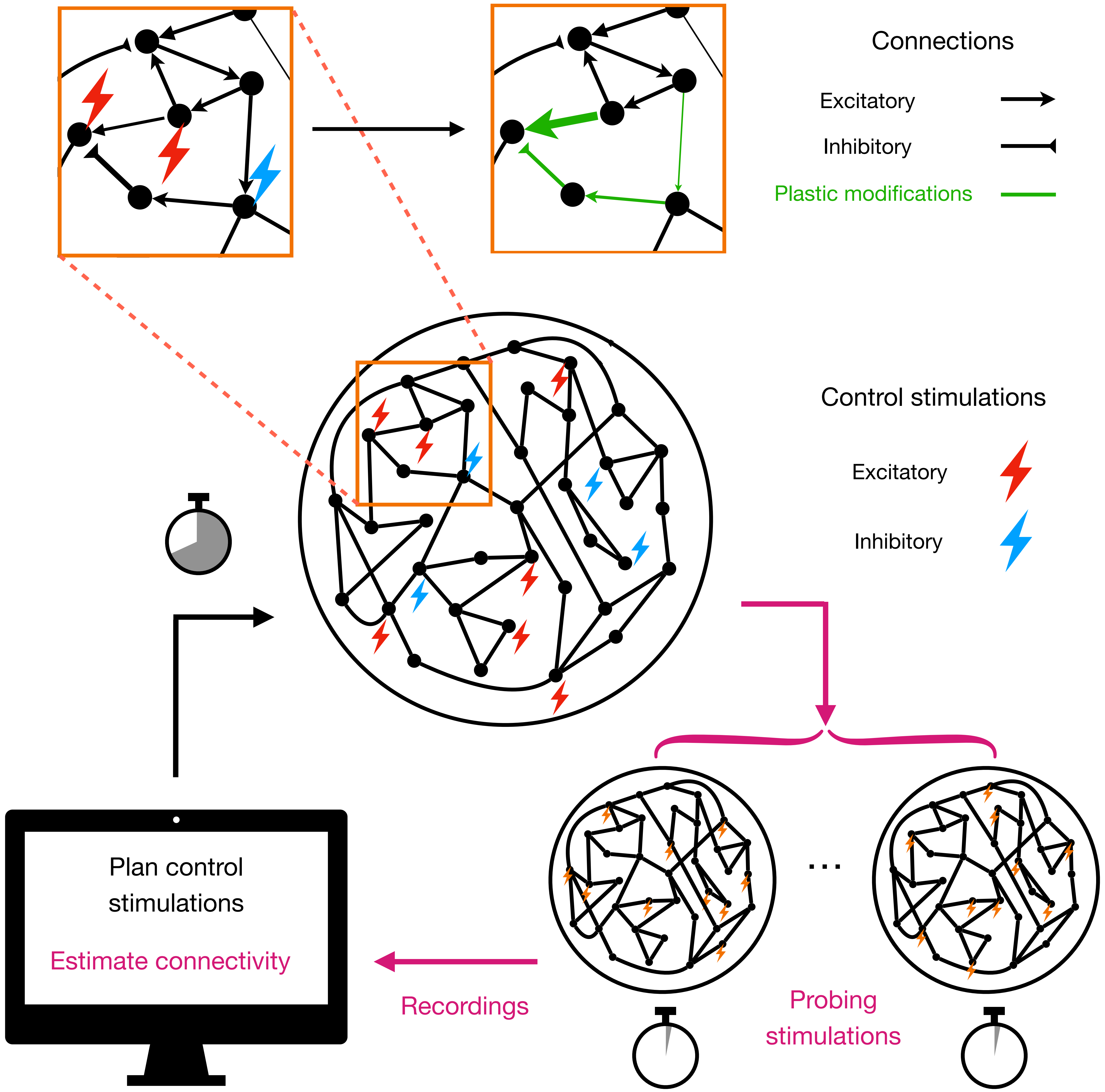


LOOP

Inferring

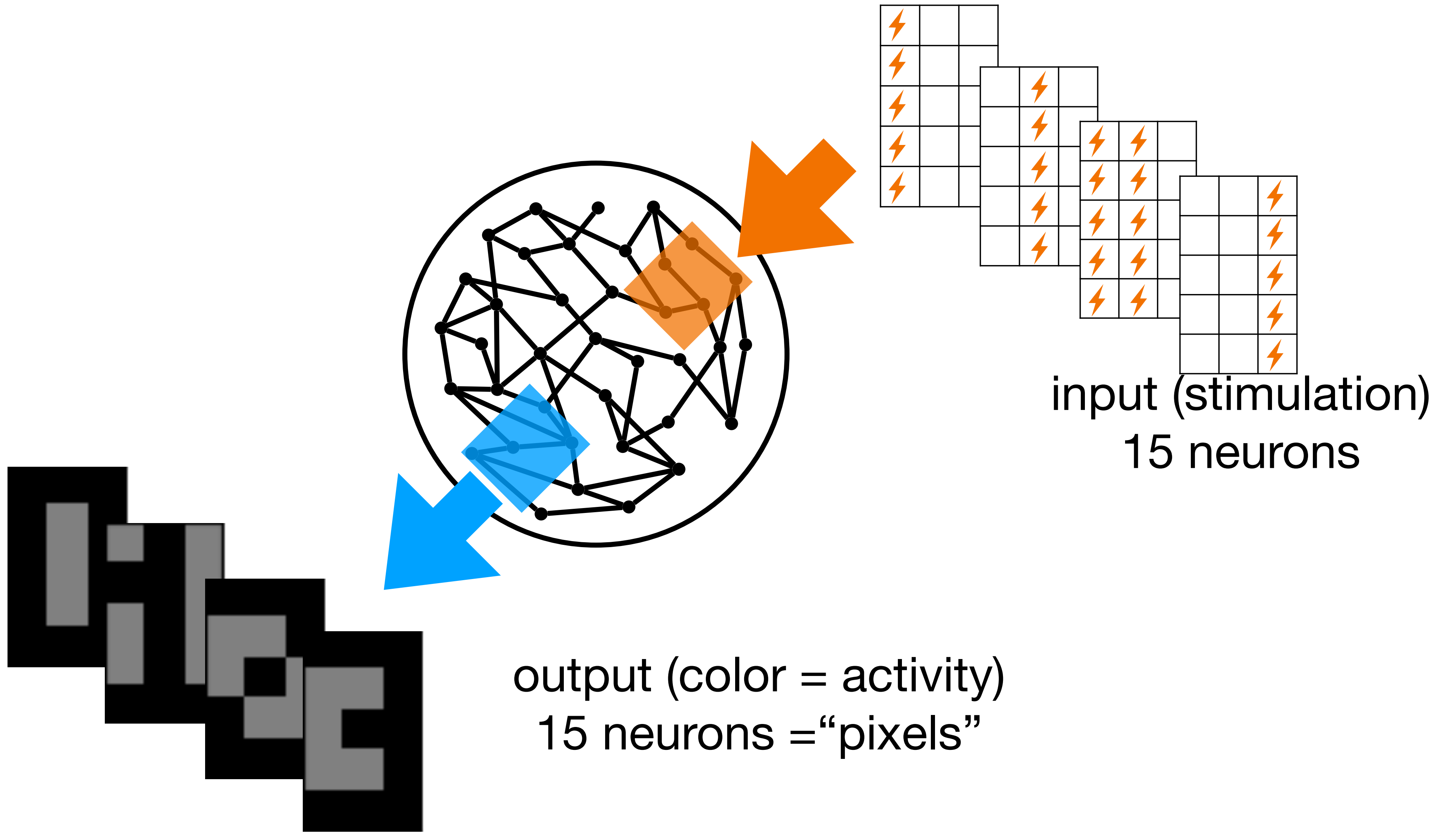
Computing

Stimulating = “training”



Input-output associative task:

generating digit images



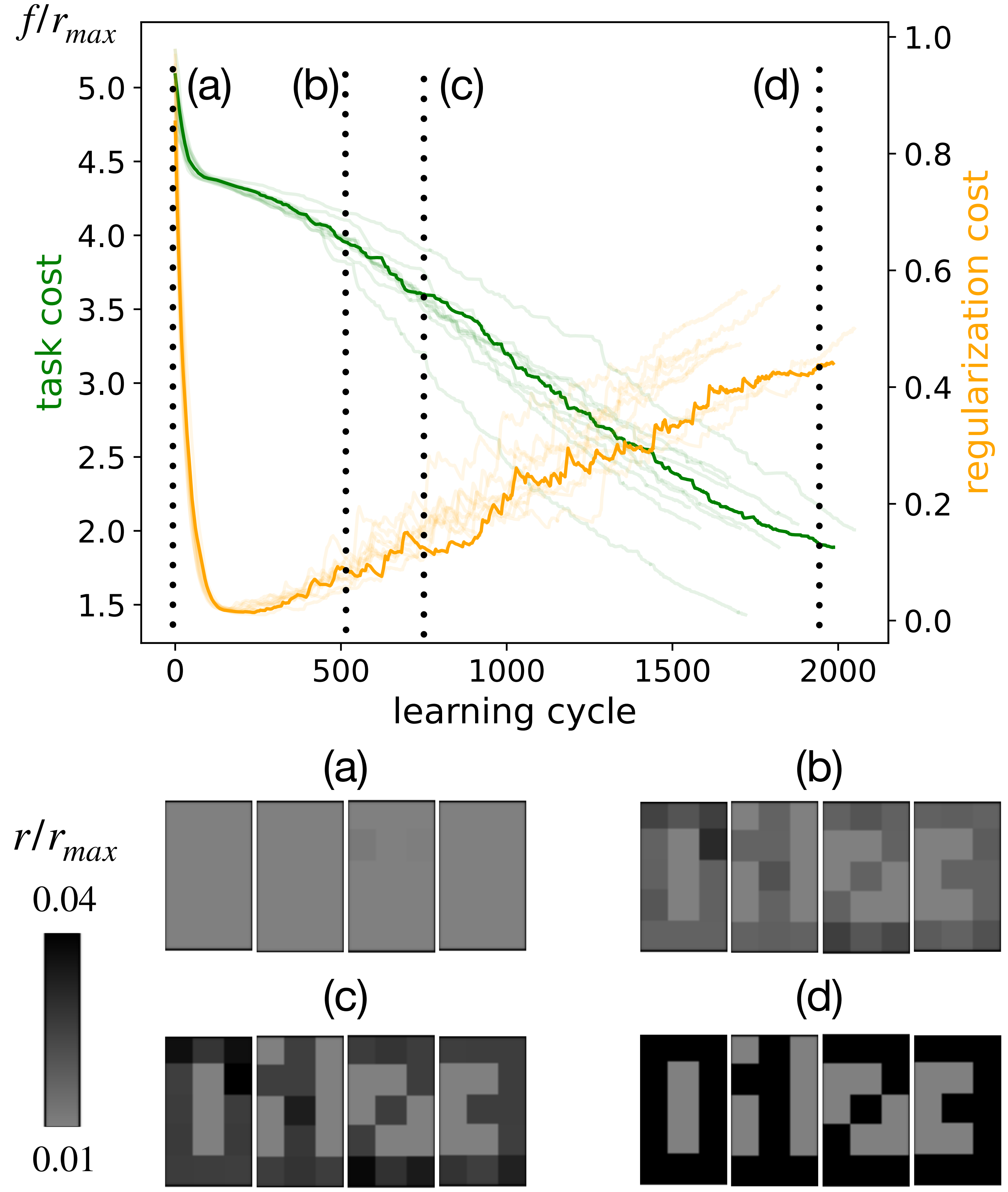
Cost function $U = U_{task} + U_{reg}$

Cost function: (smoothed) softmax quadratic error of the most ambiguous pair of pixels:

$$U_{task}(\mathbf{J}) \propto \sum_{(i\mu)(j\nu)|\sigma_{i\mu}=0,\sigma_{j\nu}=0} \Delta(i\mu, j\nu)\ \exp(\gamma\Delta(i\mu, j\nu))$$

$$\Delta(i\mu, j\nu) = (r_i(\mathbf{f}_\mu) - r_j(\mathbf{f}_\nu) - \delta r)^2\, \Theta(r_i(\mathbf{f}_\mu) - r_j(\mathbf{f}_\nu) - \delta r)$$

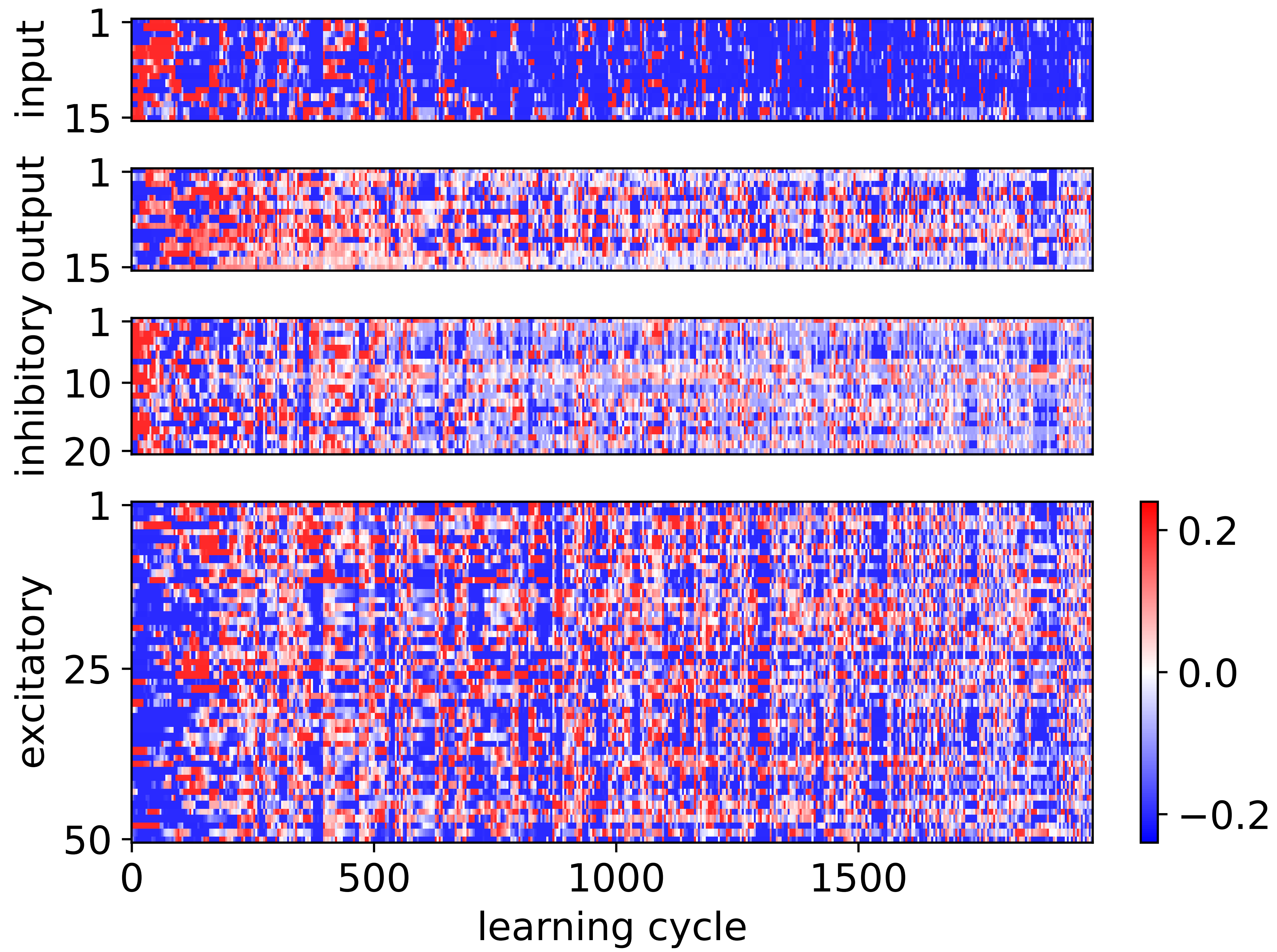
where Θ is Heaviside theta function and $\delta r = .14\, r_{max}$.



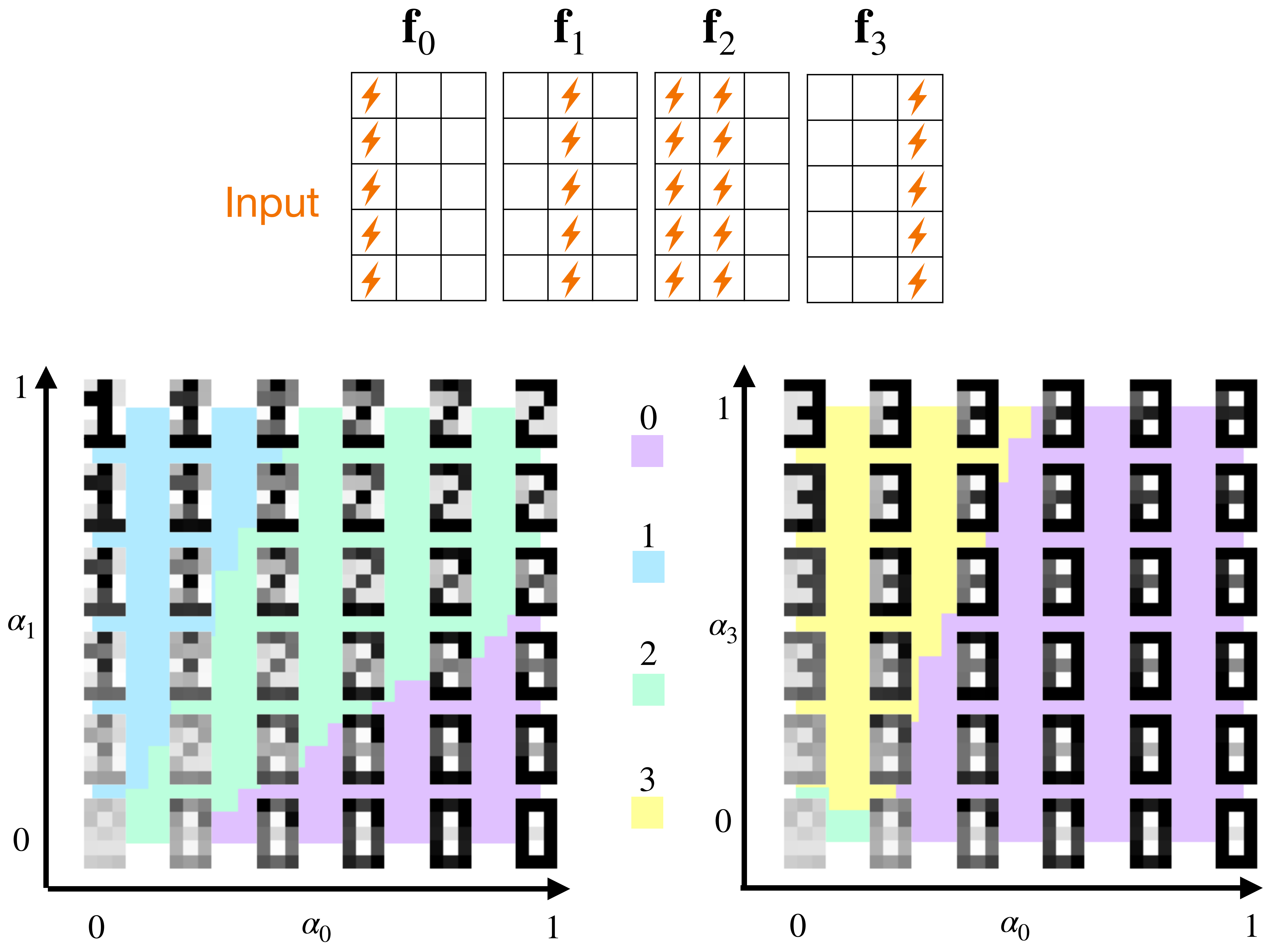
Input-output associative task:

generating digit images

The protocol



A non-linear task



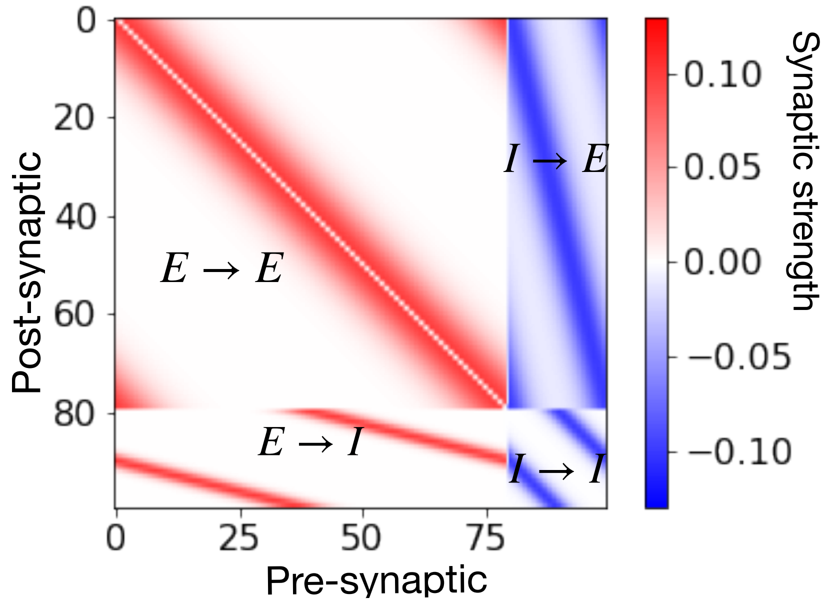
$\mathbf{f} = \alpha_0 \mathbf{f}_0 + \alpha_1 \mathbf{f}_2$

$\mathbf{f} = \alpha_0 \mathbf{f}_0 + \alpha_3 \mathbf{f}_3$

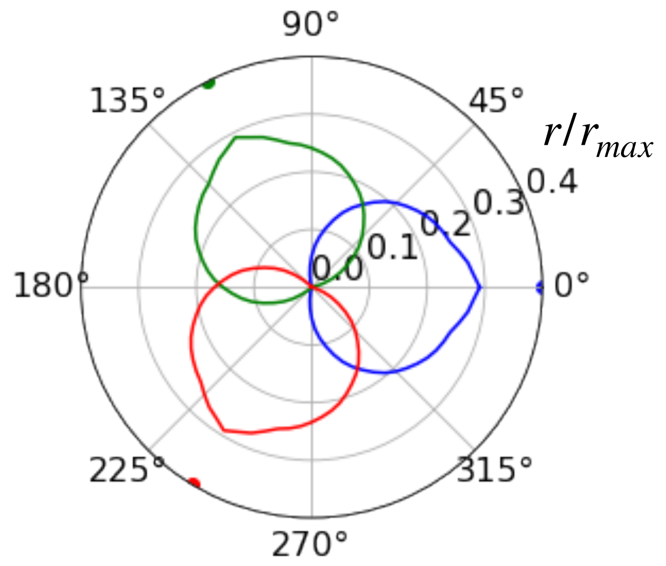
Structural task: creating a specific connectivity structure

Target connectivity

Specifically, we try to build a continuous attractor



connection map $\mathbf{J}^*$



receptive fields

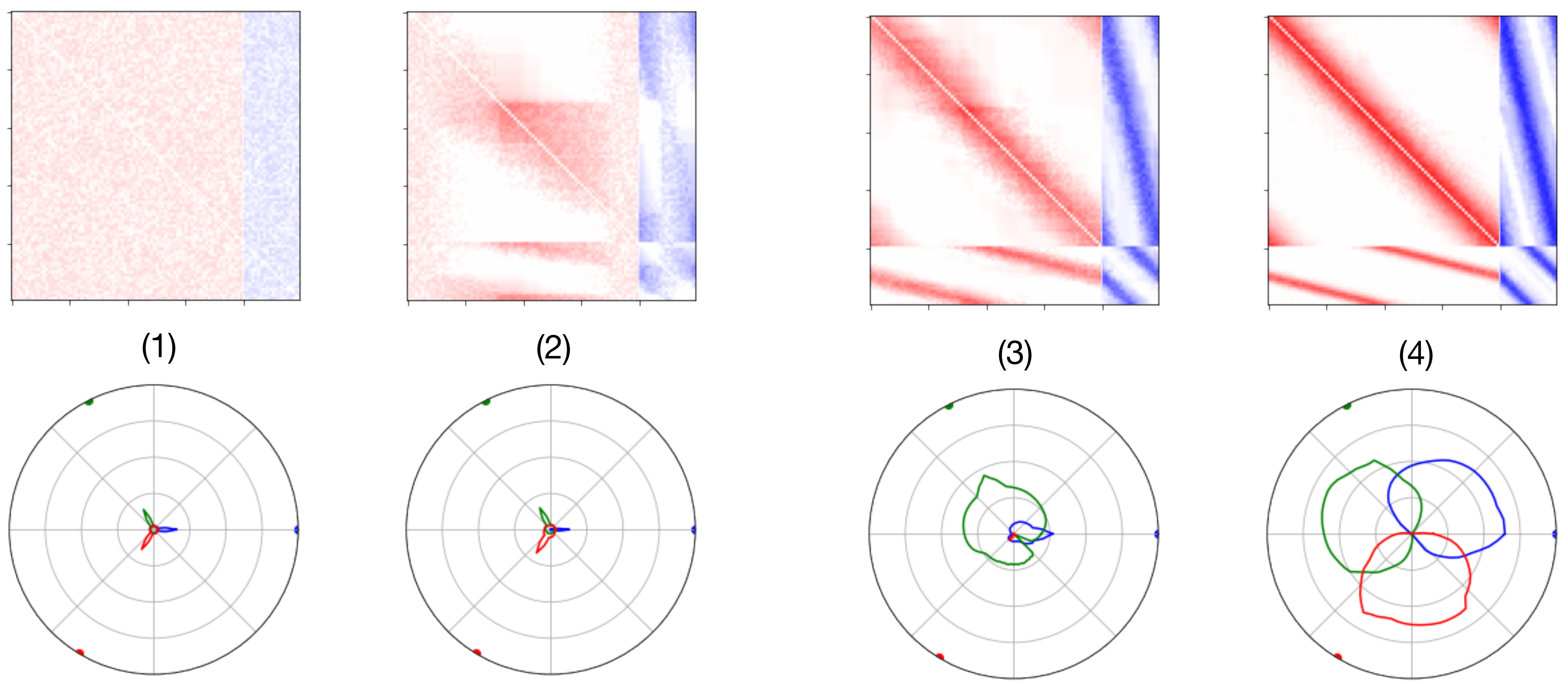
Cost function

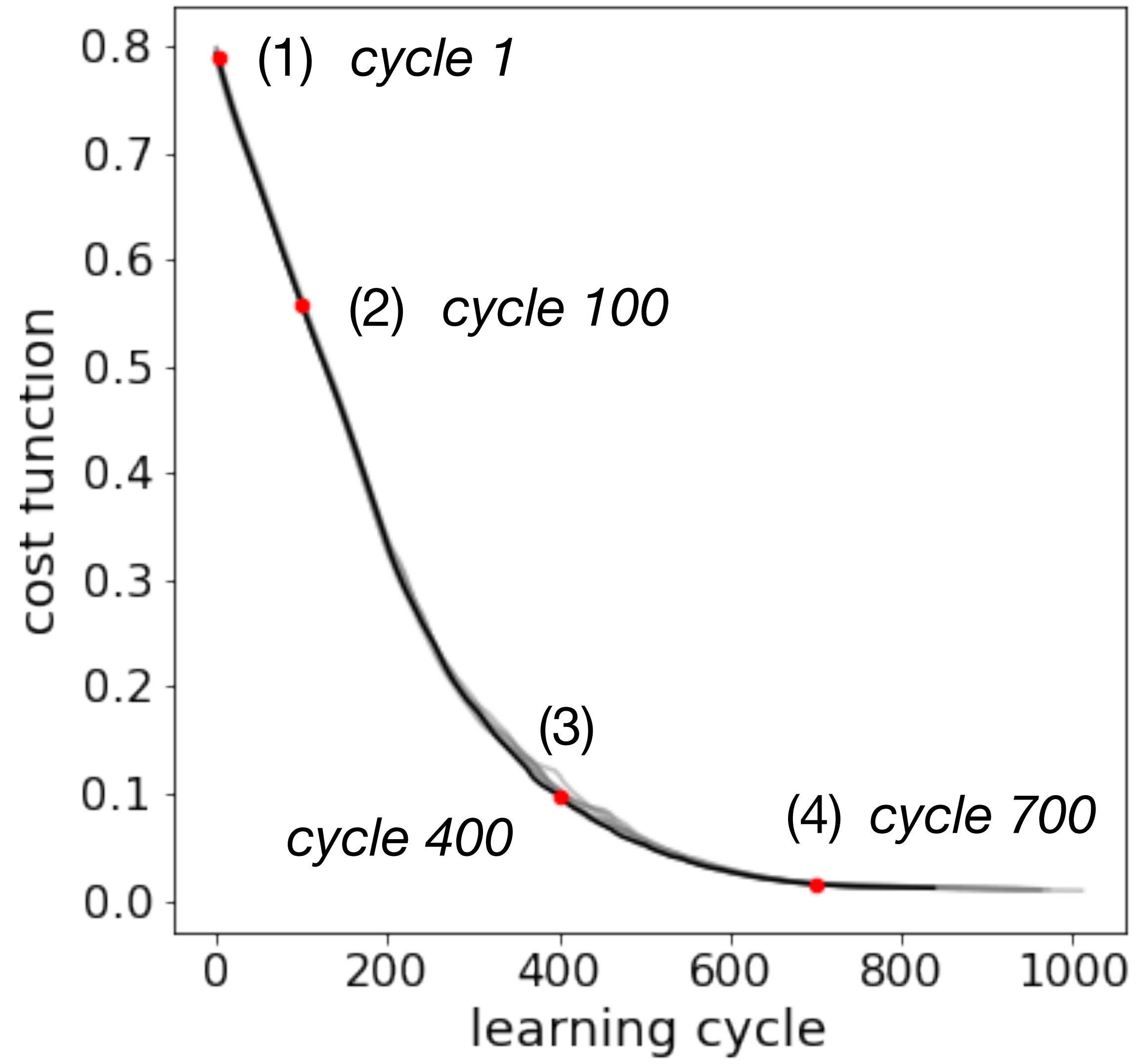
$$U = \sum_{ij} w_{ij}(J_{ij} - J^*_{ij})^2$$

J^*_{ij} = target connectivity

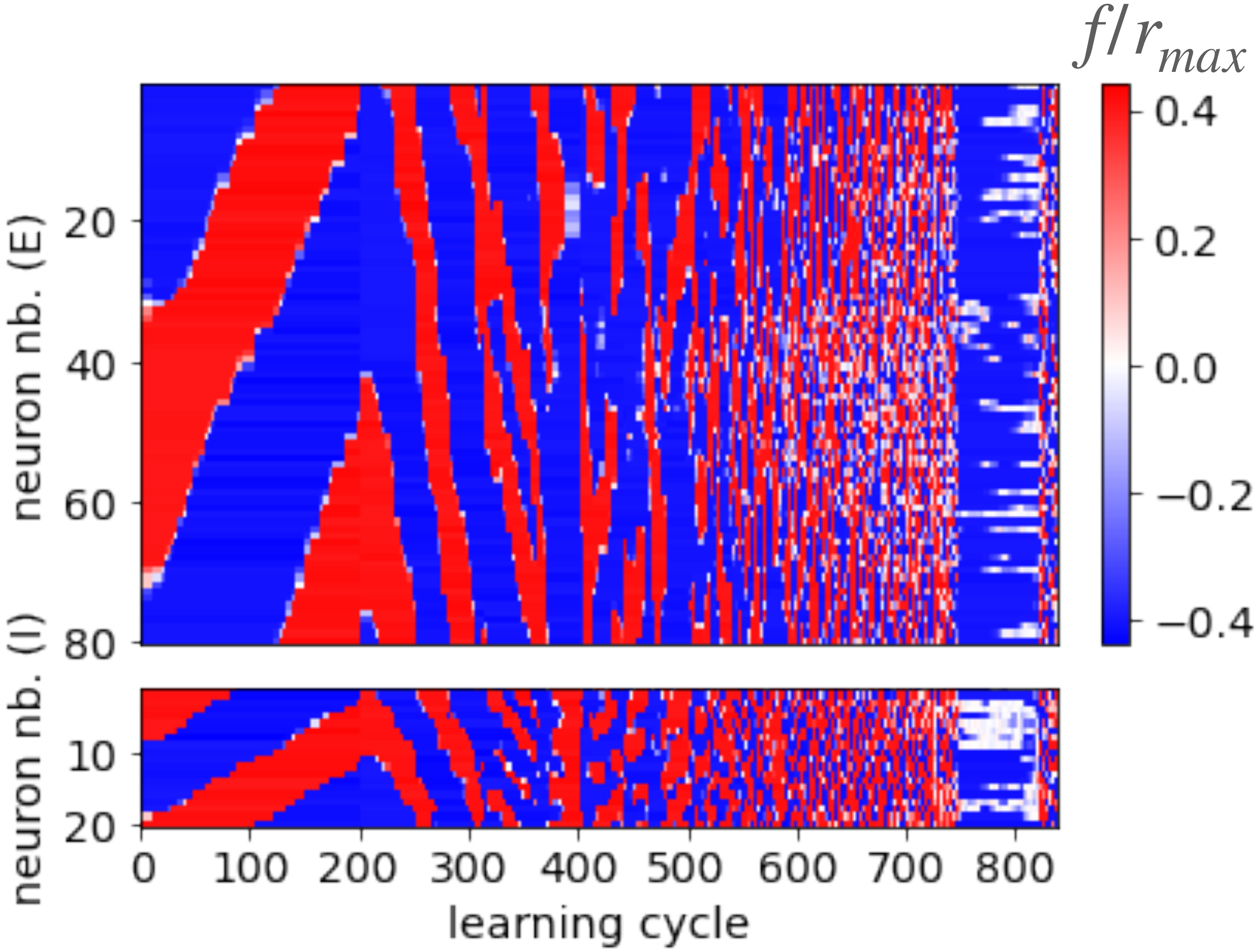
w_{ij} = balancing weights

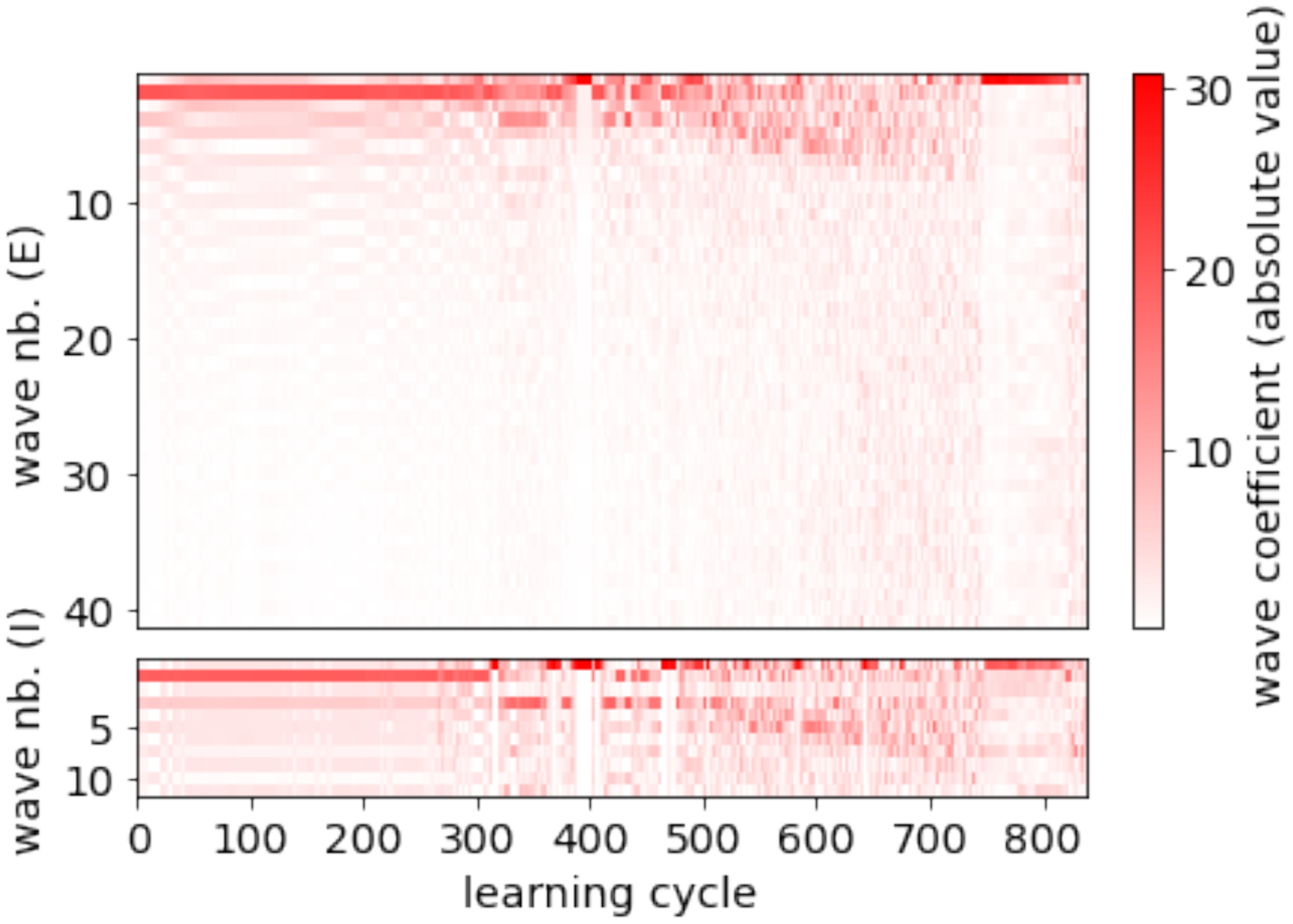
Training





Structural task: an interpretable protocol





Technique features

1) General and flexible: different learning/plasticity rules, activation functions, tasks can be implemented. We tried Hebbian, anti-Hebbian rules and different parameters settings

2) Some robustness with respect to parameter error (though this would require a more complete investigation)

3) While our implementation assumes neuron wise control, there is no algorithmic difference between working with individual neurons and groups of neurons

Delicate points

1) Strog noise and uncertainty might require some modifications

2) Certain steps of the algorithm are sensitive to implementation

3) Very large network might be difficult to handle

4) A good knowledge of the system properties is required

5) <u>Is control always possible? Let's see…</u>

Thank you for your attention

Borra, Francesco, Simona Cocco, and Rémi Monasson.
“Supervised task learning via stimulation-induced plasticity in rate-based neural networks." (2023).

NEU-Chip project

The NEU-ChiP project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement Nº:964877