Embeddings of neural field dynamics : simulations and fractal mappings

Lab meeting 2023-04-20

Nicole Vadot

quick recap : neural field equations

connectivity kernel

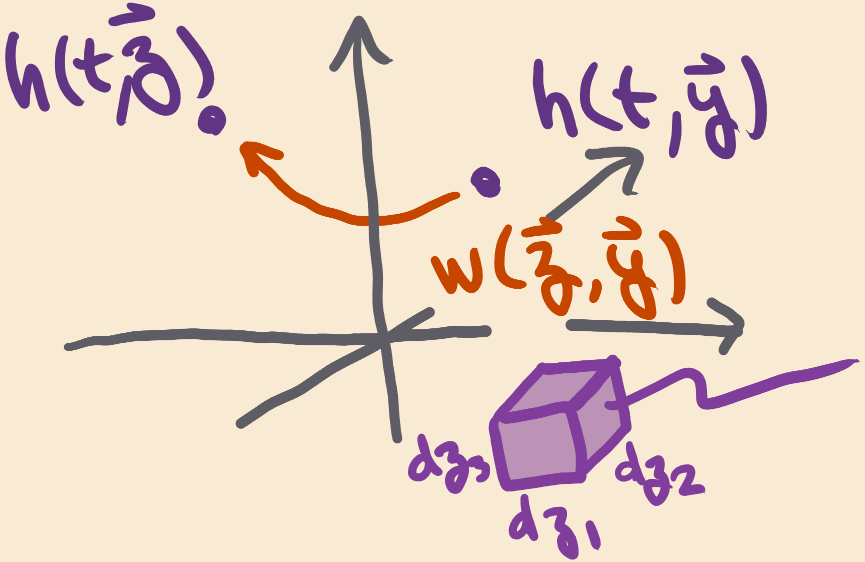
$$\underbrace{\partial_t h(t,\vec{z})}_{\text{neural field (units of voltage)}} = -h(t,\vec{z}) + \int_{\mathbb{R}^p} w(\vec{z},\vec{y})\phi(h(t,\vec{y}))\underbrace{\rho(\mathrm{d}\vec{y})}_{\text{density of neurons in embedding}}$$

neural field
(units of voltage)

embedding
space

activation
function
(gives firing rate,
units of Hz)

density
of neurons
in embedding

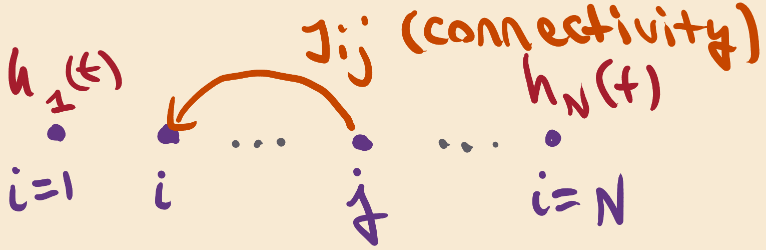


$\rho(\vec{z})dz_1 \cdots dz_p$ neurons

low-rank RNNs (rate neural networks) converge to a neural field equation in R^p as $N \to \infty$ [1]

[1] Valentin Schmutz, Johanni Brea, Wulfram Gerstner
Convergence of redundancy-free spiking neural networks to rate networks

low-rank RNNs (rate neural networks) converge to a neural field equation in R^p as $N \to \infty$ [1]

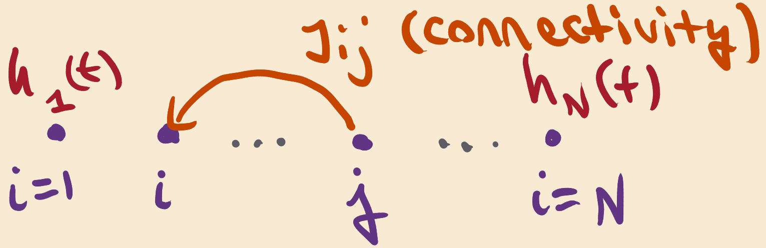


$$\dot{h}_i(t) = -h_i(t) + \sum_{j=1}^{N} J_{ij}\phi(h_j(t))$$

N neurons :
dynamics in R^N

[1] Valentin Schmutz, Johanni Brea, Wulfram Gerstner
Convergence of redundancy-free spiking neural networks to rate networks

low-rank RNNs (rate neural networks) converge to a neural field equation in R^p as $N\to\infty$ [1]



$$\dot{h}_i(t) = -h_i(t) + \sum_{j=1}^{N} J_{ij}\phi(h_j(t))$$

N neurons : dynamics in R^N

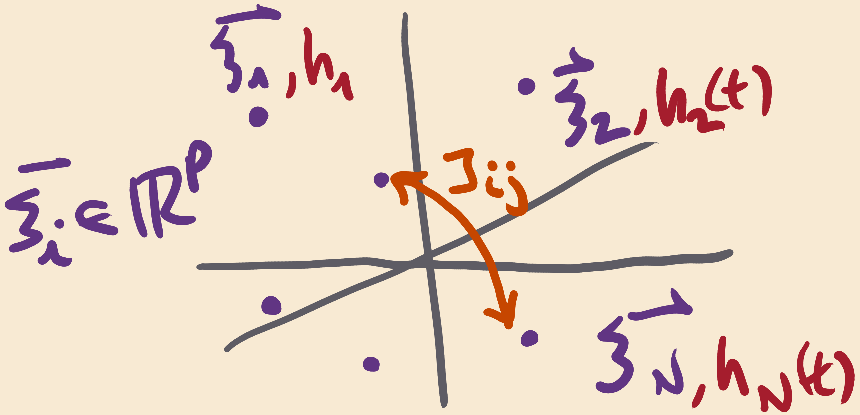
> if J is low-rank, neurons span a subspace of R^N

> $(\xi_{i1},\cdots,\xi_{ip})$ gives the embedding of neuron i

J has rank p (at most)

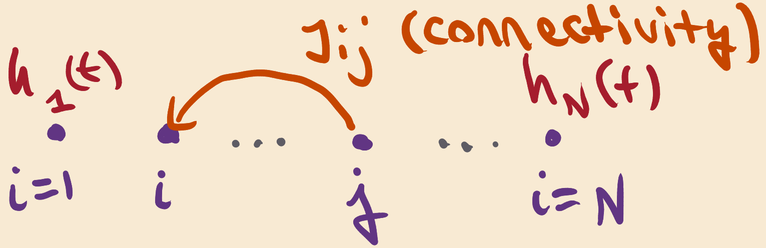
$$J_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_{\mu,i}\tilde{\phi}(\xi_{\mu,j}), \quad \xi_{\mu,i} \sim \mathcal{N}(0,1), \quad \tilde{\phi}(\xi) = \frac{\phi(\xi) - \mathrm{E}[\phi(\xi)]}{\mathrm{Var}[\phi(\xi)]}$$

low-rank vectors



[1] Valentin Schmutz, Johanni Brea, Wulfram Gerstner
Convergence of redundancy-free spiking neural networks to rate networks

low-rank RNNs (rate neural networks) converge to a neural field equation in R^p as $N \to \infty$ [1]



$$\dot{h}_i(t) = -h_i(t) + \sum_{j=1}^{N} J_{ij}\phi(h_j(t))$$

N neurons : dynamics in R^N

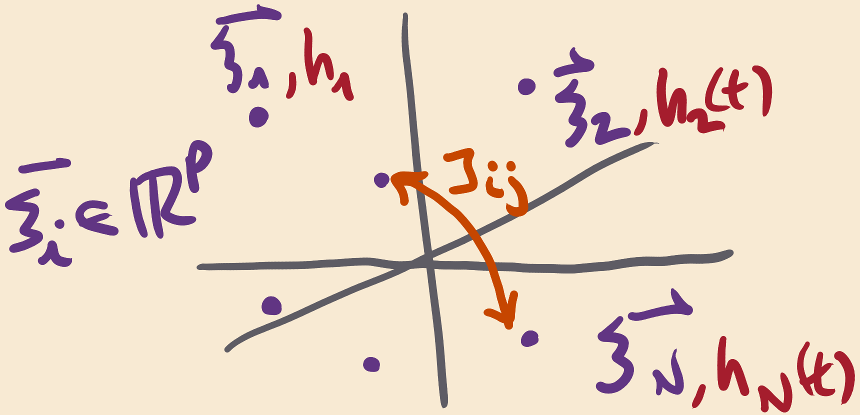
> if J is low-rank, neurons span a subspace of R^N

> $(\xi_{i1}, \cdots, \xi_{ip})$ gives the embedding of neuron i

J has rank p (at most)

$$J_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_{\mu,i}\tilde{\phi}(\xi_{\mu,j}), \quad \xi_{\mu,i} \sim \mathcal{N}(0,1), \quad \tilde{\phi}(\xi) = \frac{\phi(\xi) - \mathrm{E}[\phi(\xi)]}{\mathrm{Var}[\phi(\xi)]}$$

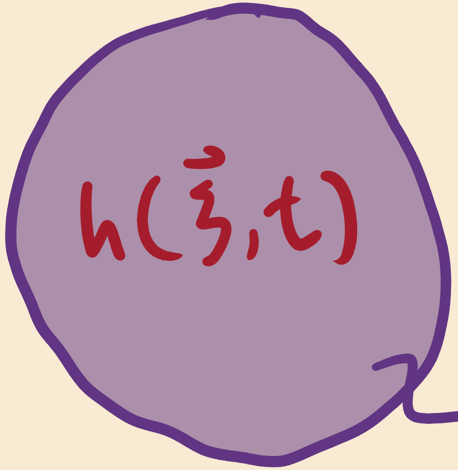
low-rank vectors



as $N \to \infty$

> neurons sample a gaussian

> potential looks smooth



Neural field in R^p

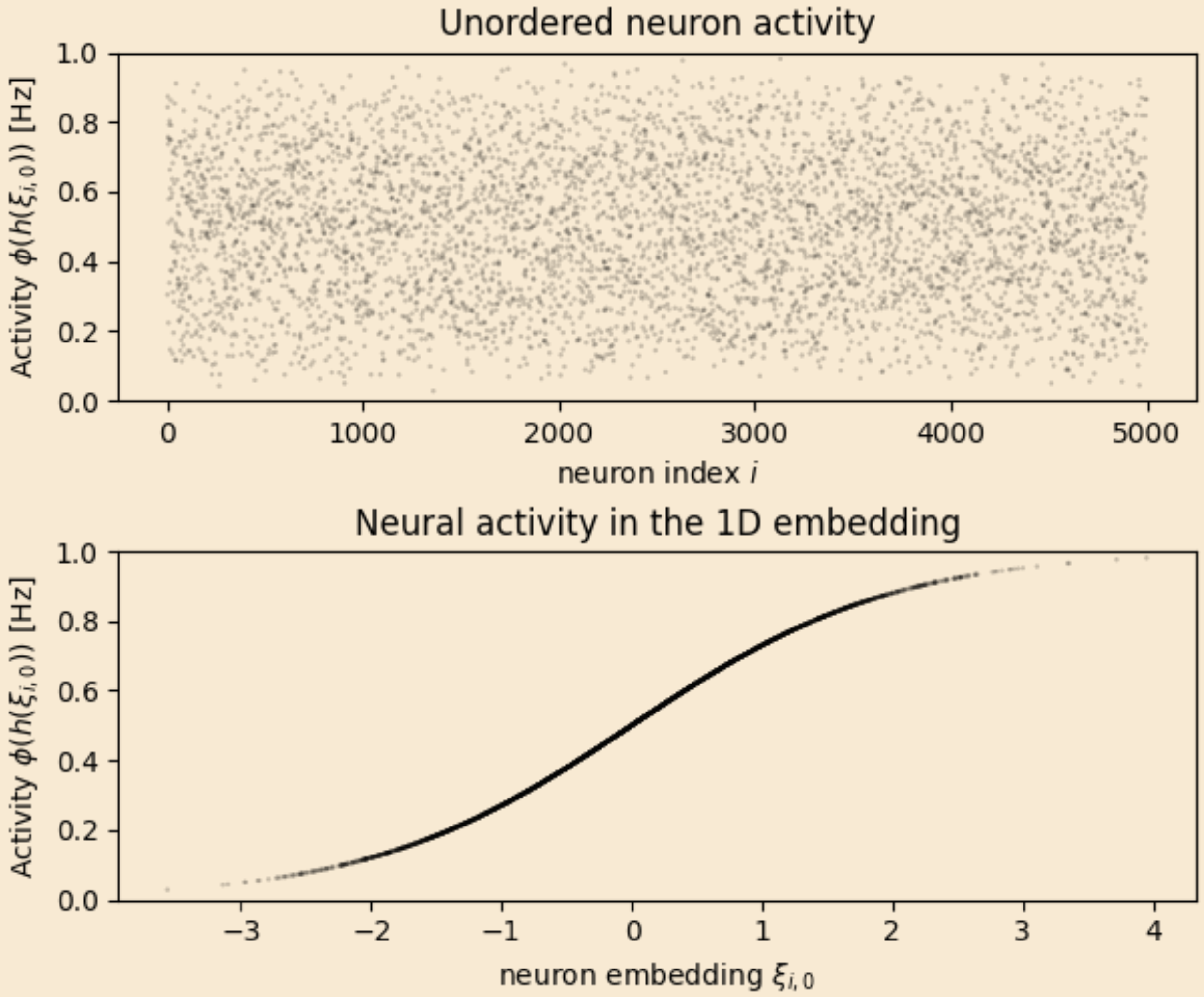
"embedding"

gaussian in $\mathbb{R}^p$

[1] Valentin Schmutz, Johanni Brea, Wulfram Gerstner
Convergence of redundancy-free spiking neural networks to rate networks

demo : embedding in p=1 dimensions

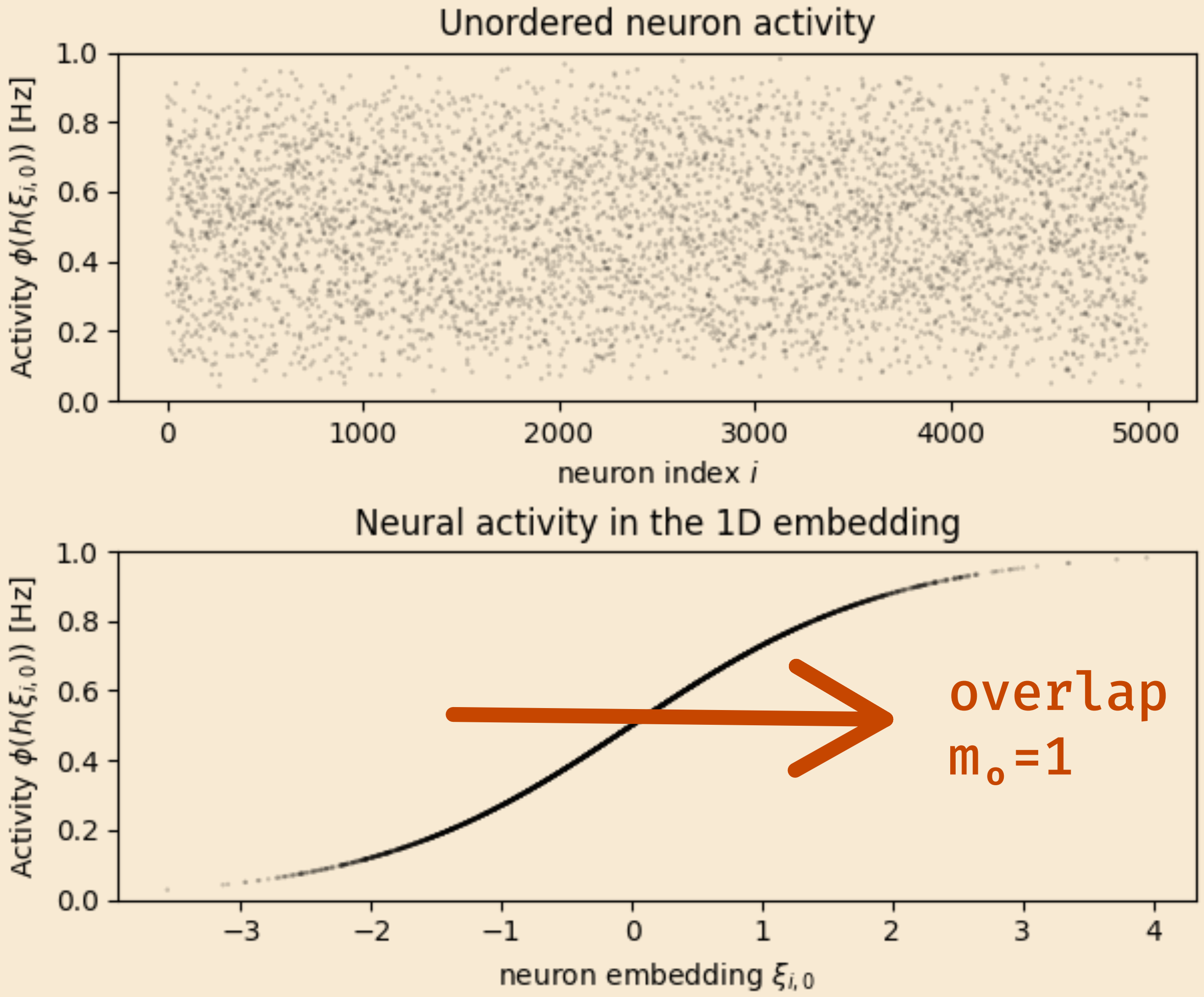
demo : embedding in p=1 dimensions



→ the dynamics is *structured* in the embedding

^^ simulation N=50_000, p=1, phi=sigmoid.
at fixed point μ=0 → overlap m_0=1 (only 5000 neurons are shown)

demo : embedding in p=1 dimensions



→ the dynamics is *structured* in the embedding

→ at the fixed point, $h(t;\xi_0) = \xi_0$

→ the activity follows $\phi(h(t;\xi_0)) = \phi(\xi_0)$ in the embedding

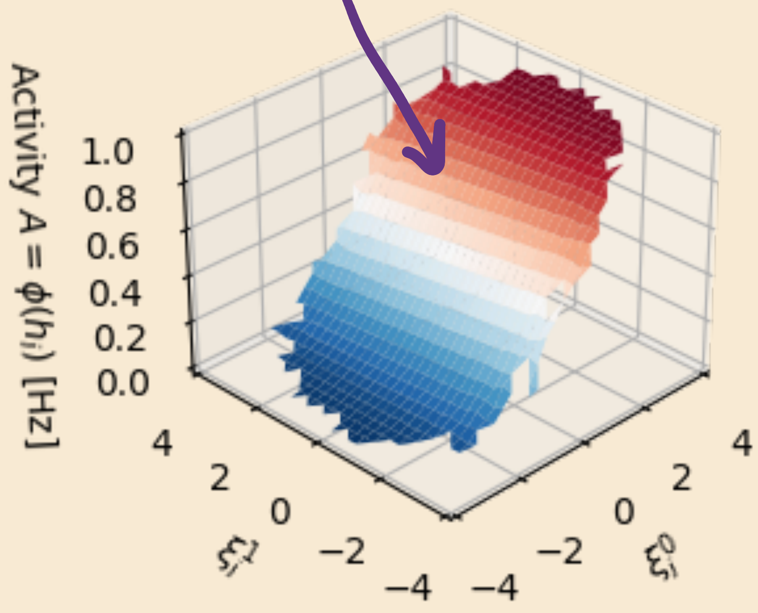
see animations

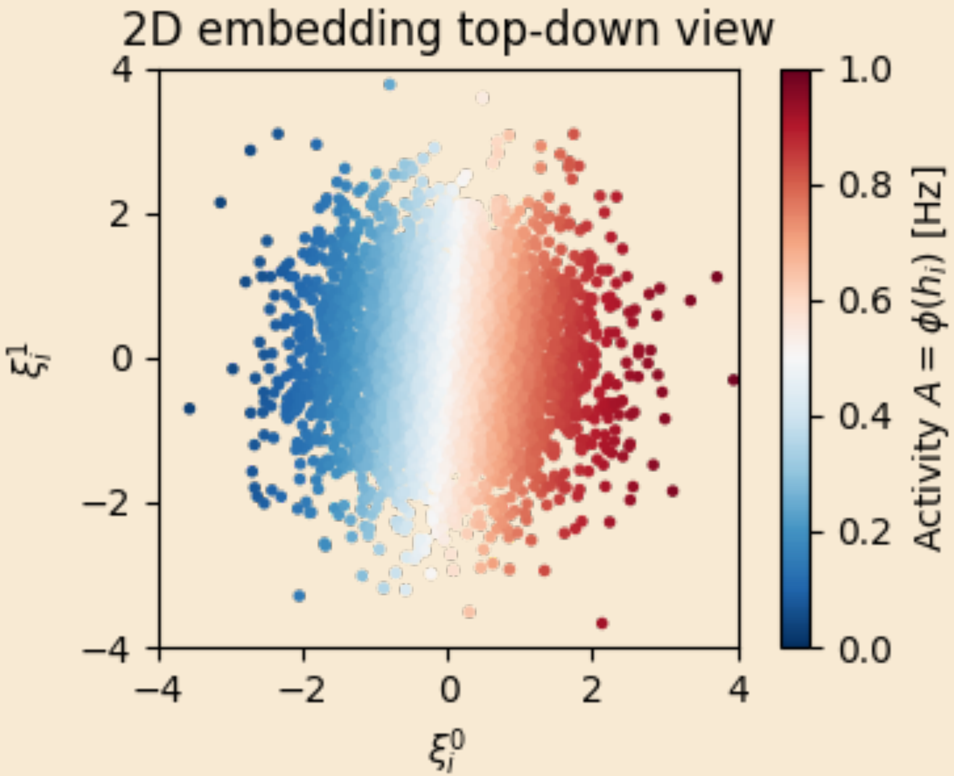
^^ simulation N=50_000, p=1, phi=sigmoid.
at fixed point $\mu=0$ → overlap m_0=1 (only 5000 neurons are shown)

demo : embedding in p=2 dimensions

demo : embedding in p=2 dimensions

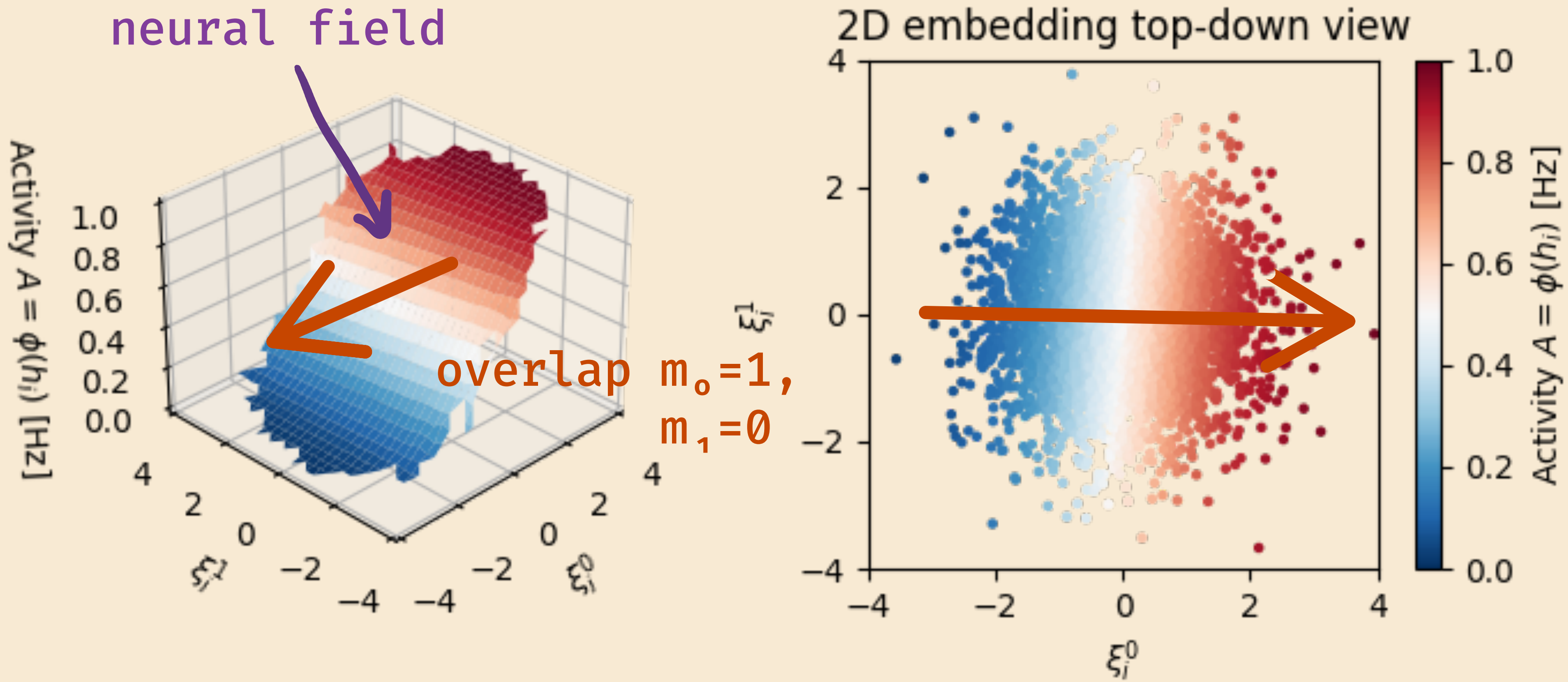
neural field





^^ simulation N=100_000, p=2, phi=sigmoid (only 5000 neurons shown)
overlap m_0=1, m_1=0

demo : embedding in p=2 dimensions



^^ simulation N=100_000, p=2, phi=sigmoid (only 5000 neurons shown)
overlap $m_0=1$, $m_1=0$

→ all the variation is along ξ_0, constant along ξ_1
→ $h(t;\xi_0,\xi_1) = \xi_0$

demo : embedding in p=2 dimensions
- cycling RNN version

demo : embedding in p=2 dimensions
- cycling RNN version

$$\dot{h}_i(t) = -h_i(t) + \sum_{j=1}^{N} \frac{1}{N} \sum_{\mu=1}^{p} \xi_{\mu+1,i} \tilde{\phi}(\xi_{\mu,j}) \phi(h_j(t-\delta))$$

δ : delay

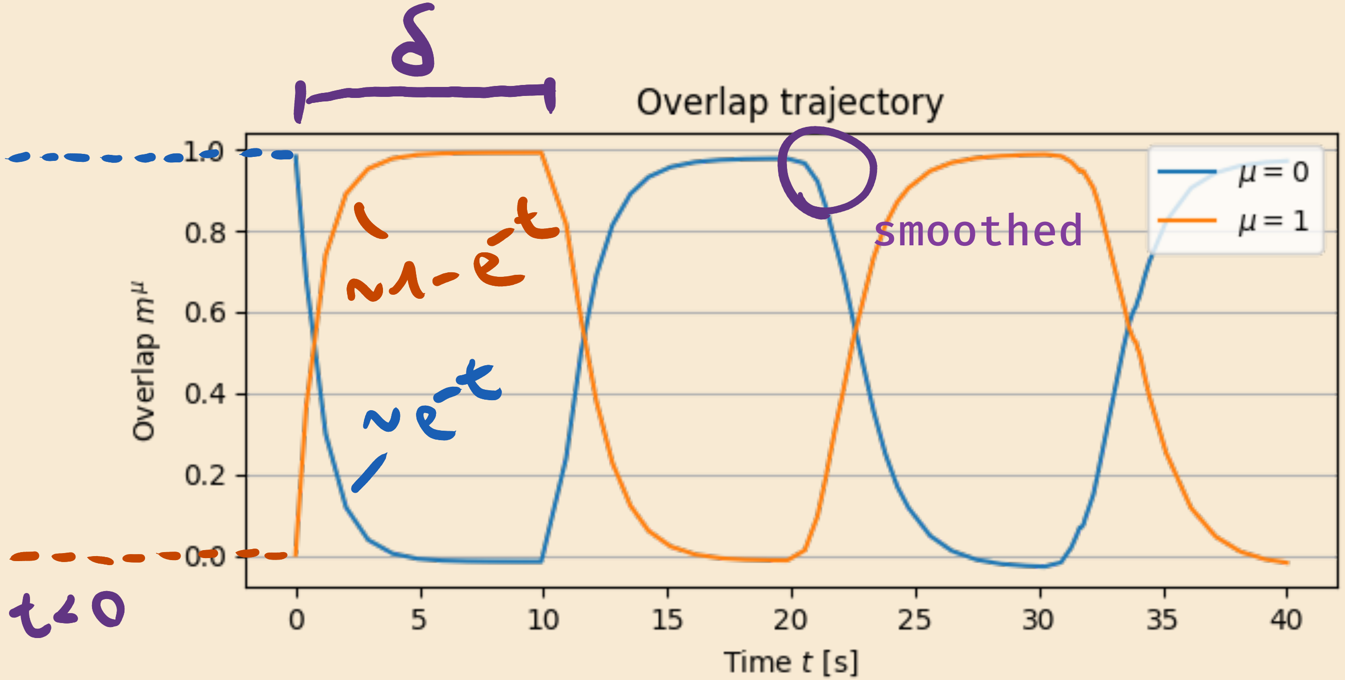
μ+1 : "rolling"

(p+1=1 by convention)

demo : embedding in p=2 dimensions
- cycling RNN version

$$\dot{h}_i(t) = -h_i(t) + \sum_{j=1}^{N} \frac{1}{N} \sum_{\mu=1}^{p} \xi_{\mu+1,i} \tilde{\phi}(\xi_{\mu,j}) \phi(h_j(t-\delta))$$

δ : delay
μ+1 : "rolling"
(p+1=1 by convention)



initial : $h(t<0;\xi_0,\xi_1) = \xi_0$
→ overlaps are exponentials that get smoothed every cycle
→ oscillations flatten out over time

see animation

^^ N=20_000, p=2, delta=10, shift=1, phi=sigmoid

can we go to an even lower dimension ?

can we go to an even lower dimension ?

number of dimensions

"embedding"
= neural field
equation in R^p

discrete RNN
in R^N

as $N \to \infty$, dynamics
gain spatial structure

(not true in general)

can we go to an even lower dimension ?

number of dimensions

1 ⟵——————— p ——————— … ———————⟶ N

dynamics in
one dimension

"embedding"
= neural field
equation in R^p

discrete RNN
in R^N

mapping φ of R^p to [0,1] (or R)

→ what analytical conditions ?
→ continuity ?
bijectivity ? both ?

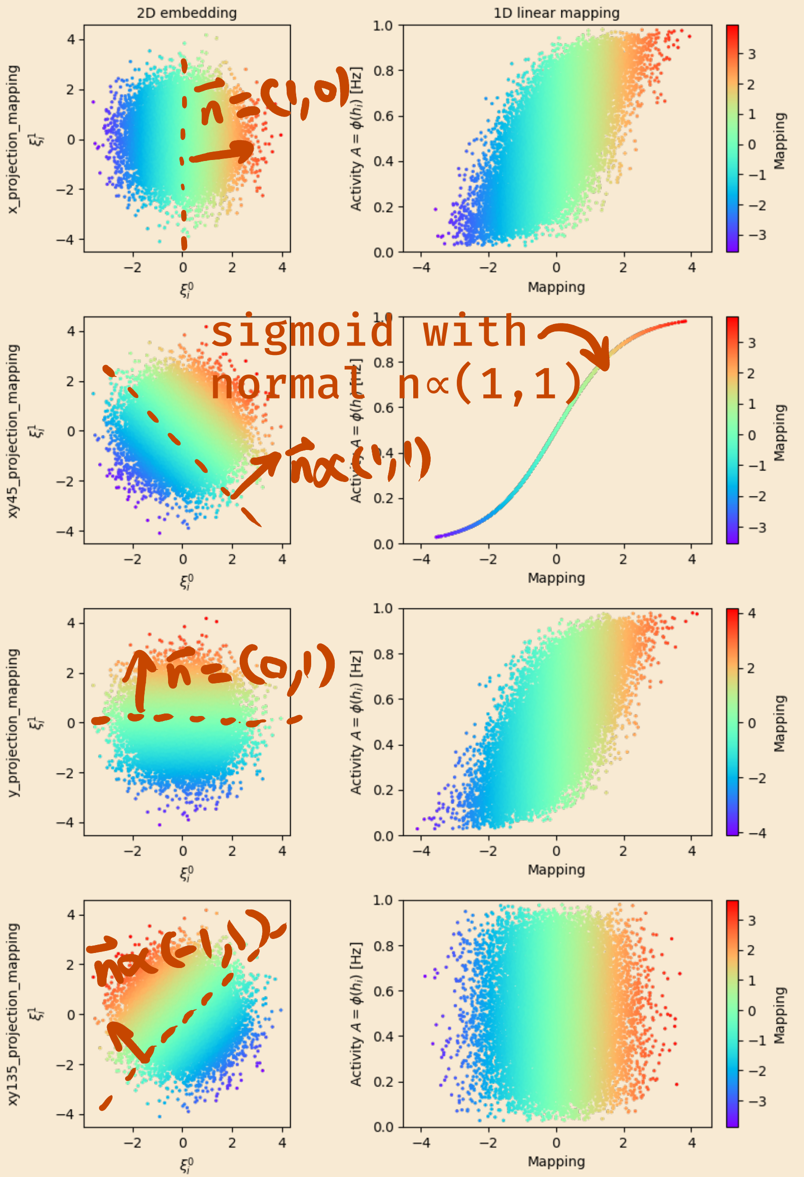
as $N \to \infty$, dynamics
gain spatial structure
(not true in general)

mapping a $[0,1]^2$ square into a $[0,1]$ segment [2]

[2] Kharazishvili, A. (2017).
Strange Functions in Real Analysis (3rd ed.)

mapping a $[0,1]^2$ square into a $[0,1]$ segment [2]

- Peano : surjective and continuous
  > easy example : linear projections
  (see animation)
  > clearly not bijective



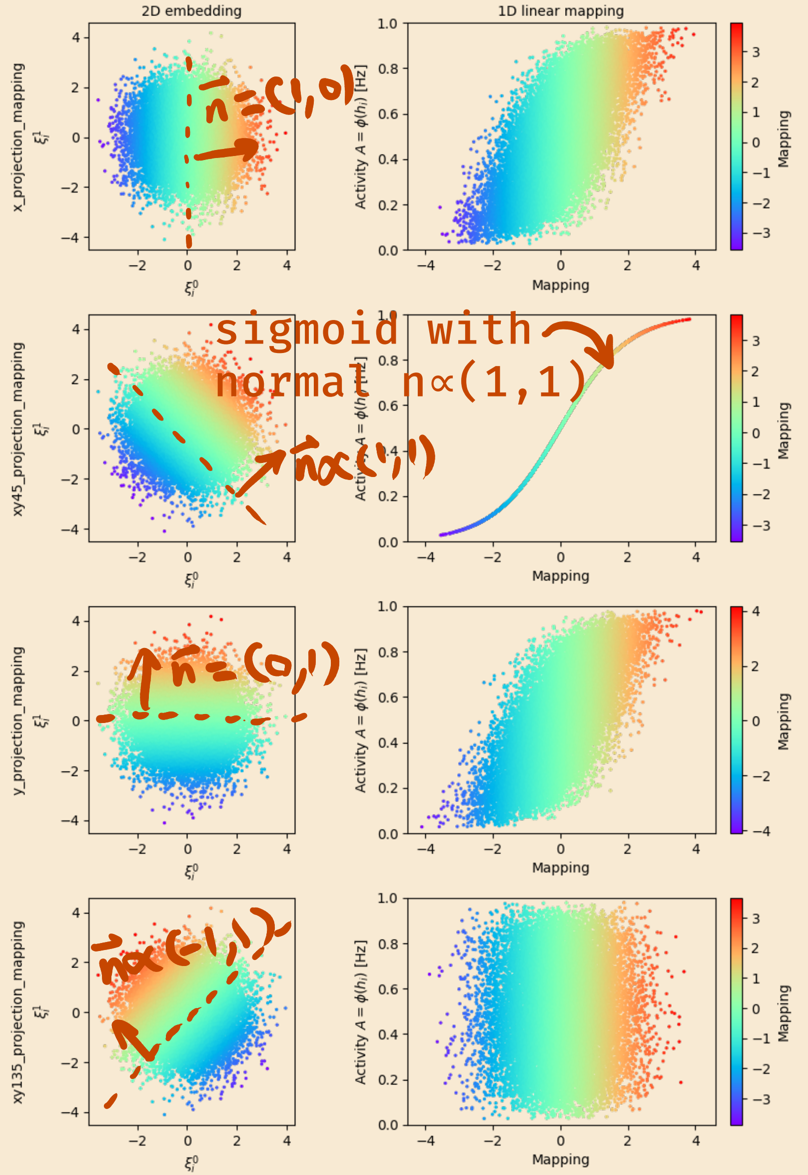
linear mappings of a RNN in state $h=\xi_0+\xi_1$
N=20_000, p=2, phi=sigmoid >>

[2] Kharazishvili, A. (2017).
Strange Functions in Real Analysis (3rd ed.)

mapping a $[0,1]^2$ square into a $[0,1]$ segment [2]

- Peano : surjective and continuous
  > easy example : linear projections
  (see animation)
  > clearly not bijective

- Cantor : bijective (= 1-to-1 mapping)
  > example : see next slides



linear mappings of a RNN in state $h=\xi_0+\xi_1$
N=20_000, p=2, phi=sigmoid

\>>

[2] Kharazishvili, A. (2017).
Strange Functions in Real Analysis (3rd ed.)

mapping a $[0,1]^2$ square into a $[0,1]$ segment [2]

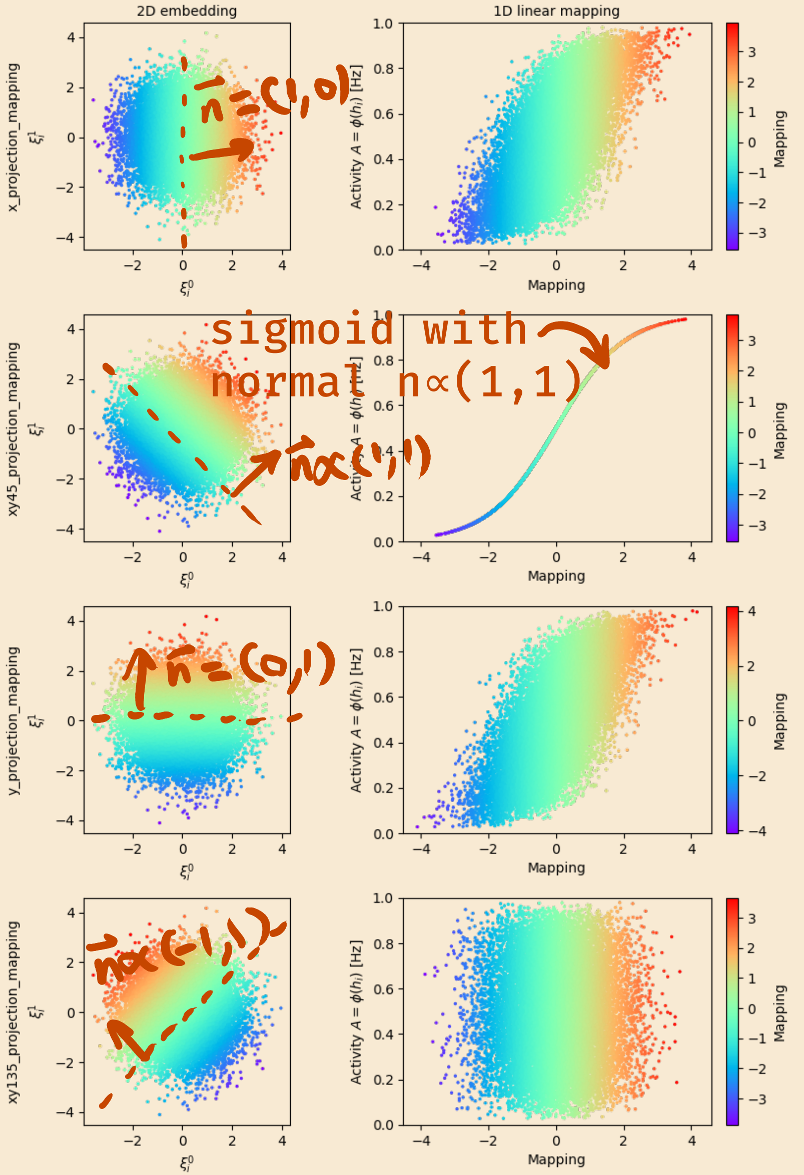
- Peano : surjective and continuous
  > easy example : linear projections (see animation)
  > clearly not bijective

- Cantor : bijective (= 1-to-1 mapping)
  > example : see next slides

Fact : there is no homeomorphism (continuous bijective function) between $[0,1]^2$ and $[0,1]$

linear mappings of a RNN in state $h=\xi_0+\xi_1$
N=20_000, p=2, phi=sigmoid

>>



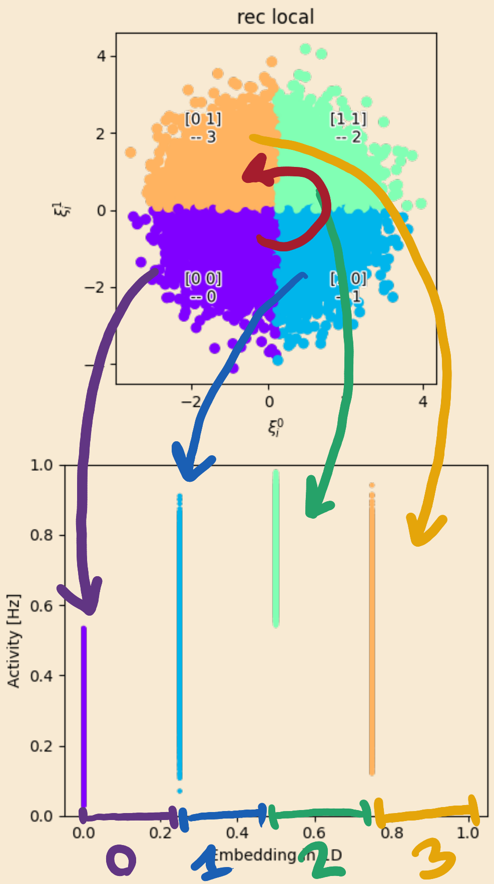
[2] Kharazishvili, A. (2017).
Strange Functions in Real Analysis (3rd ed.)

a bijective fractal mapping $[0,1]^2 \to [0,1]$ [3]

[3] Cong Dan Pham. (2016)
The existence of a measure-preserving bijection from a unit square to unit segment

a bijective fractal mapping $[0,1]^2 \to [0,1]$ [3]

nrec=1



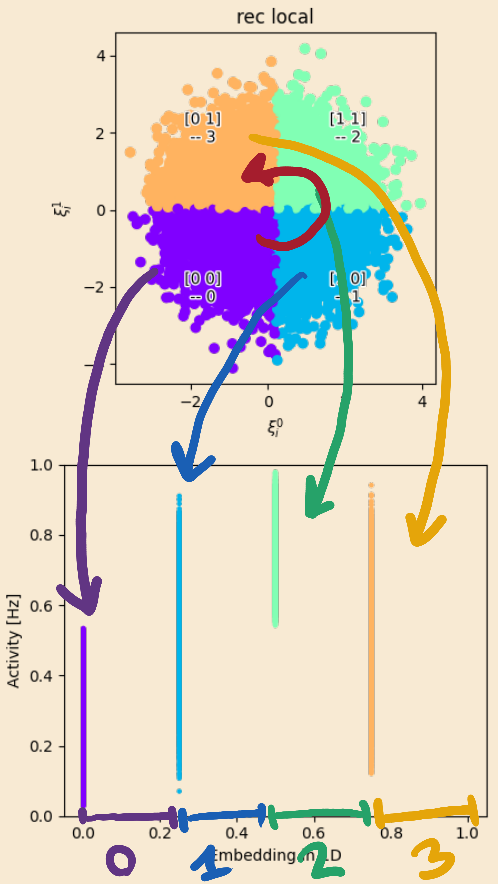
4 bins

1. split into 4 quadrants
2. assign mapping coordinate
3. recurse for every quadrant

[3] Cong Dan Pham. (2016)
The existence of a measure-preserving bijection from a unit square to unit segment

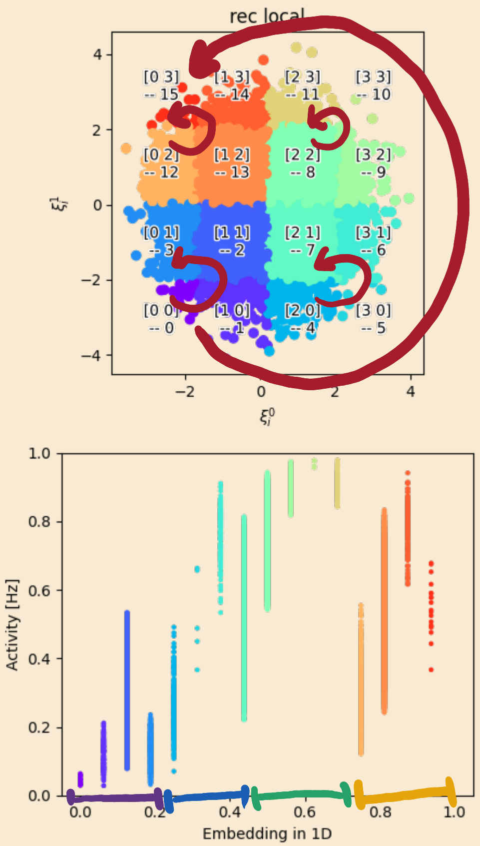
a bijective fractal mapping $[0,1]^2 \to [0,1]$ [3]

nrec=1



4 bins

nrec=2



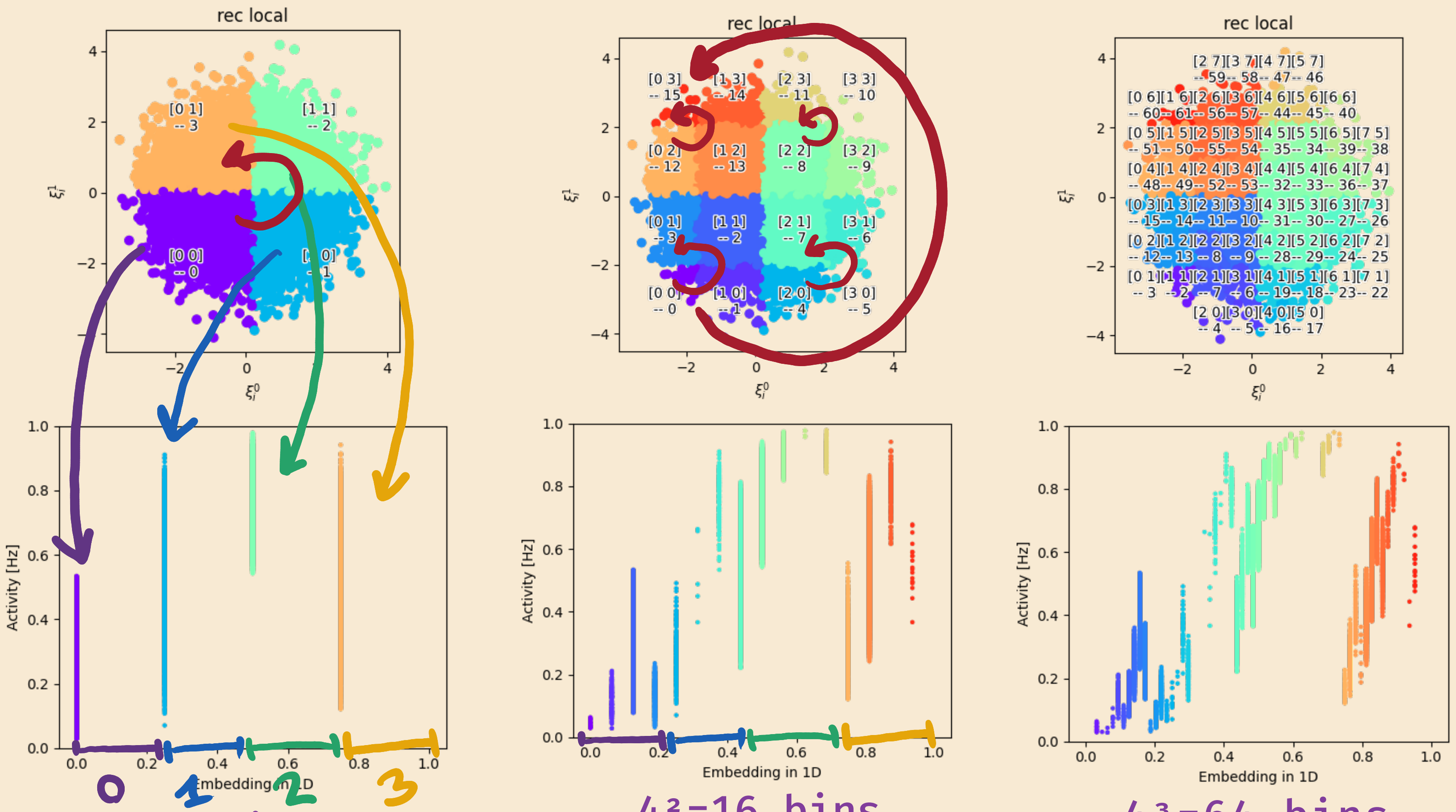
4^2=16 bins

1. split into 4 quadrants
2. assign mapping coordinate
3. recurse for every quadrant

[3] Cong Dan Pham. (2016)
The existence of a measure-preserving bijection from a unit square to unit segment

a bijective fractal mapping $[0,1]^2 \to [0,1]$ [3]

nrec=1 nrec=2 nrec=3



4 bins

4^2=16 bins

4^3=64 bins

1. split into 4 quadrants
2. assign mapping coordinate
3. recurse for every quadrant

→ discontinuous, but bijective

see animation
- mean activity in each bin
- there is some "locality"

[3] Cong Dan Pham. (2016)
The existence of a measure-preserving bijection from a unit square to unit segment

can we write a neural field
in an even lower dimension ?

can we write a neural field in an even lower dimension ?

number of dimensions

1 — p — … — N

mapping in [0,1]

embedding in R^p

discrete RNN in R^N

1. simulate neural field in R^p

2. map R^p to [0,1] using φ

can we write a neural field in an even lower dimension ?

number of dimensions

1 … p … N

mapping in $[0,1]$

embedding in R^p

discrete RNN in R^N

1. simulate neural field in R^p
2. map R^p to $[0,1]$ using φ

→ can we do it the other way around ?

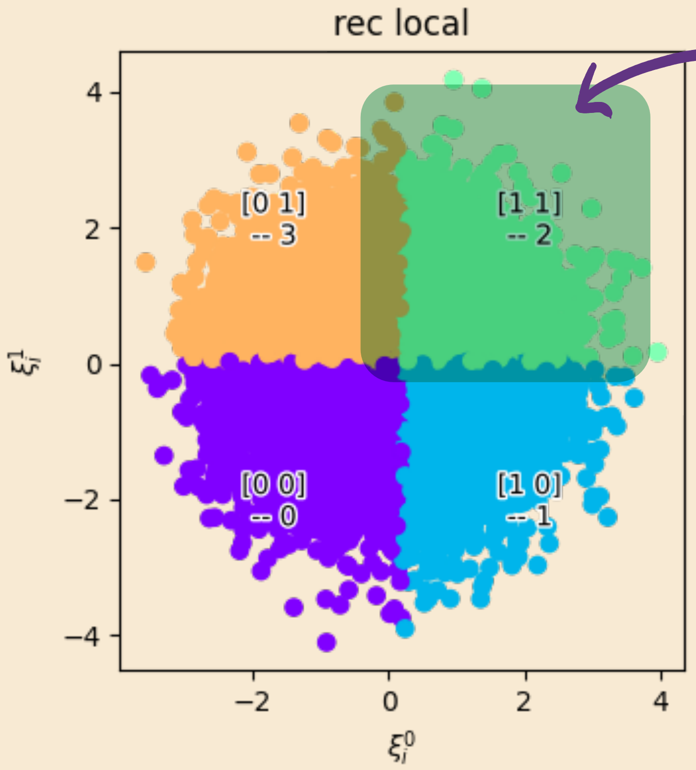
1. simulate dynamics in $[0,1]$
2. assert that the dynamics in $[0,1]$ are the same as in R^p (e.g. map $[0,1]$ to R^p using φ^{-1}, and plot to compare)

numerical aspects of simulating fractal mappings

numerical aspects of simulating fractal mappings

smoothness in $\mathbb{R}^p$ $\Rightarrow$ all neurons inside each bin are similar

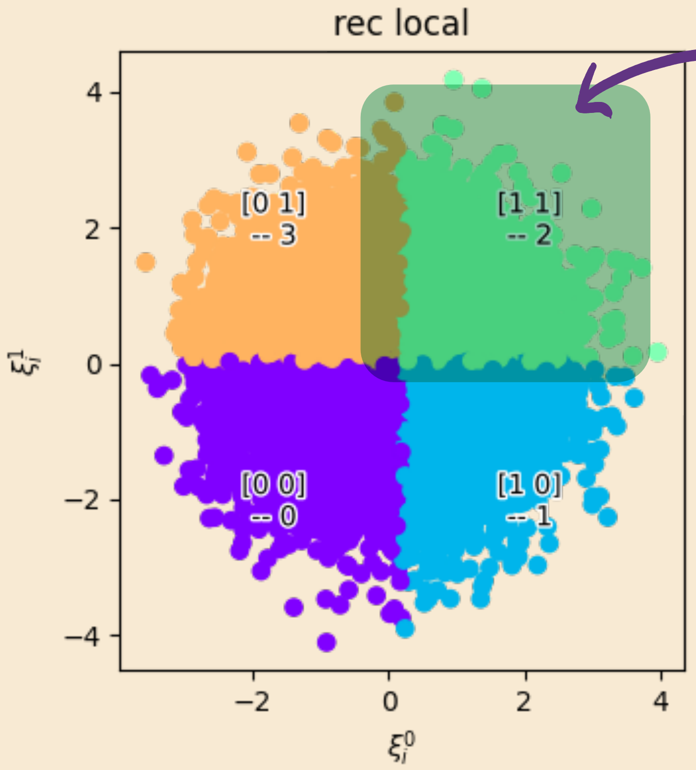
$\Rightarrow$ take the average potential



$$H_2 := \frac{1}{\#\text{neurons in bin } \alpha=2} \sum_{i \in 2} h_i$$

numerical aspects of simulating fractal mappings

smoothness in R^p ⇒ all neurons inside each bin are similar
⇒ take the average potential



$$H_2 := \frac{1}{\#\text{neurons in bin } \alpha=2} \sum_{i \in 2} h_i$$

$$H_\alpha(t) = \frac{1}{|\alpha|} \sum_{i\in\alpha} h_i(t)$$

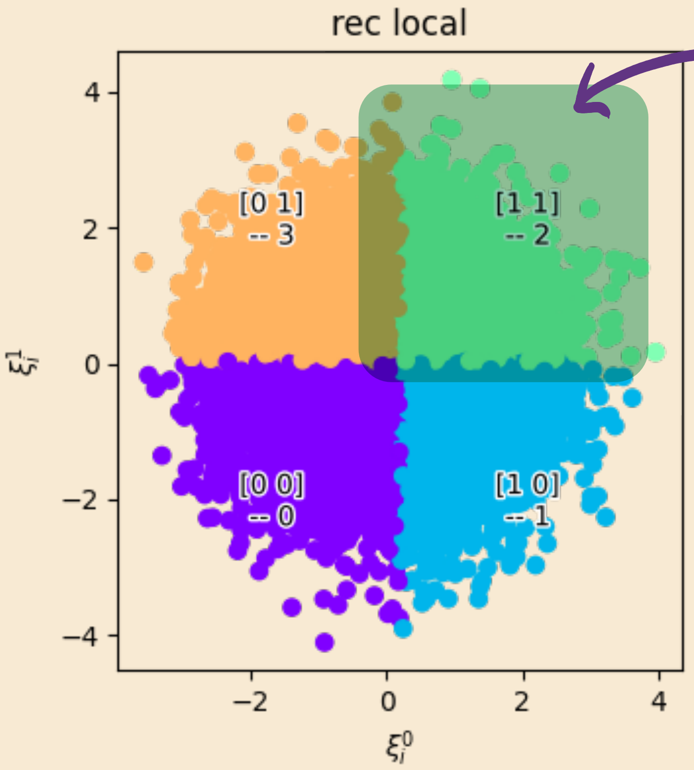
⇊ discretized dynamics, with rescaled J

$$\dot{H}_\alpha(t) = -H_\alpha(t) + \sum_{\beta \in \text{segments of length } 4^{-n}} \tilde{J}_{\alpha,\beta}\phi(H_\beta(t))$$

$$\tilde{J}_{\alpha,\beta} = \frac{1}{|\alpha|} \sum_{i\in\alpha}\sum_{j\in\beta} J_{ij}$$

numerical aspects of simulating fractal mappings

smoothness in $\mathbb{R}^p$ ⇒ all neurons inside each bin are similar
⇒ take the average potential



$$H_2 := \frac{1}{\#\text{neurons in bin } \alpha=2} \sum_{i\in 2} h_i$$

$$H_\alpha(t) = \frac{1}{|\alpha|} \sum_{i\in\alpha} h_i(t)$$

⇊ discretized dynamics, with rescaled J

$$\dot{H}_\alpha(t) = -H_\alpha(t) + \sum_{\beta\in\text{segments of length } 4^{-n}} \tilde{J}_{\alpha,\beta}\,\phi(H_\beta(t))$$

$$\tilde{J}_{\alpha,\beta} = \frac{1}{|\alpha|} \sum_{i\in\alpha} \sum_{j\in\beta} J_{ij}$$

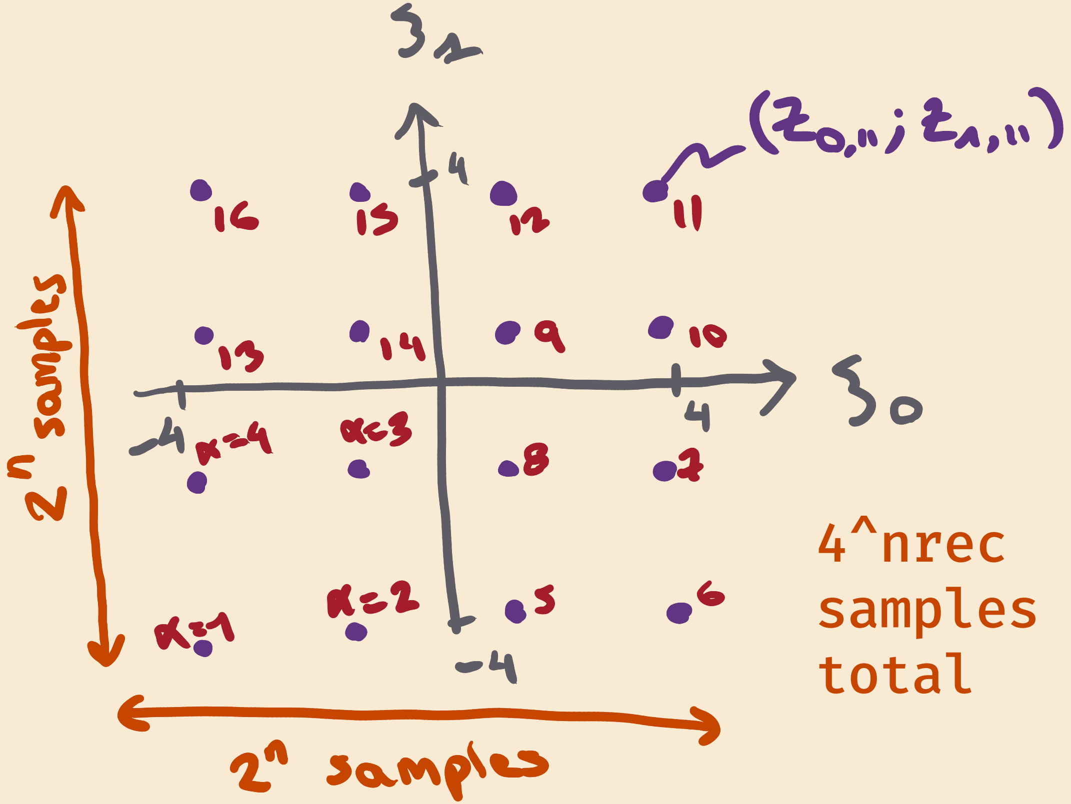
→ J_αβ converges to a (possibly discontinuous !)
connectivity kernel w(α,β) as nrec→∞

Note : since J_ij is low-rank, (we can prove that) J_αβ is too !
→ good news, because low-rank RNNs
are computationally cheap to simulate

directly computing the connectivity $J_{\alpha\beta}$ in [0,1]

directly computing the connectivity J_αβ in [0,1]

steps : 1. discretize the PDF on a grid
2. apply mapping
3. generate J_αβ

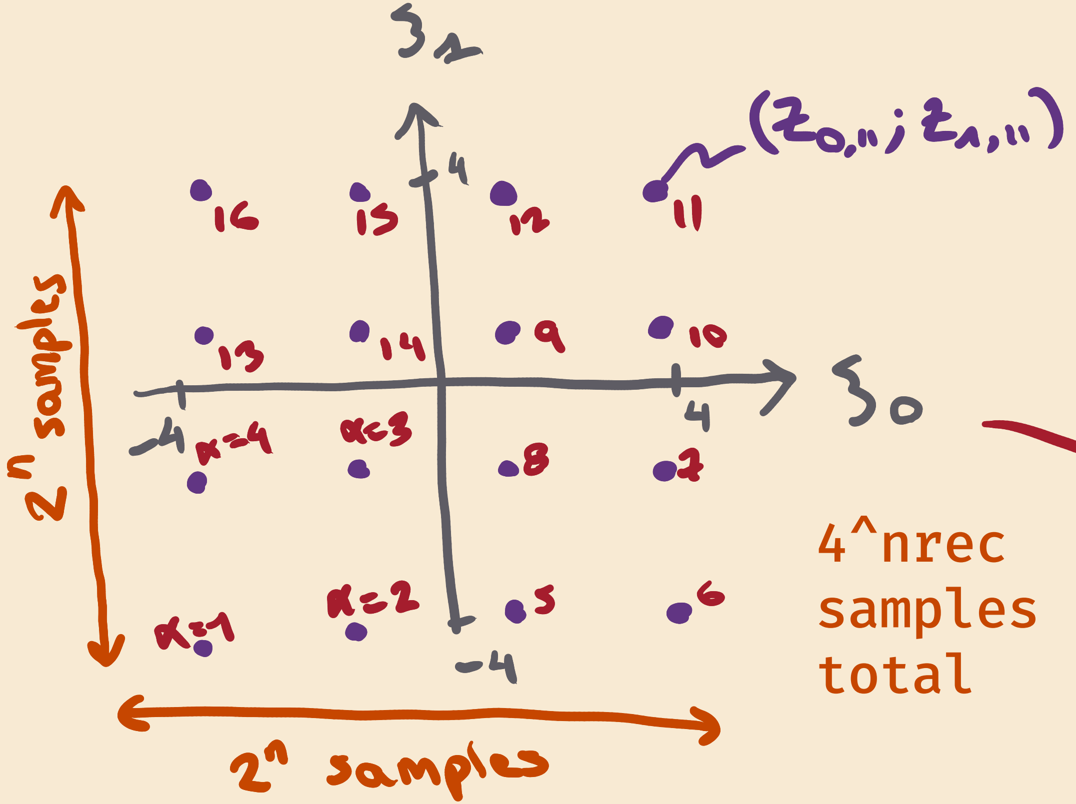


4^nrec
samples
total

Note : gaussian wings decay fast
⇒ approximate PDF by compact support
⇒ the $[-4,4]^2$ bounding box approximation
is OK

directly computing the connectivity J_αβ in [0,1]

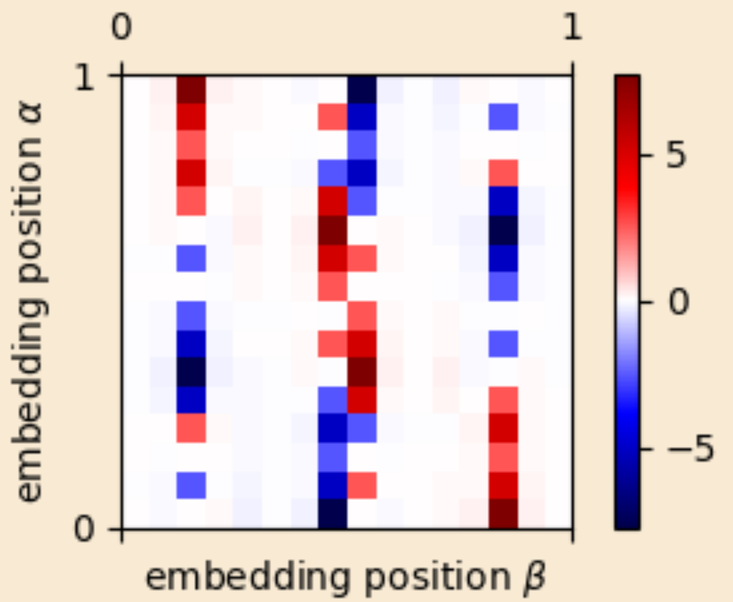
steps : 1. discretize the PDF on a grid
2. apply mapping
3. generate J_αβ



4^nrec
samples
total

gaussian
PDF

$$\tilde{F}_{\mu,\alpha} = Z_{\mu,\alpha}, \quad \tilde{G}_{\mu,\alpha} = \tilde{\phi}(Z_{\mu,\alpha}), \quad \tilde{J}_{\alpha,\beta} = \tilde{\rho}(Z_{:,\beta}) \sum_{\mu=1}^{p} \tilde{F}_{\mu,\alpha} \tilde{G}_{\mu,\beta}$$



^^ connectivity inside
the mapping, nrec=2

Note : gaussian wings decay fast
⇒ approximate PDF by compact support
⇒ the $[-4,4]^2$ bounding box approximation
is OK

are dynamics in [0,1] are the same as in R^p ?

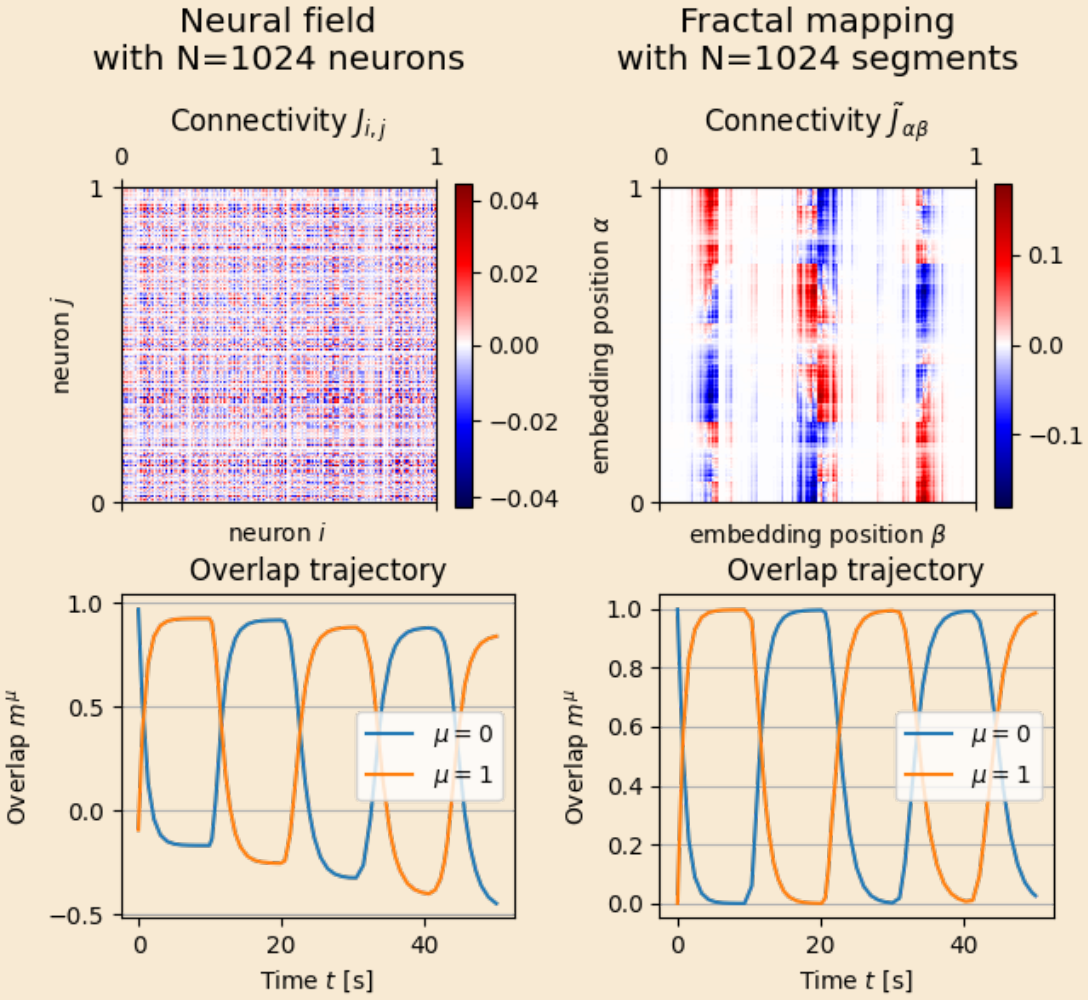
are dynamics in [0,1] are the same as in R^p ?

> for a fair comparison, we *downsample* the mapping to N=1024 segments

> the neural field is the "ground truth"

Note : the neural field drifts due to finite N effects

are dynamics in [0,1] are the same as in R^p ?



> for a fair comparison, we *downsample* the mapping to N=1024 segments
> the neural field is the "ground truth"

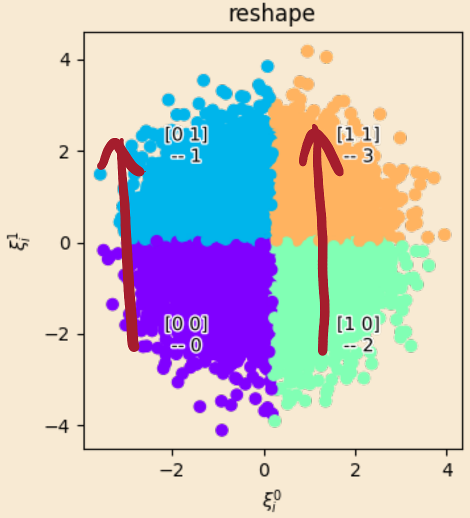
→ fractal mapping has oscillations

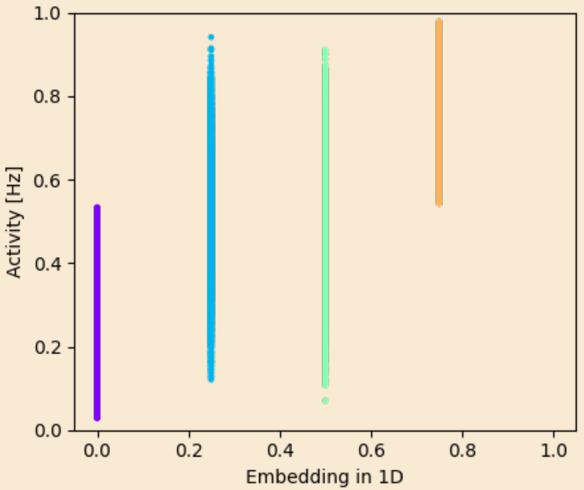
Note : the neural field drifts due to finite N effects

the reshape mapping

the reshape mapping

nrec=1





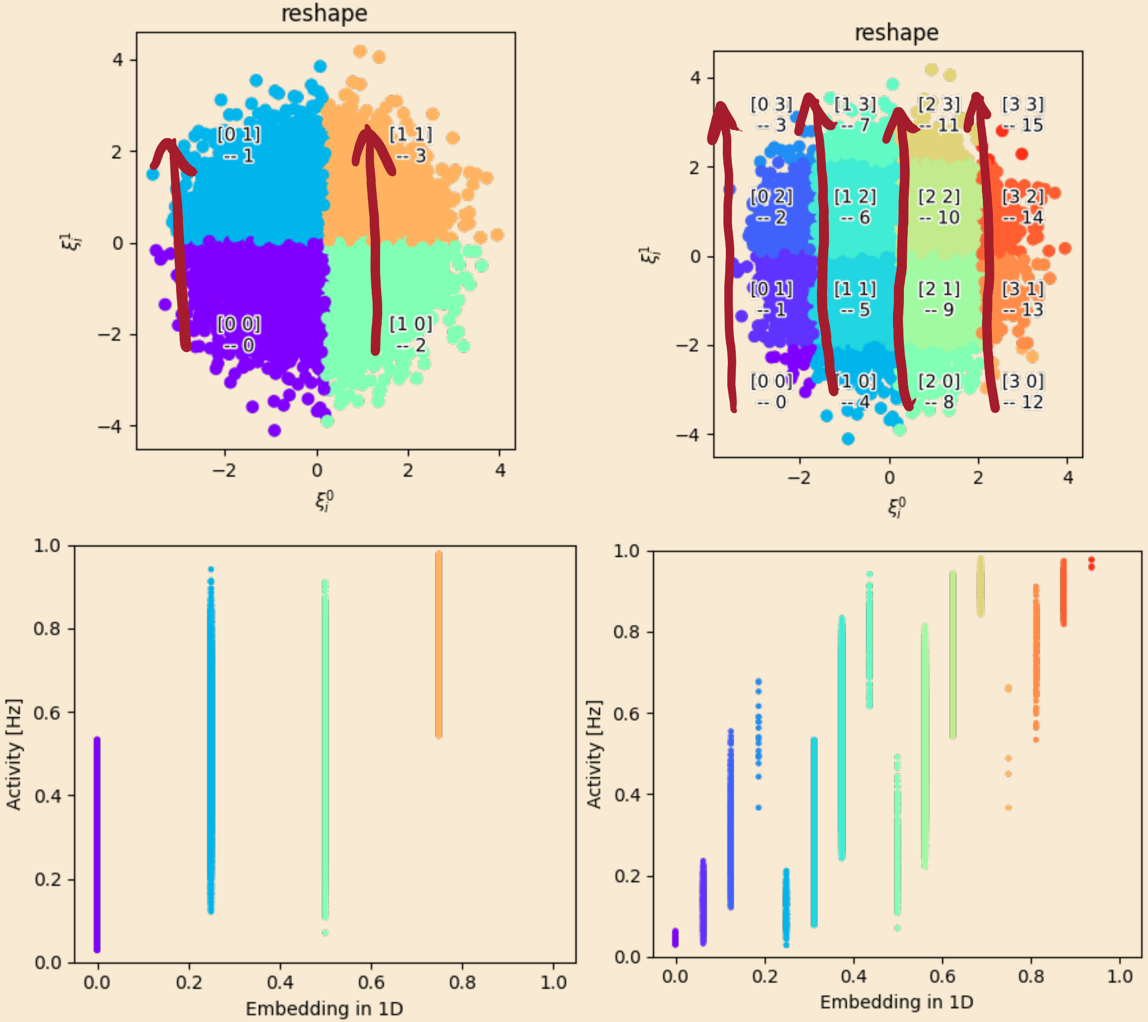
steps

1. split into 4^nrec quadrants
2. enumerate every square column by column

the reshape mapping

nrec=1

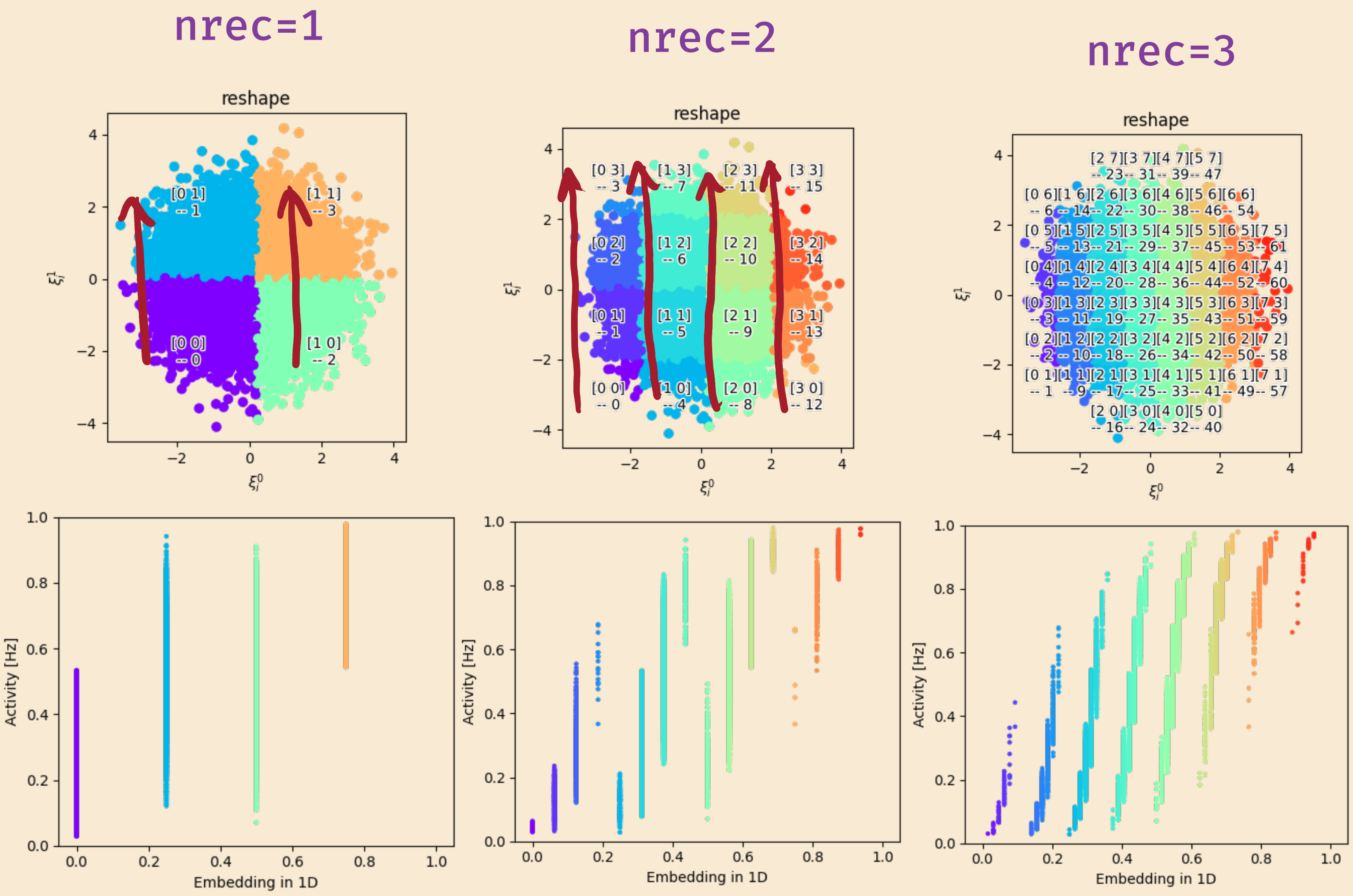
nrec=2



steps

1. split into 4^nrec quadrants
2. enumerate every square column by column

the reshape mapping

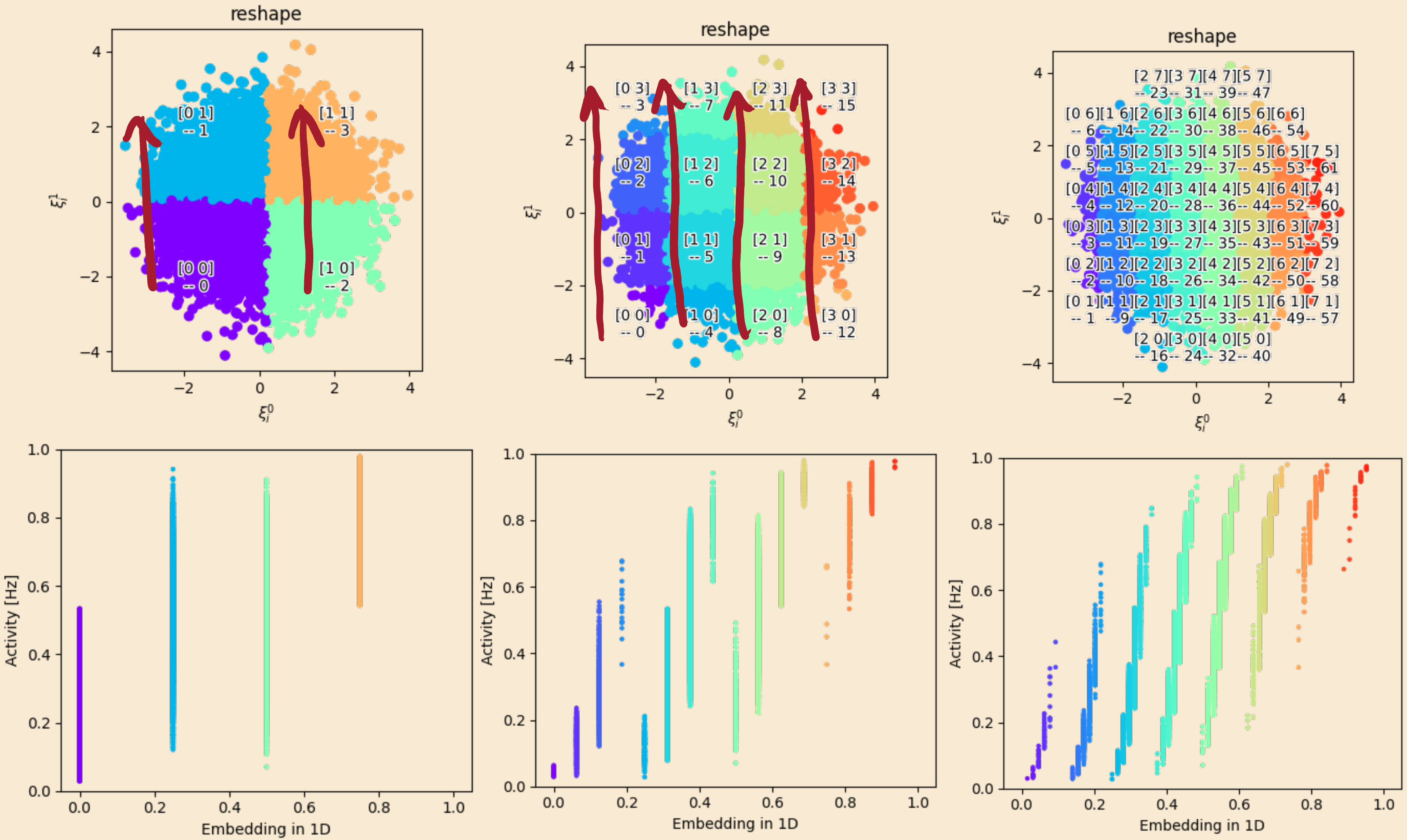


steps

1. split into 4^nrec quadrants
2. enumerate every square column by column

the reshape mapping

nrec=1 nrec=2 nrec=3



steps

1. split into 4^nrec quadrants
2. enumerate every square column by column

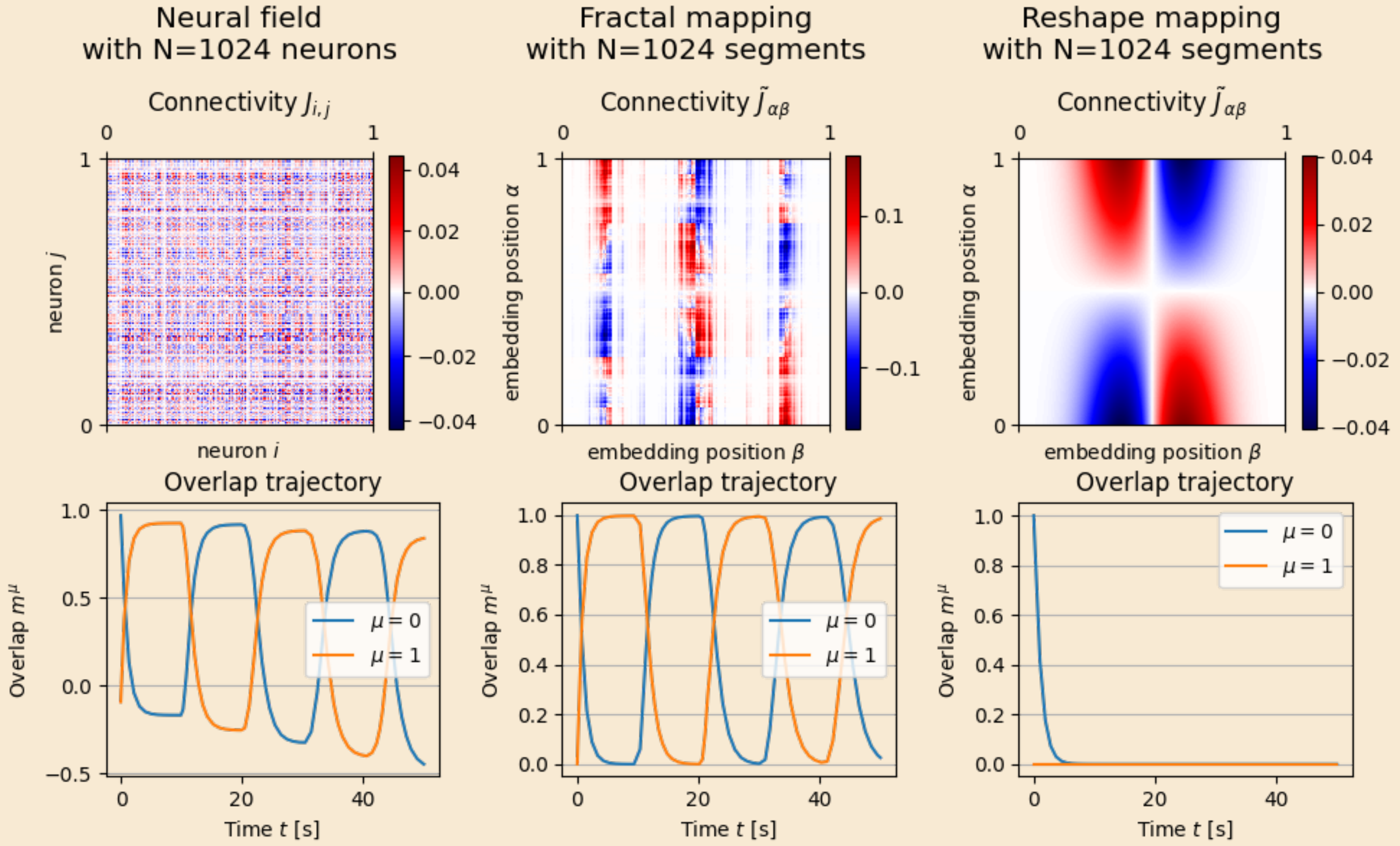
→ in the limit, tends to a projection along the ξ_0 axis

→ "locality" only along ξ_0

→ limit is not bijective

are dynamics in [0,1] are the same as in R^p ?

are dynamics in [0,1] are the same as in R^p ?



> for a fair comparison, we *downsample* the mappings to N=1024 segments

> the neural field is the "ground truth"

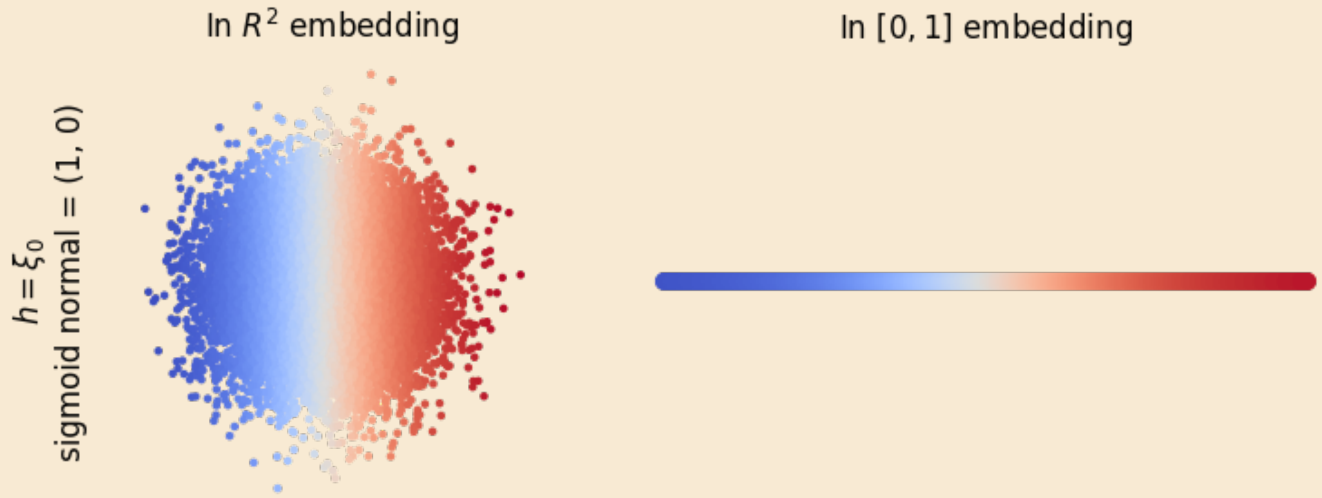
→ fractal mapping has oscillations

→ reshape mapping does not

Note : the neural field drifts due to finite N effects

why doesn't the reshape mapping work ?

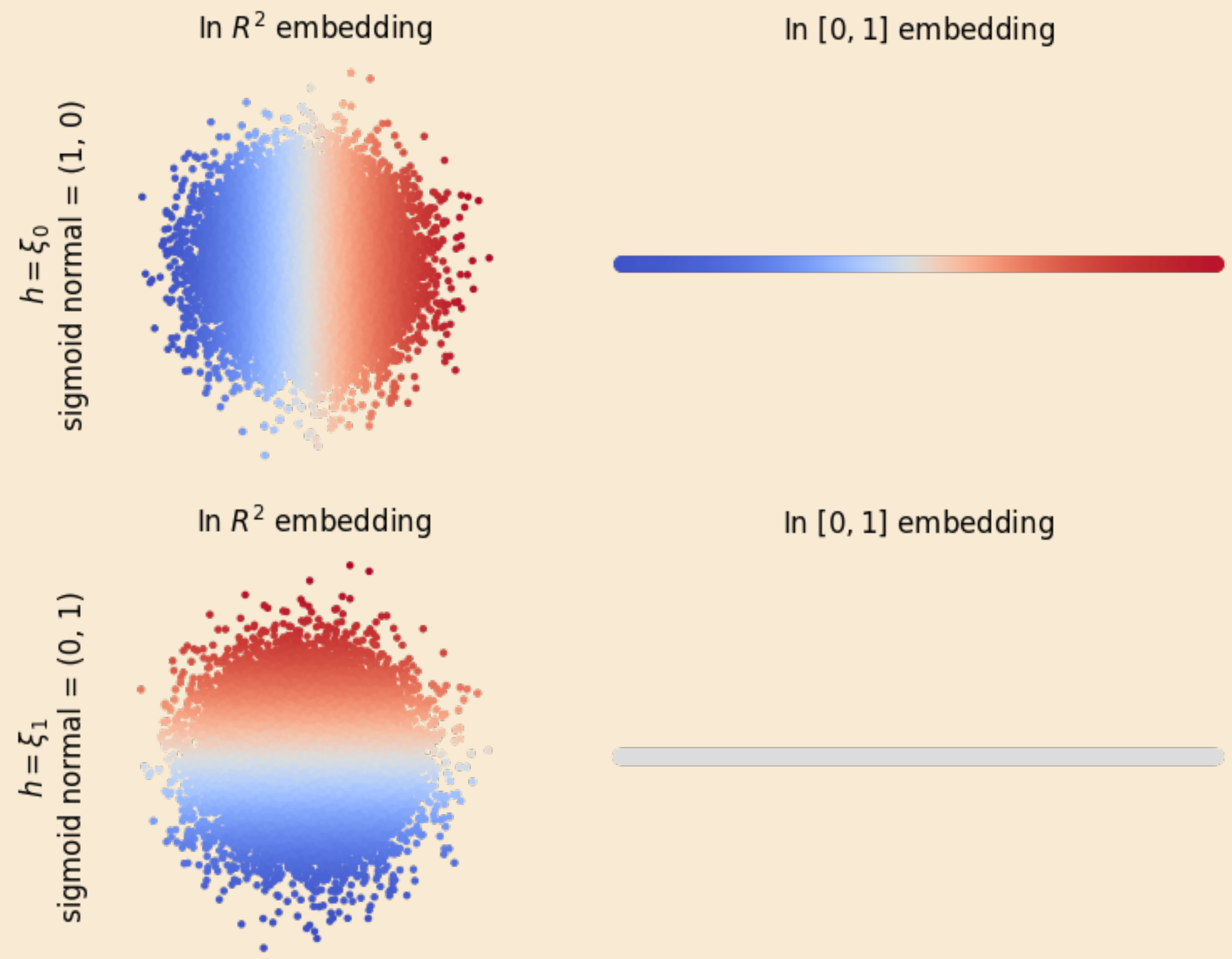
why doesn't the reshape mapping work ?



→ the reshape mapping converges to a projection along ξ_0

> in the state ξ_0, the projection can "encode the variation"

why doesn't the reshape mapping work ?



→ the reshape mapping converges to a projection along ξ_0

> in the state ξ_0, the projection can "encode the variation"

> but in the state ξ_1, all the variation gets "averaged out"

summary of mappings so far

Mapping	Bijective in limit	"Locality"	Equivalent dynamics
Recursive Local	Yes	Yes	Yes
Reshape	No (→ projection)	Only along one axis	No

summary of mappings so far

Mapping	Bijective in limit	"Locality"	Equivalent dynamics
Recursive Local	Yes	Yes	Yes
Reshape	No (→ projection)	Only along one axis	No

what's next ?
→ what conditions on the mapping
must be met for equivalent dynamics in [0,1] ?

summary of mappings so far

Mapping	Bijective in limit	"Locality"	Equivalent dynamics
Recursive Local	Yes	Yes	Yes
Reshape	No (→ projection)	Only along one axis	No

what's next ?

→ what conditions on the mapping must be met for equivalent dynamics in [0,1] ?

→ explore tradeoff between dimensionality and "regularity"
in $\mathbb{R}^2$ the kernel is continuous -- but high dimension
in [0,1] we lose continuity -- but low dimension
⇒ here we have presented the first example of such equivalent neural fields

extra slides →

rank-p RNNs converge to
a neural field equation in R^p as $N \to \infty$

rank-p RNNs converge to a neural field equation in $\mathbb{R}^p$ as $N \to \infty$

$$\dot{h}_i(t) = -h_i(t) + \sum_{j=1}^{N} J_{ij}\phi(h_j(t))$$

$$J_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_{\mu,i}\tilde{\phi}(\xi_{\mu,j})$$

N-dimensional system

$$\xi_{\mu,i} \sim \mathcal{N}(0,1), \quad \tilde{\phi}(\xi) = \frac{\phi(\xi) - \mathrm{E}[\phi(\xi)]}{\mathrm{Var}[\phi(\xi)]}$$

- $h=0$ and $h=\xi$ are fixed points

rank-p RNNs converge to a neural field equation in R^p as $N \to \infty$

$$\dot h_i(t) = -h_i(t) + \sum_{j=1}^{N} J_{ij}\phi(h_j(t)) \qquad J_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_{\mu,i}\tilde\phi(\xi_{\mu,j})$$

N-dimensional system

- $h=0$ and $h=\xi$ are fixed points

$$\xi_{\mu,i} \sim \mathcal{N}(0,1), \quad \tilde\phi(\xi) = \frac{\phi(\xi) - \mathrm{E}[\phi(\xi)]}{\mathrm{Var}[\phi(\xi)]}$$

sum $\to$ int limit, h spans a p-dim subspace of R^N

$$\partial_t h(t,\vec z) = -h(t,\vec z) + \int_{\mathbb{R}^p} \underbrace{w(\vec z,\vec y)}_{J_{ij}}\phi(\underbrace{h(t,\vec y)}_{h_i(t)})\underbrace{\rho(\mathrm{d}\vec y)}_{\xi_i\sim\rho} \qquad w(\vec z,\vec y) = \sum_{\mu=1}^{p}\tilde\phi(y_\mu)z_\mu$$

p-dimensional system

numerical aspects of simulating fractal mappings

low-rank case

numerical aspects of simulating fractal mappings

low-rank case

"mean patterns" lead to low-rank dynamics in the mapping
→ inexpensive to simulate

$$\tilde{F}_{\mu,\alpha} = \frac{1}{|\alpha|}\sum_{i\in\alpha} \underbrace{F_{\mu,i}}_{=\,\xi_{\mu i}}, \quad \tilde{G}_{\mu,\alpha} = \frac{1}{|\alpha|}\sum_{i\in\alpha} \underbrace{G_{\mu,i}}_{=\,\phi(\xi_{\mu i})}$$

numerical aspects of simulating fractal mappings

low-rank case

"mean patterns" lead to low-rank dynamics in the mapping
→ inexpensive to simulate

$$\tilde{F}_{\mu,\alpha} = \frac{1}{|\alpha|}\sum_{i\in\alpha} \underbrace{F_{\mu,i}}_{=\,\xi_{\mu i}}, \quad \tilde{G}_{\mu,\alpha} = \frac{1}{|\alpha|}\sum_{i\in\alpha} \underbrace{G_{\mu,i}}_{=\,\phi(\xi_{\mu i})}$$

$$\tilde{J}_{\alpha,\beta} = \frac{|\beta|}{\sum_{\beta'}|\beta'|}\left(\sum_{\mu=1}^{p}\tilde{F}_{\mu,\alpha}\tilde{G}_{\mu,\beta} - \delta_{\alpha,\beta}\underbrace{\sum_{\mu=1}^{p}\sum_{i\in\alpha}\frac{F_{\mu,i}}{|\alpha|}\frac{G_{\mu,i}}{|\alpha|}}_{\gamma_\alpha}\right)$$

weighted binning

low-rank term

exclude self-connections (vanishes at large N)

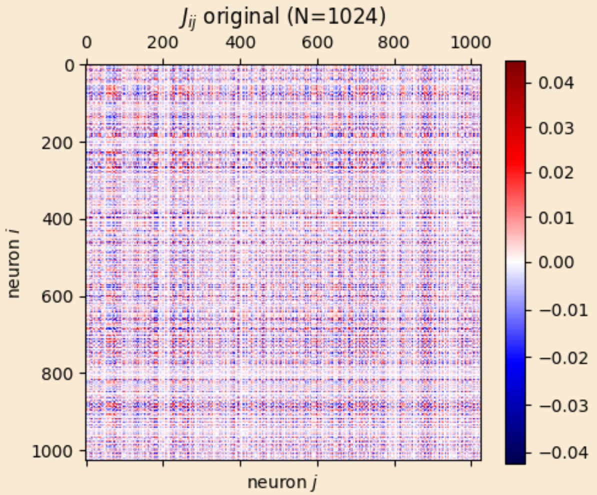
numerical aspects of simulating fractal mappings

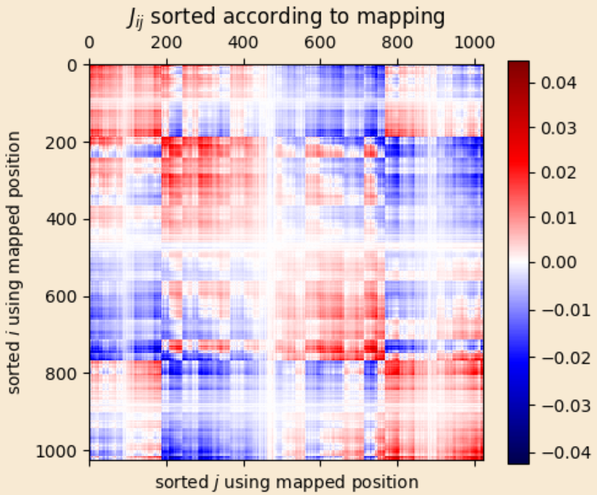
low-rank case

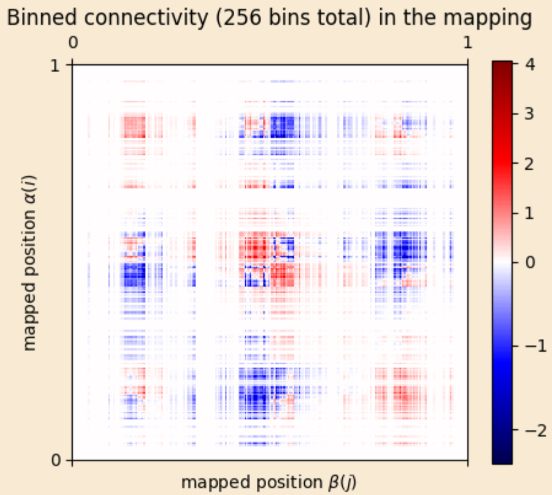
"mean patterns" lead to low-rank dynamics in the mapping
→ inexpensive to simulate

$$\tilde{F}_{\mu,\alpha} = \frac{1}{|\alpha|}\sum_{i\in\alpha} \underbrace{F_{\mu,i}}_{= \xi_{\mu i}},\quad \tilde{G}_{\mu,\alpha} = \frac{1}{|\alpha|}\sum_{i\in\alpha} \underbrace{G_{\mu,i}}_{= \phi(\xi_{\mu i})}$$

$$\tilde{J}_{\alpha,\beta} = \underbrace{\frac{|\beta|}{\sum_{\beta'}|\beta'|}}_{\text{weighted binning}}\left(\underbrace{\sum_{\mu=1}^{p}\tilde{F}_{\mu,\alpha}\tilde{G}_{\mu,\beta}}_{\text{low-rank term}} - \underbrace{\delta_{\alpha,\beta}\underbrace{\sum_{\mu=1}^{p}\sum_{i\in\alpha}\frac{F_{\mu,i}}{|\alpha|}\frac{G_{\mu,i}}{|\alpha|}}_{\gamma_\alpha}}_{\text{exclude self-connections (vanishes at large N)}}\right)$$







< numerical demo of binning the connectivity
N=1024, nrec=4

numerical aspects of simulating fractal mappings

low-rank case

"mean patterns" lead to low-rank dynamics in the mapping
→ inexpensive to simulate

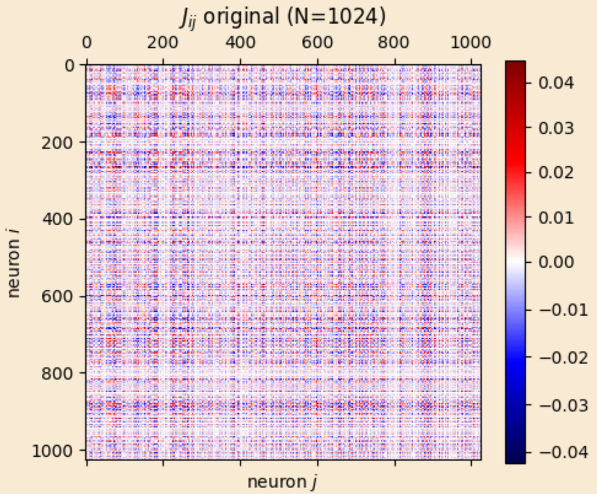
$$\tilde{F}_{\mu,\alpha} = \frac{1}{|\alpha|}\sum_{i\in\alpha} \underbrace{F_{\mu,i}}_{=\,\xi_{\mu i}}, \quad \tilde{G}_{\mu,\alpha} = \frac{1}{|\alpha|}\sum_{i\in\alpha} \underbrace{G_{\mu,i}}_{=\,\phi(\xi_{\mu i})}$$

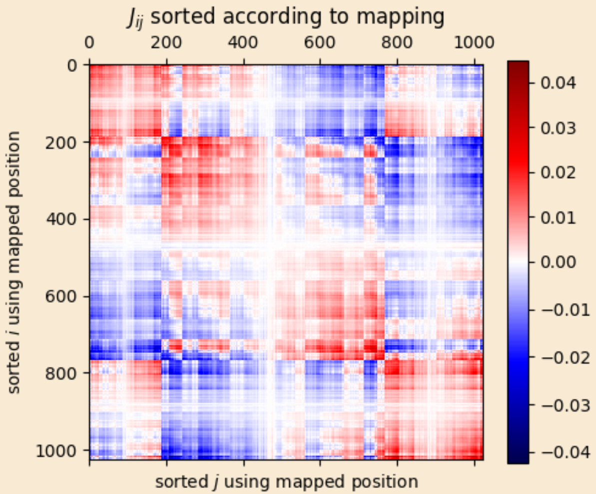
$$\tilde{J}_{\alpha,\beta} = \frac{|\beta|}{\sum_{\beta'}|\beta'|}\left(\sum_{\mu=1}^{p}\tilde{F}_{\mu,\alpha}\tilde{G}_{\mu,\beta} - \delta_{\alpha,\beta}\underbrace{\sum_{\mu=1}^{p}\sum_{i\in\alpha}\frac{F_{\mu,i}}{|\alpha|}\frac{G_{\mu,i}}{|\alpha|}}_{\gamma_\alpha}\right)$$

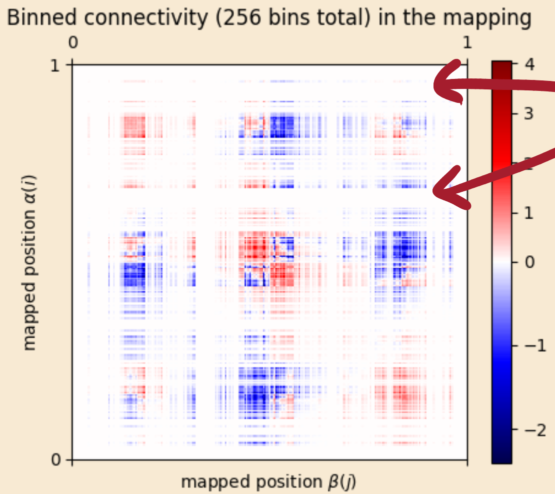
weighted binning

low-rank term

exclude self-connections (vanishes at large N)





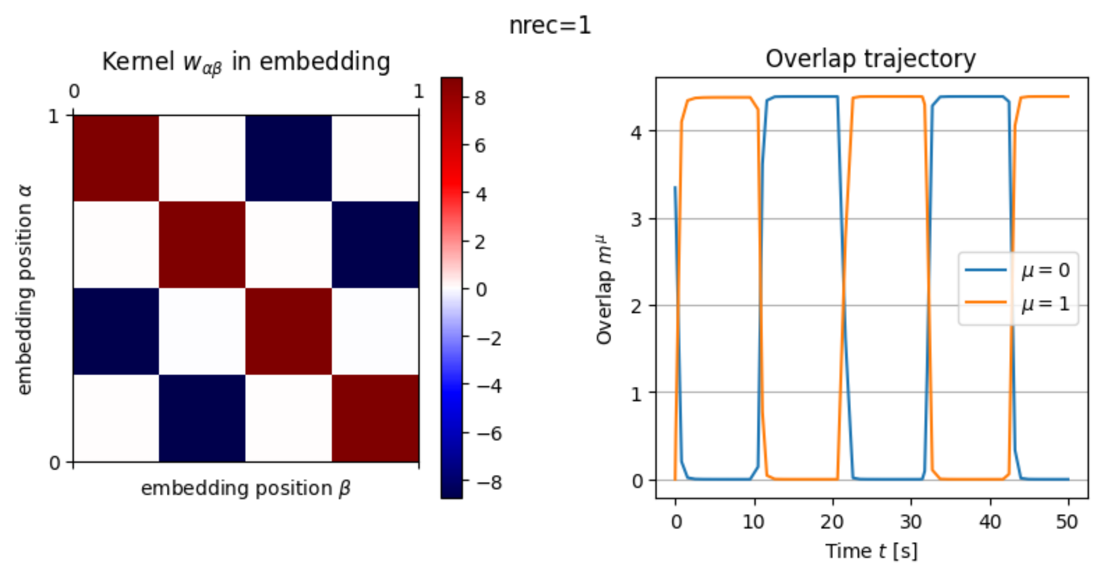


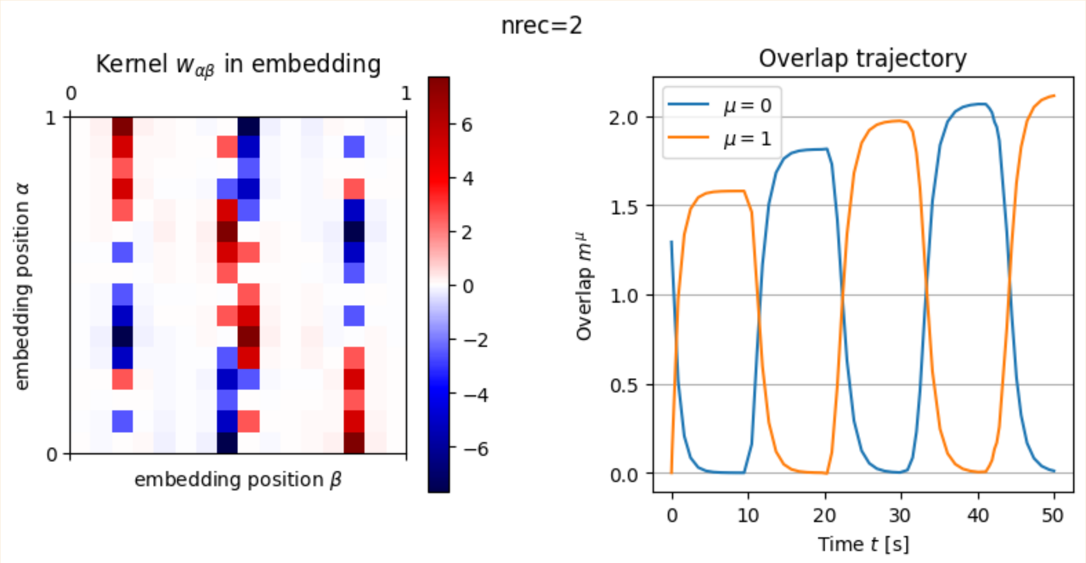
rare neurons might not show up !

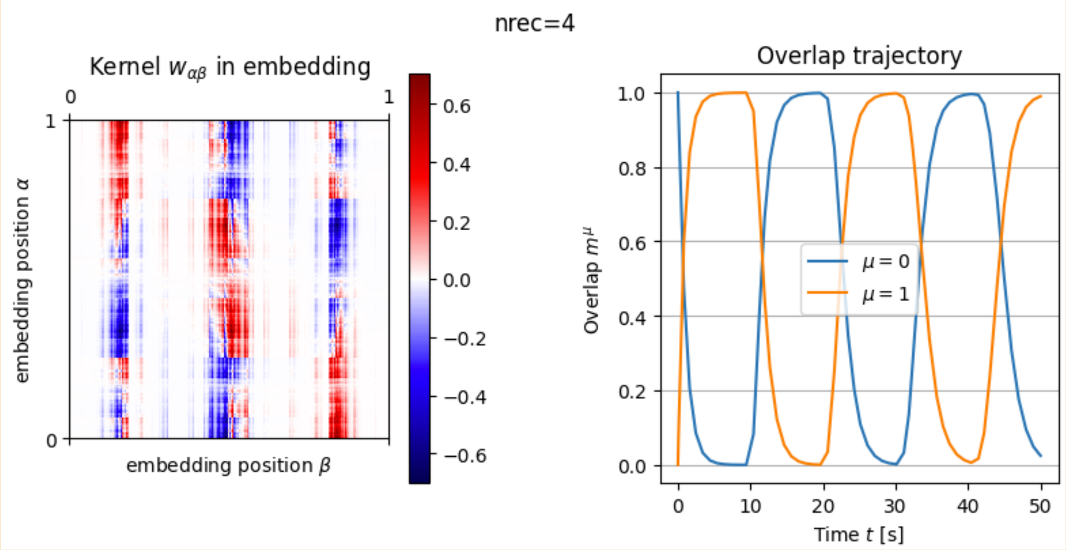
< numerical demo of binning the connectivity N=1024, nrec=4

dynamics inside the fractal mapping converge

dynamics inside the fractal mapping converge







→ we get the same oscillatory behavior !

Note : this is expected, as we are just doing a mean field, and the mappings are just reorderings of the bins. Connectivity is invariant to permutation of neurons

(if time remains)
discrete probability spaces
reduce to population dynamics
which can easily be embedded in [0,1]