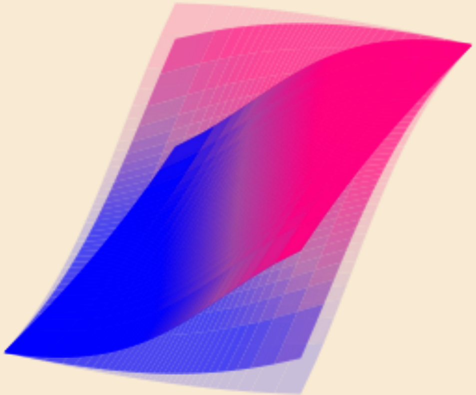
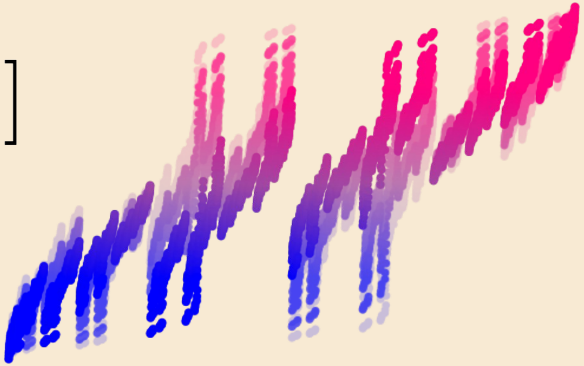
Equivalence of neural field dynamics with different embedding dimensionality

Nicole Vadot
(Supervisor)
Dr. Valentin Schmutz
(Supervisor, LCN director)
Prof. Wulfram Gerstner
(External expert)
Prof. Romain Veltz



$$S : \mathbb{R}^p \mapsto [0, 1]$$

$$\Longleftrightarrow$$



Neural field in $\mathbb{R}^p$

Neural field in $[0, 1]$

Online: https://ninivert.github.io/lcnthesis/

recap: neural fields & embedding spaces

activation (firing rate)

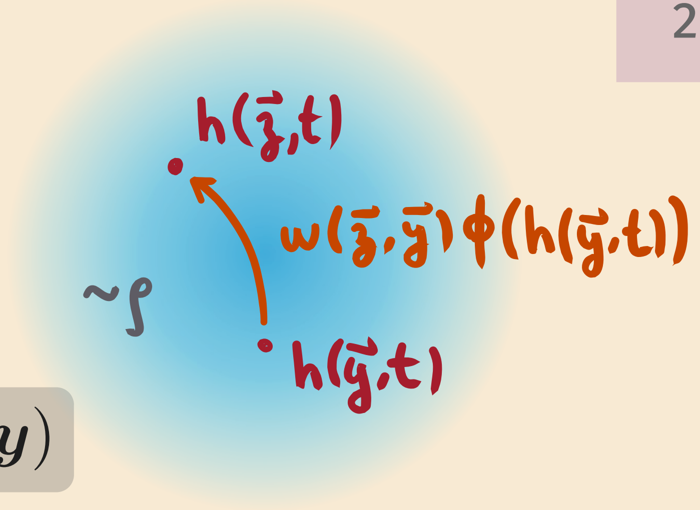
$$\partial_t h(\boldsymbol{z},t) = -h(\boldsymbol{z},t) + \int_{\mathbb{R}^p} w(\boldsymbol{z},\boldsymbol{y})\phi(h(\boldsymbol{y},t))\rho(\mathrm{d}\boldsymbol{y})$$

neural field

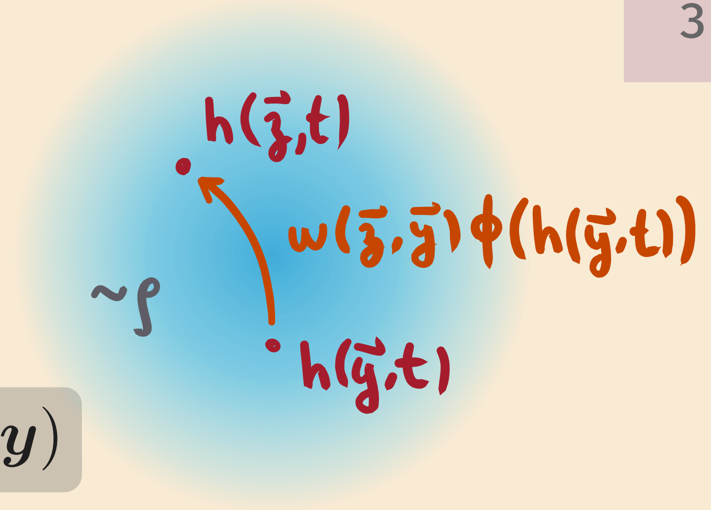
embedding space

connectivity kernel

population density



recap: neural fields & embedding spaces



activation (firing rate)

$$\partial_t h(\boldsymbol{z},t) = -h(\boldsymbol{z},t) + \int_{\mathbb{R}^p} w(\boldsymbol{z},\boldsymbol{y})\phi(h(\boldsymbol{y},t))\rho(\mathrm{d}\boldsymbol{y})$$

neural field

embedding space

connectivity kernel

population density

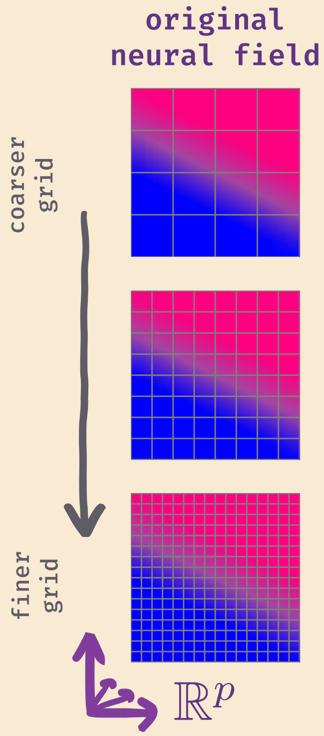
by redefining, we can "hide" the density in the kernel:
⇒ we get a **neural field in [0,1]^p**

$$\partial_t h_U(\boldsymbol{v},t) = -h_U(\boldsymbol{v},t) + \int_{[0,1]^p} w_U(\boldsymbol{v},\boldsymbol{u})\phi(h_U(\boldsymbol{u},t))\mathrm{d}\boldsymbol{u}$$

problem statement

neural field in p dimensions

$$\partial_t h_U(t,\boldsymbol{v}) = -h_U(t,\boldsymbol{v}) + \int_{[0,1]^p} w_U(\boldsymbol{v},\boldsymbol{u})\phi(h_U(t,\boldsymbol{u}))\mathrm{d}\boldsymbol{u}$$



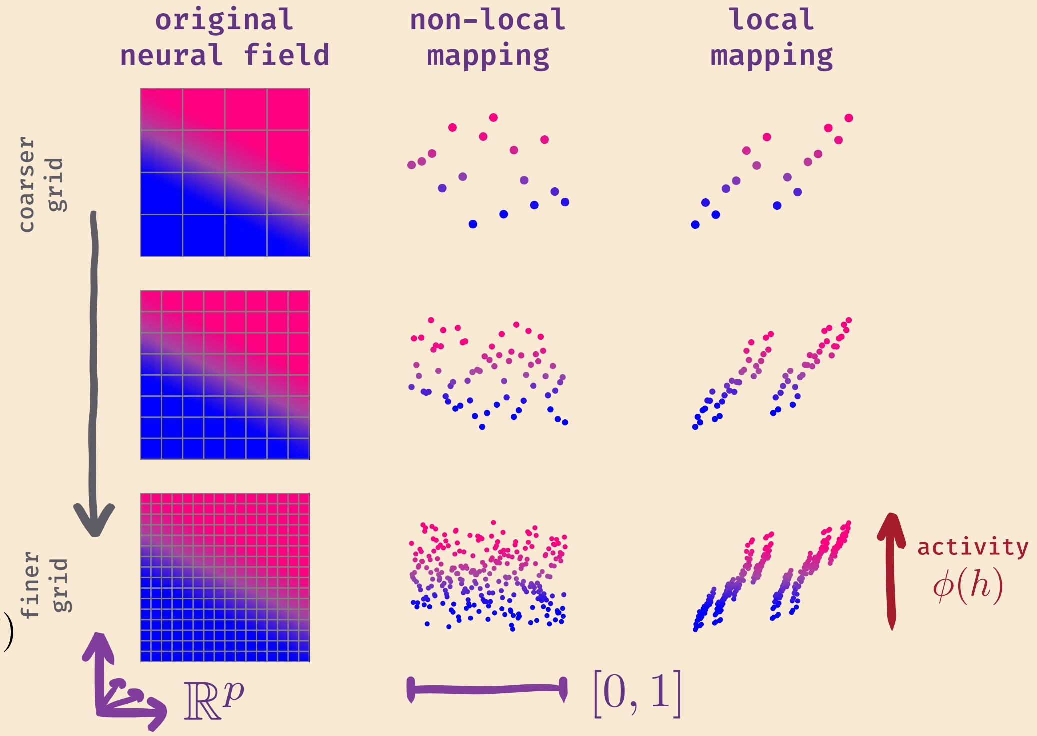
problem statement

neural field in p dimensions

$$\partial_t h_U(t,\boldsymbol{v}) = -h_U(t,\boldsymbol{v}) + \int_{[0,1]^p} w_U(\boldsymbol{v},\boldsymbol{u})\phi(h_U(t,\boldsymbol{u}))\mathrm{d}\boldsymbol{u}$$

map to neural field in one dimension?

$$\partial_t \tilde{h}(\alpha,t) = -\tilde{h}(\alpha,t) + \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda \circ S^{-1}\right](\mathrm{d}\beta)$$



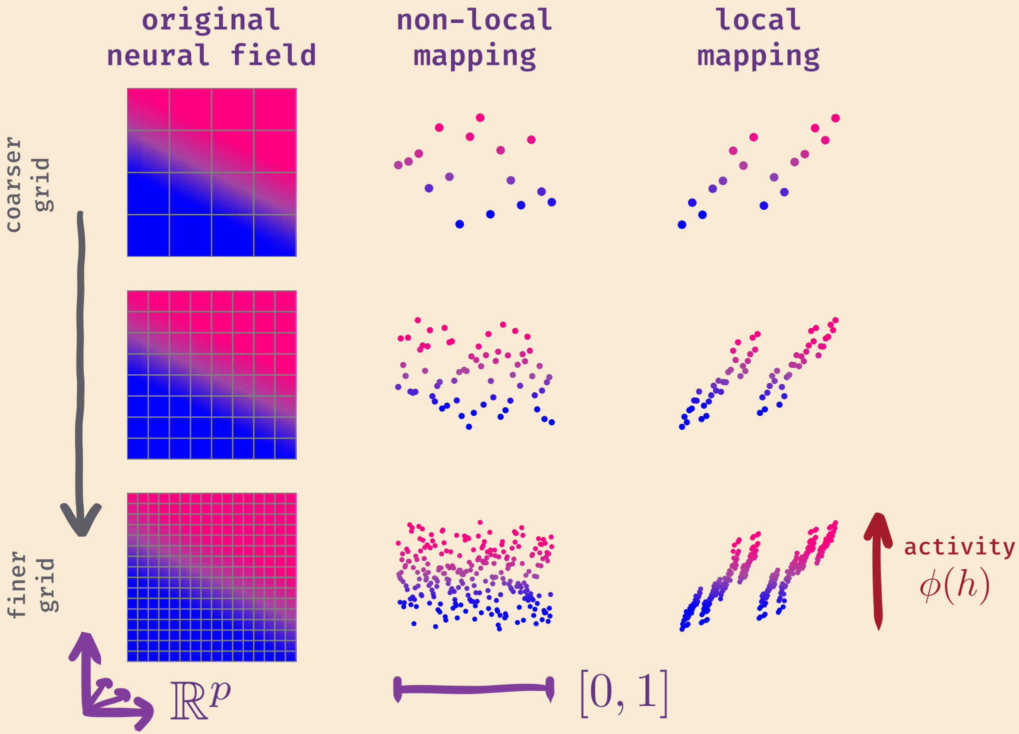
problem statement

neural field in p dimensions

$$\partial_t h_U(t,\boldsymbol{v}) = -h_U(t,\boldsymbol{v}) + \int_{[0,1]^p} w_U(\boldsymbol{v},\boldsymbol{u})\phi(h_U(t,\boldsymbol{u}))\mathrm{d}\boldsymbol{u}$$

map to neural field in one dimension?

$$\partial_t \tilde{h}(\alpha,t) = -\tilde{h}(\alpha,t) + \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda \circ \boxed{S^{-1}}\right](\mathrm{d}\beta)$$



- → "regularity" of the activity is what allows writing the neural field
- → can an equivalent neural field in [0,1] also be regular?
- → what conditions on mapping S to preserve regularity? (→ locality)

smoothness of the kernel: what properties on S^{-1} can we hope for?

$$\partial_t \tilde{h}(\alpha,t) = -\tilde{h}(\alpha,t) + \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t)) \left[\lambda \circ S^{-1}\right](\mathrm{d}\beta)$$

kernel of the [0,1] embedding
defined as: $\tilde{w}(\alpha,\beta) = w_U(S^{-1}(\alpha), S^{-1}(\beta))$

→ w_U is highly regular, but
$\tilde{w}$ might not be, it depends on S^{-1}

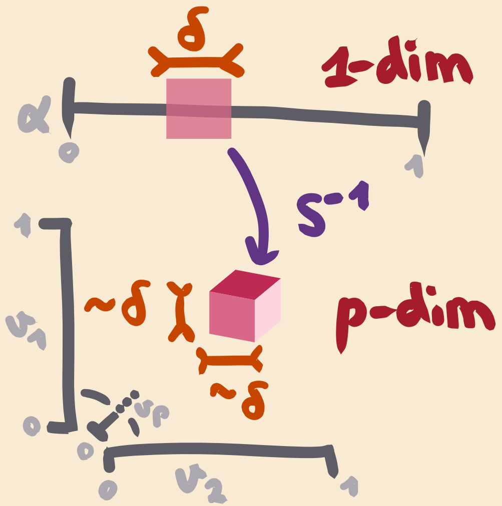
smoothness of the kernel: what properties on S^{-1} can we hope for?

$$\partial_t \tilde{h}(\alpha,t) = -\tilde{h}(\alpha,t) + \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t)) \left[\lambda \circ S^{-1}\right](\mathrm{d}\beta)$$

kernel of the [0,1] embedding
defined as: $\tilde{w}(\alpha,\beta) = w_U(S^{-1}(\alpha), S^{-1}(\beta))$

→ w_U is highly regular, but
$\tilde{w}$ might not be, it depends on S^{-1}

→ small neighbourhoods in [0,1] should be mapped to small neighbourhoods in $[0,1]^p$ →
→ this is tricky



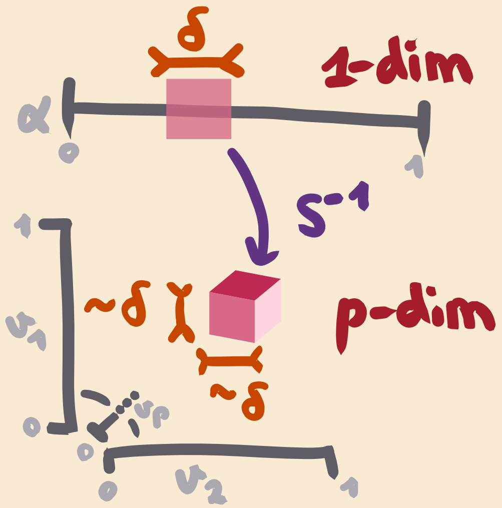
smoothness of the kernel: what properties on S^{-1} can we hope for?

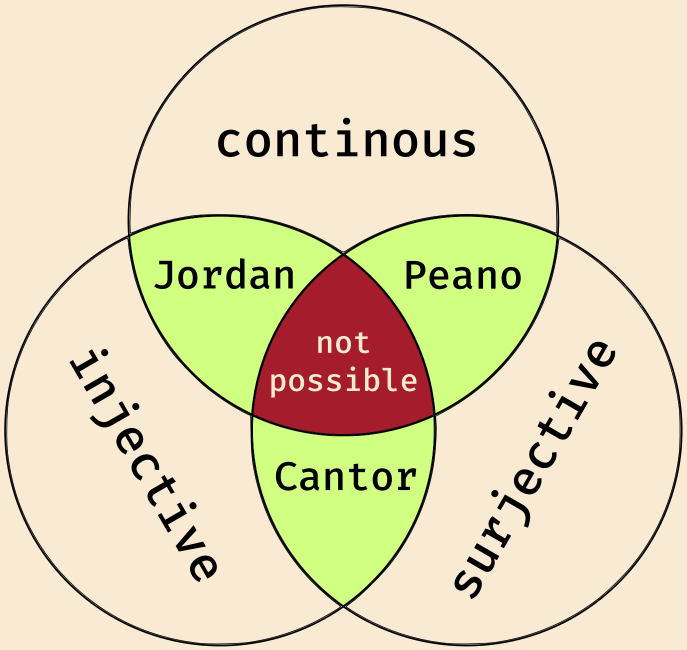
$$\partial_t \tilde{h}(\alpha,t) = -\tilde{h}(\alpha,t) + \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t)) \left[\lambda \circ S^{-1}\right](\mathrm{d}\beta)$$

kernel of the [0,1] embedding
defined as: $\tilde{w}(\alpha,\beta) = w_U(S^{-1}(\alpha), S^{-1}(\beta))$

→ w_U is highly regular, but
$\tilde{w}$ might not be, it depends on S^{-1}

→ small neighbourhoods in [0,1] should be mapped to small neighbourhoods in $[0,1]^p$ →
→ this is tricky





↑ Netto's theorem states there are no continuous bijections $S^{-1}: [0,1] \mapsto [0,1]^2$

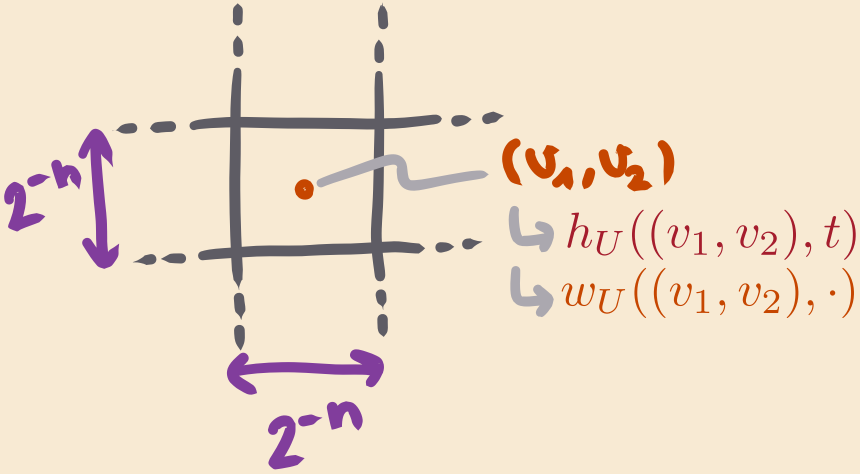
S as the limit of a sequence of mappings S^n – motivation

why? → **grid discretization of the neural field in p dimensions**

→ 2^n bins along each dimension

→ case p=2, positions of bins given by: $\boldsymbol{v_i} = (v_{1,i}, v_{2,i}),\ i \in \{1, \cdots 4^n\}$

↓ each bin "samples" the neural field and kernel



S as the limit of a sequence of mappings S^n - motivation

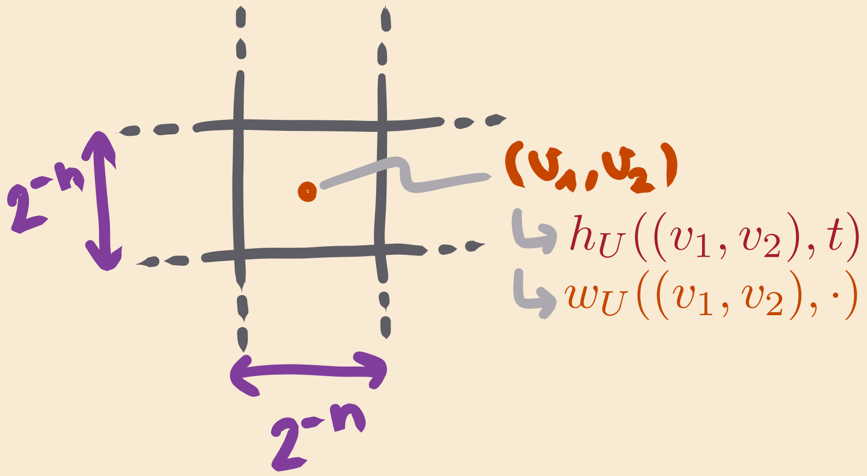
why? → **grid discretization of the neural field in p dimensions**

→ 2^n bins along each dimension

→ case p=2, positions of bins given by: $\boldsymbol{v_i} = (v_{1,i}, v_{2,i}),\ i \in \{1, \cdots 4^n\}$

→ numerical kernel: $J_{ij} = w_U(\boldsymbol{v_i}, \boldsymbol{v_j})$

↓ each bin "samples" the neural field and kernel



S as the limit of a sequence of mappings S^n – motivation

why? → **grid discretization of the neural field in p dimensions**

→ 2^n bins along each dimension

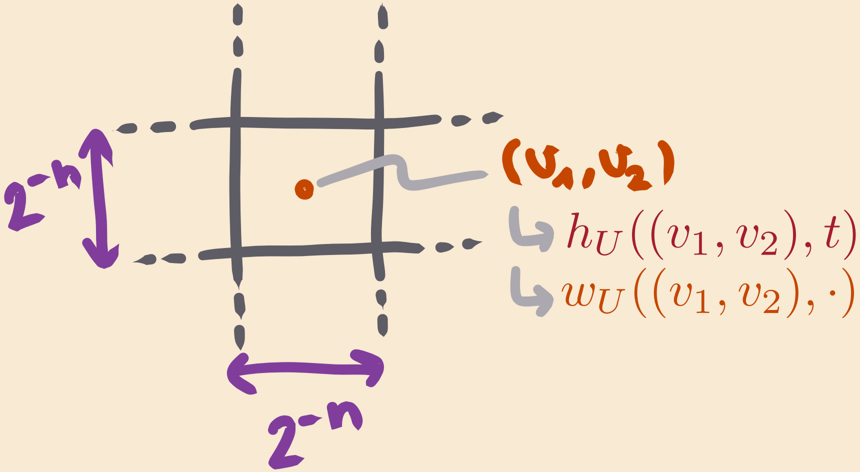
→ case p=2, positions of bins given by: $\boldsymbol{v_i} = (v_{1,i}, v_{2,i}),\ i \in \{1, \cdots 4^n\}$

→ numerical kernel: $J_{ij} = w_U(\boldsymbol{v_i}, \boldsymbol{v_j})$

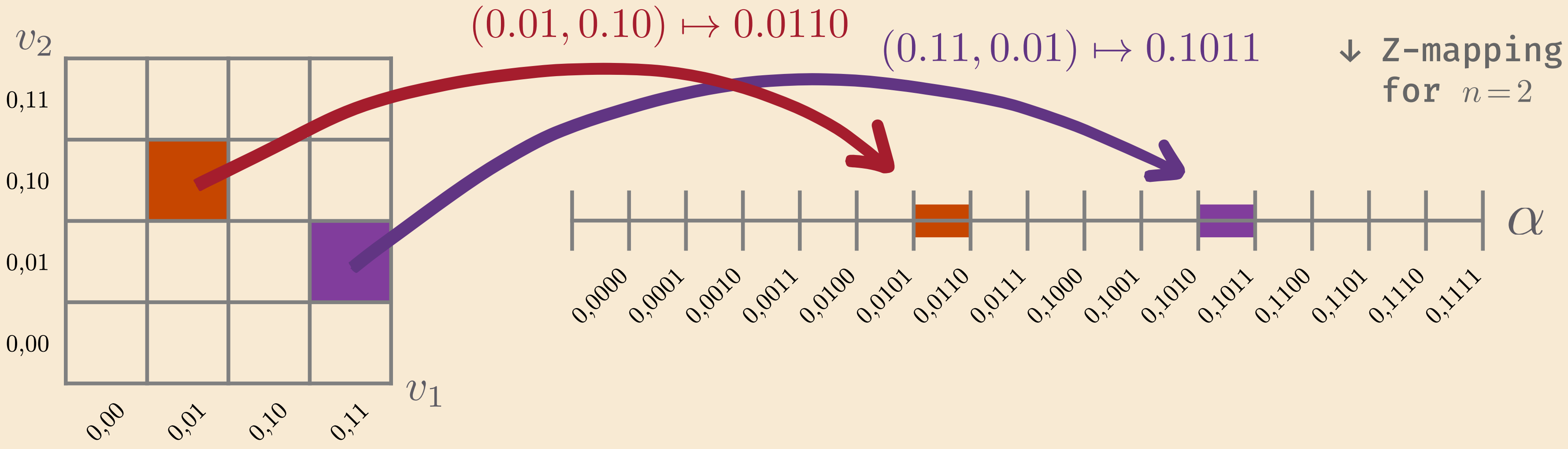
→ **we simulate the discretized neural field**

$$\frac{\mathrm{d}}{\mathrm{d}t} h_i(t) = -h_i(t) + \sum_{j=1}^{4^n} J_{ij}\phi(h_j(t))$$

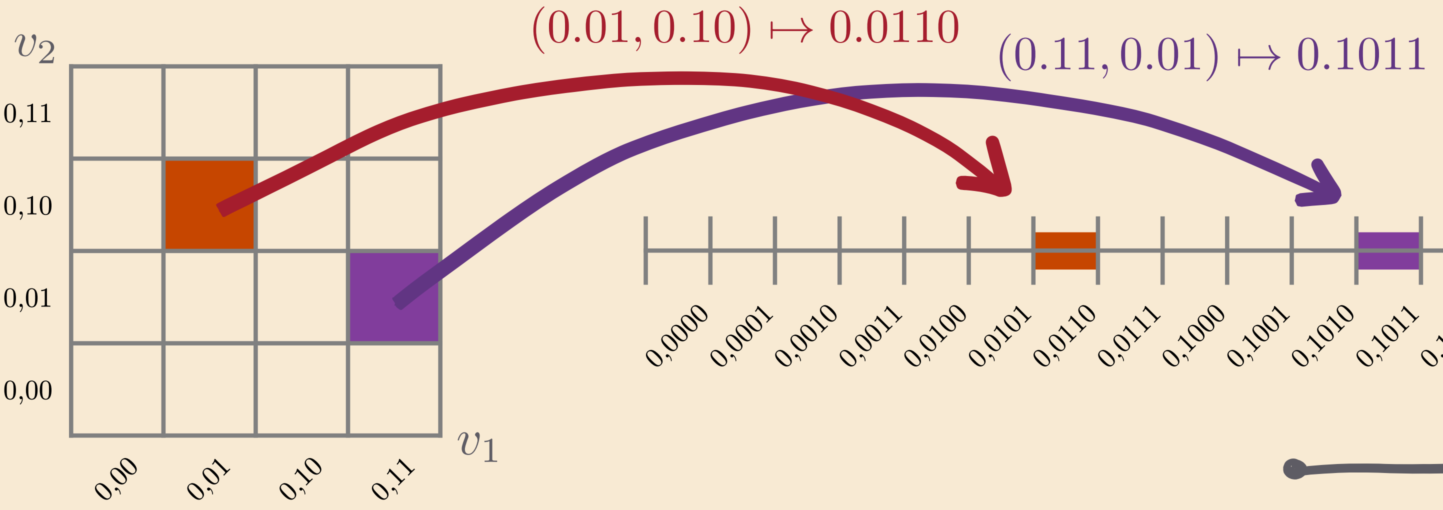
↓ each bin "samples" the neural field and kernel



$\mathcal{S}$ as the limit of a sequence of mappings $\mathcal{S}^n$ – definition



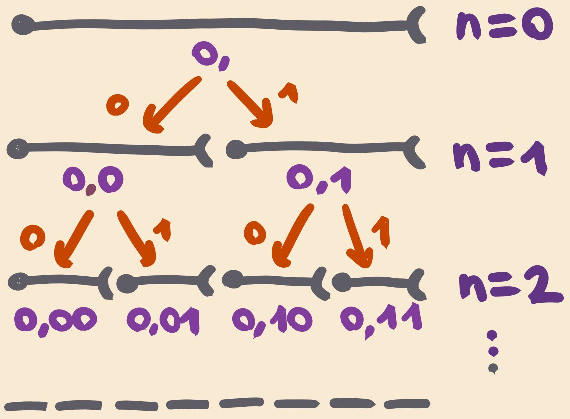
S as the limit of a sequence of mappings S^n – definition



↓ Z-mapping for $n=2$

→ S^n maps 4ⁿ squares of size $2^{-n} \times 2^{-n}$ to 4ⁿ segments of size 4^{-n}

→ bins are located using binary expansions →

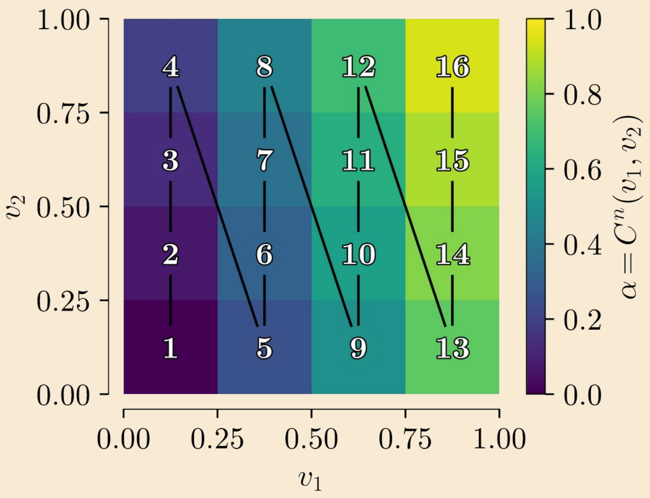


$$S^n : (v_1^{(n)}, v_2^{(n)}) = (0.b_1^1 b_2^1 \cdots b_n^1, 0.b_1^2 b_2^2 \cdots b_n^2)$$
$$\mapsto \alpha^{(n)} = 0.b_1 b_2 \cdots b_{2n}$$

a naive mapping: Column mapping

→ enumerate the populations columnwise

Column mapping for $n=2$ →

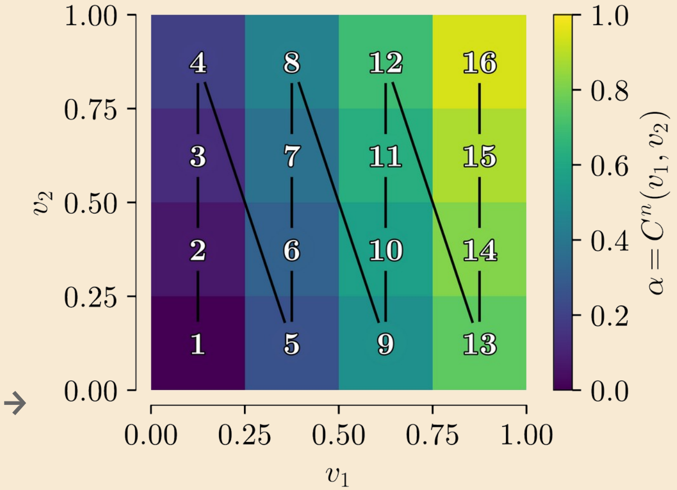


a naive mapping: Column mapping

→ enumerate the populations columnwise

$$\alpha^{(n)} = C(v_1^{(n)}, v_2^{(n)}) = 0.b_1^1 b_2^1 \cdots b_n^1 b_1^2 b_2^2 \cdots b_n^2$$
$$= \sum_{k=1}^{n} b_k^1 2^{-k} + b_k^2 2^{-(n+k)}$$

Column mapping for $n=2$ →

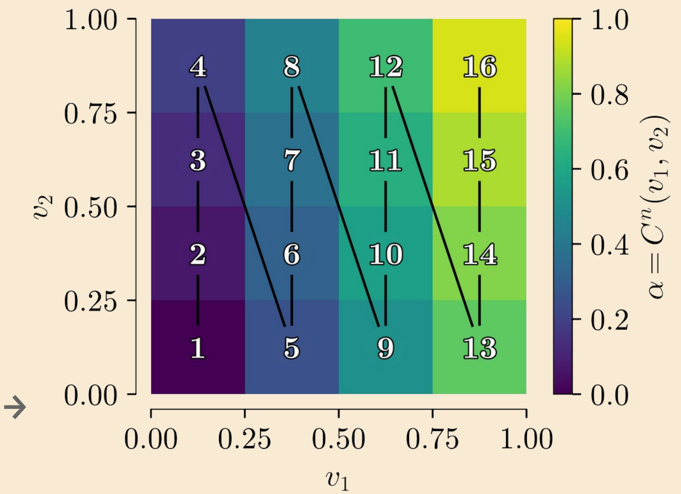


a naive mapping: Column mapping

→ enumerate the populations columnwise

$$\alpha^{(n)} = C(v_1^{(n)}, v_2^{(n)}) = 0.b_1^1 b_2^1 \cdots b_n^1 b_1^2 b_2^2 \cdots b_n^2$$
$$= \sum_{k=1}^{n} b_k^1 2^{-k} + b_k^2 2^{-(n+k)}$$

Column mapping for $n=2$ →



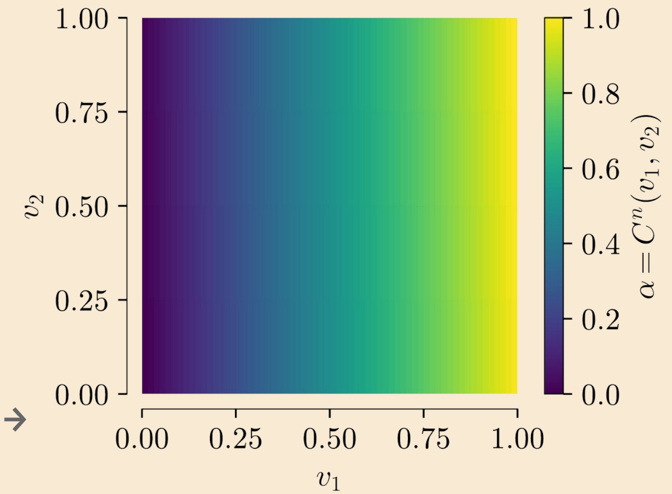
→ at finite-n, inverse is

$$v_1^{(n)} = C_1^{-1}(\alpha^{(n)}) = 0.b_1 b_2 \cdots b_n = \sum_{k=1}^{n} b_k 2^{-k}$$

$$v_2^{(n)} = C_2^{-1}(\alpha^{(n)}) = 0.b_{n+1} b_{n+2} \cdots b_{2n} = \sum_{k=1}^{n} b_{n+k} 2^{-k}$$

→ converges to a projection on the first axis

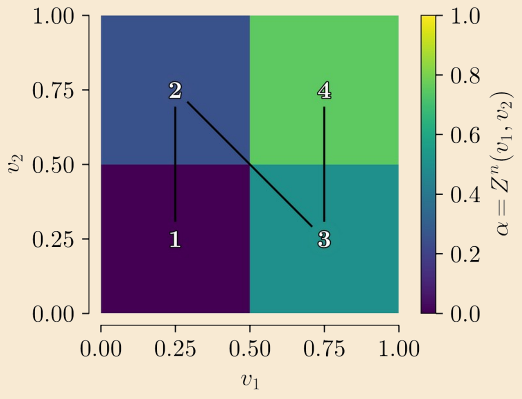
Column mapping for $n \to \infty$ →



a less naive mapping: Z-mapping

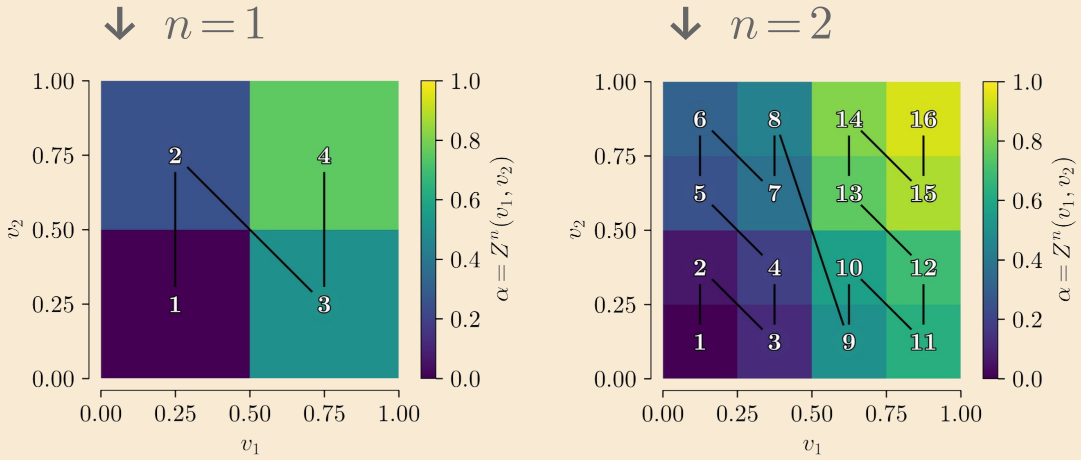
→ (recursively) draw Z shapes

↓ $n = 1$



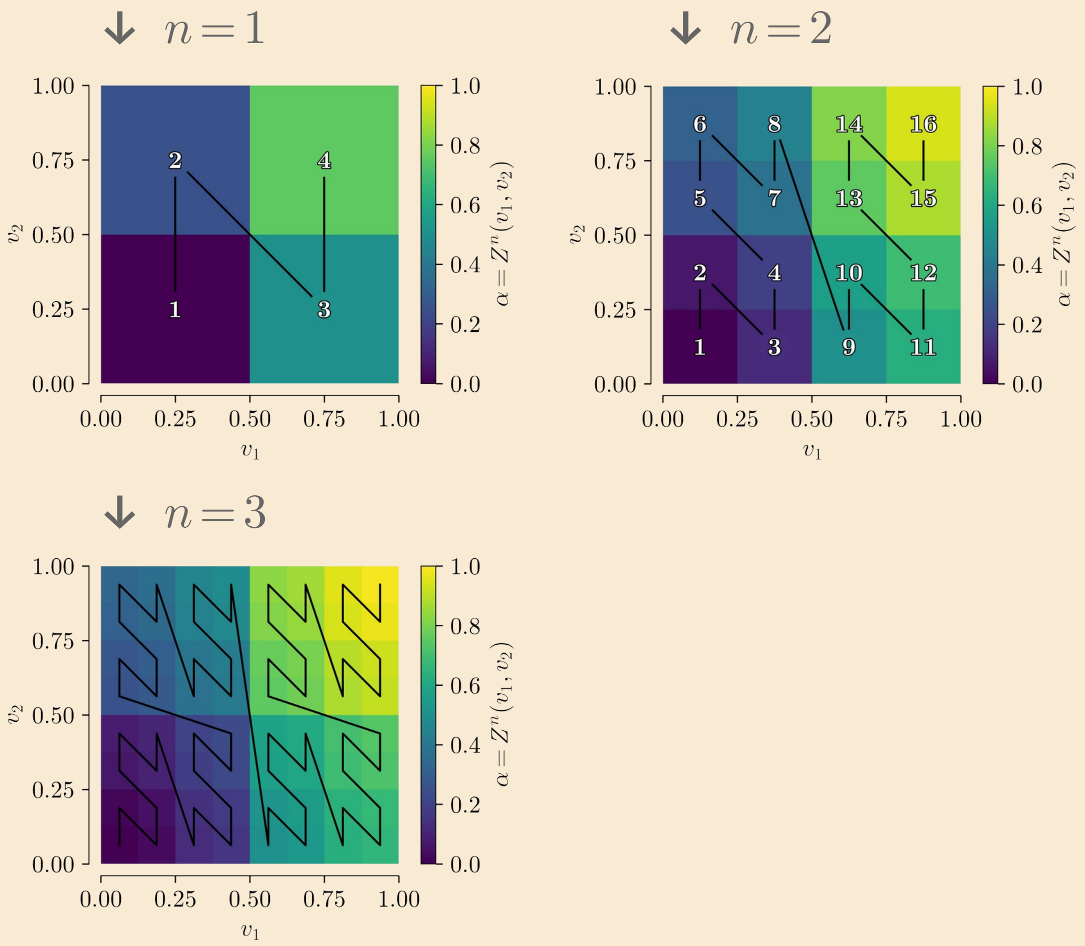
a less naive mapping: Z-mapping

- (recursively) draw Z shapes



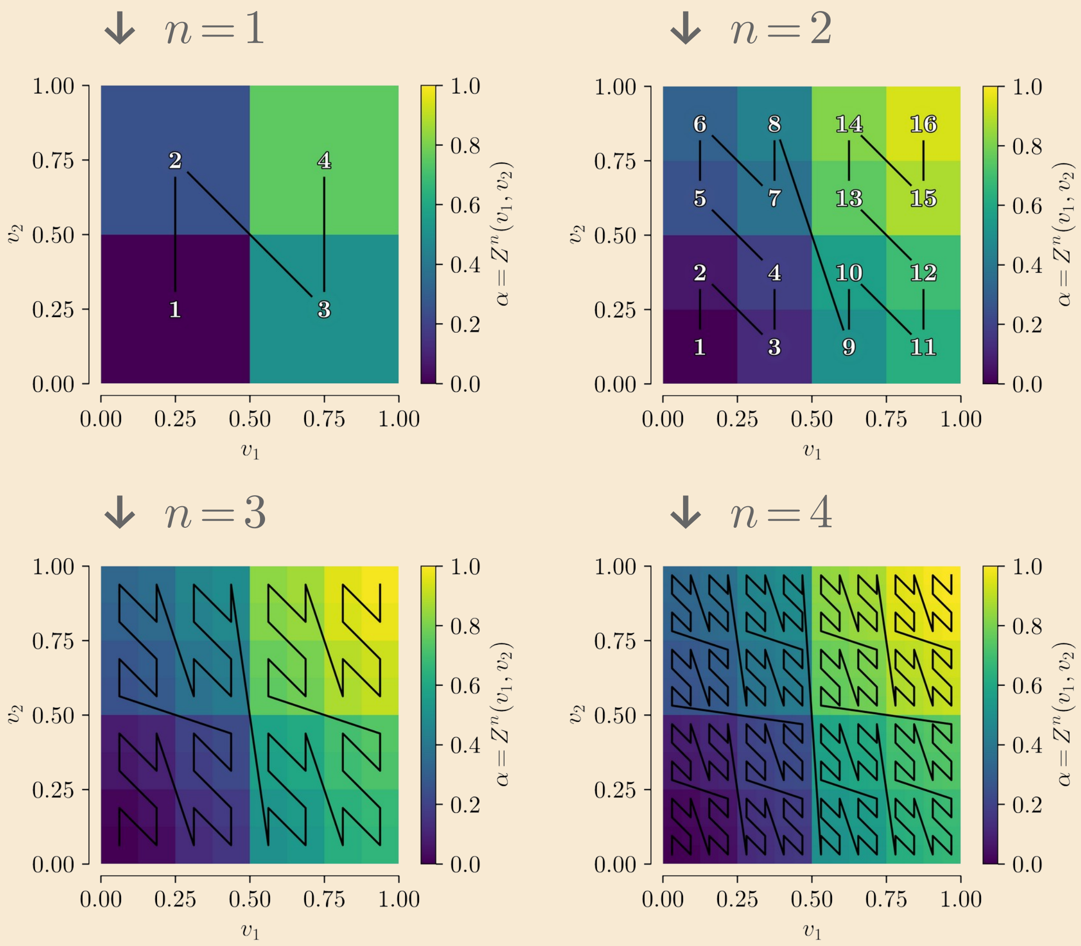
a less naive mapping: Z-mapping

- (recursively) draw Z shapes



a less naive mapping: Z-mapping

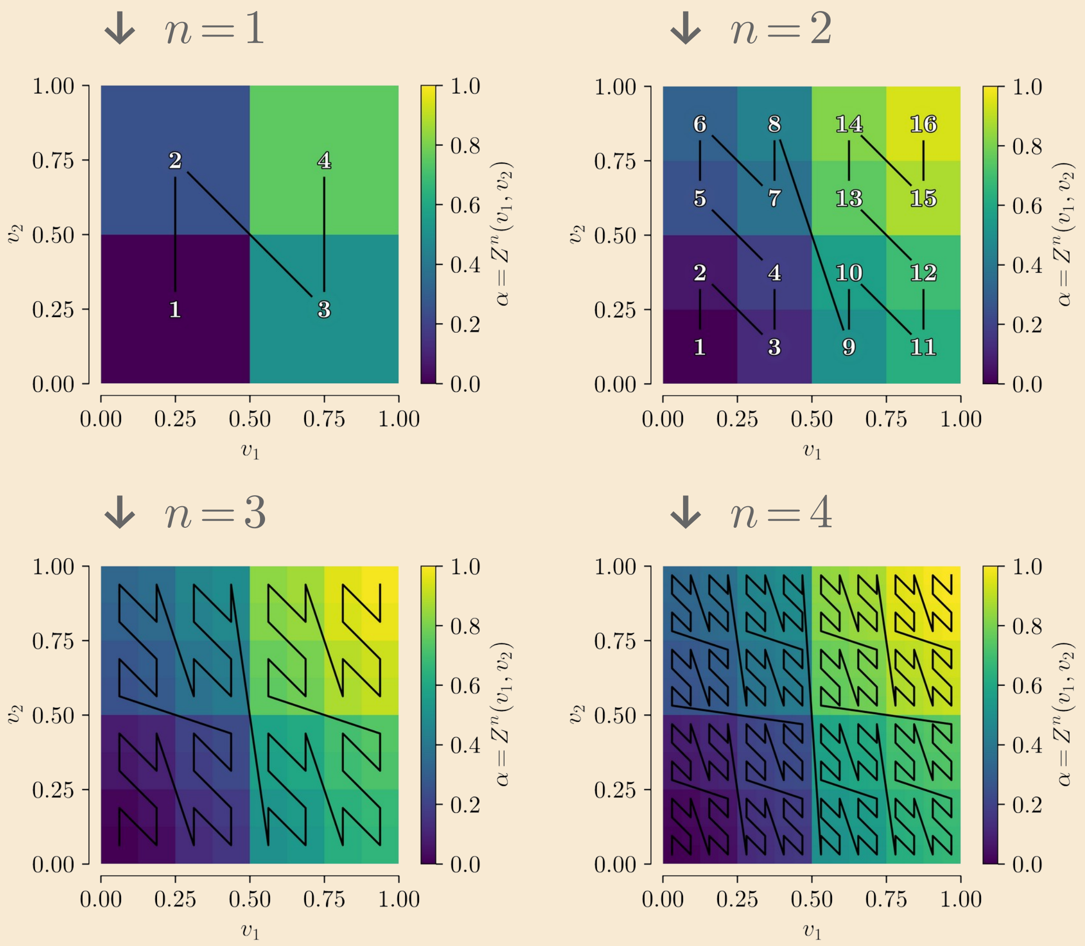
- (recursively) draw Z shapes



a less naive mapping: Z-mapping

- (recursively) draw Z shapes

$$\alpha^{(n)} = Z(v_1^{(n)}, v_2^{(n)})$$
$$= 0.b_1^1 b_1^2 b_2^1 b_2^2 \cdots b_n^1 b_n^2$$



a less naive mapping: Z-mapping

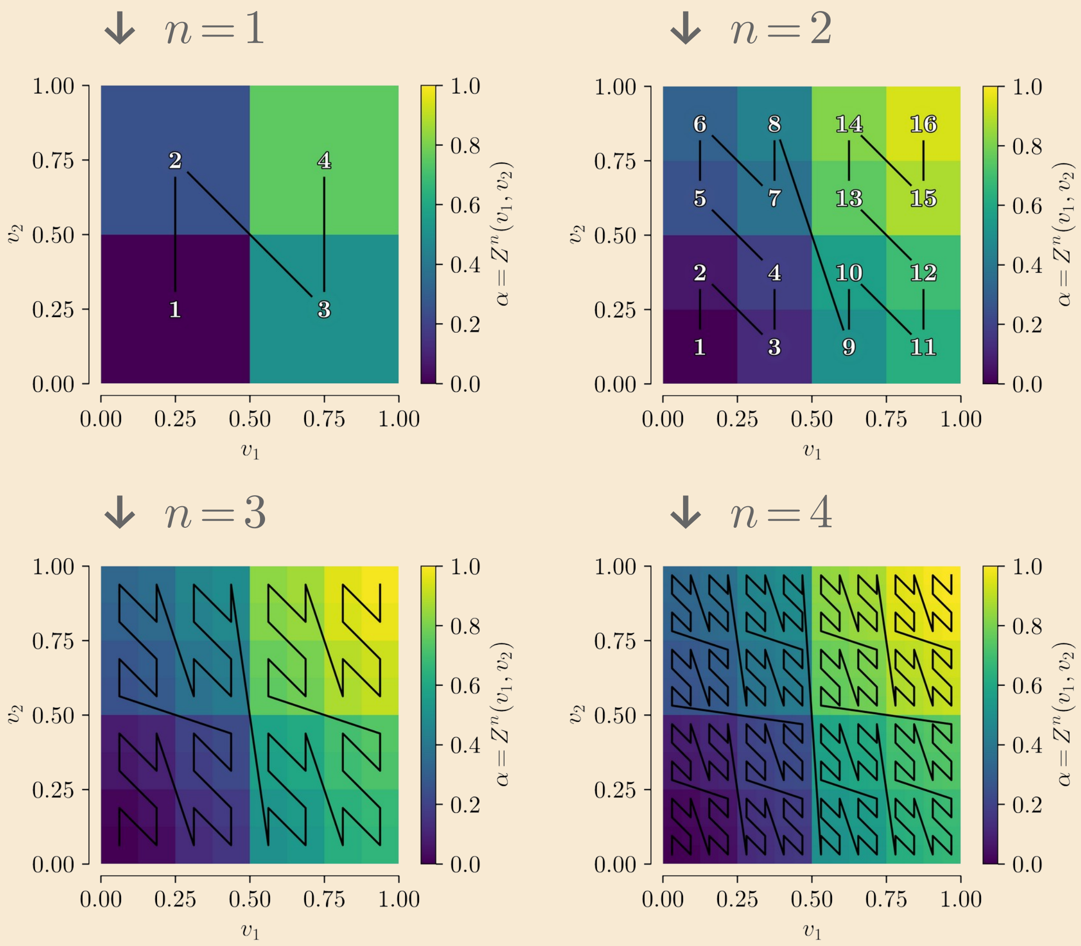
- (recursively) draw Z shapes

$$\alpha^{(n)} = Z(v_1^{(n)}, v_2^{(n)}) = 0.b_1^1 b_1^2 b_2^1 b_2^2 \cdots b_n^1 b_n^2$$

- again, inverse is

$$v_1^{(n)} = Z_1^{-1}(\alpha^{(n)}) = 0.b_1 b_3 \cdots b_{2n-1}$$
$$v_2^{(n)} = Z_2^{-1}(\alpha^{(n)}) = 0.b_2 b_4 \cdots b_{2n}$$

- converges to a bijection between the segment and the square



coarse-graining: how to simulate the 1-dimensional neural field

→ naive numerical kernel for [0,1]

$$\begin{aligned}\tilde{w}(\alpha_i, \beta_j) &= \tilde{J}_{\alpha_i, \beta_j} \\ &= \tilde{J}_{S(i), S(j)} \\ &\overset{\text{def}}{=} J_{ij} \\ &= w_U(\boldsymbol{v_i}, \boldsymbol{v_j})\end{aligned}$$

coarse-graining: how to simulate the 1-dimensional neural field

- naive numerical kernel for [0,1]

$$\begin{aligned} \tilde{w}(\alpha_i, \beta_j) &= \tilde{J}_{\alpha_i,\beta_j} \\ &= \tilde{J}_{S(i),S(j)} \\ &\stackrel{\text{def}}{=} J_{ij} \\ &= w_U(\boldsymbol{v_i}, \boldsymbol{v_j}) \end{aligned}$$

- problem: **permutation invariance**
  ⇒ identical dynamics indep. of the mapping!

coarse-graining: how to simulate the 1-dimensional neural field

→ naive numerical kernel for [0,1]

$$\begin{aligned}\tilde{w}(\alpha_i, \beta_j) &= \tilde{J}_{\alpha_i,\beta_j} \\ &= \tilde{J}_{S(i),S(j)} \\ &\stackrel{\text{def}}{=} J_{ij} \\ &= w_U(\boldsymbol{v_i}, \boldsymbol{v_j})\end{aligned}$$

→ problem: **permutation invariance**
⇒ identical dynamics indep. of the mapping!

→ in 2D: 2^n bins per dimension
in 1D: 4^n bins per dimension

coarse-graining: how to simulate the 1-dimensional neural field

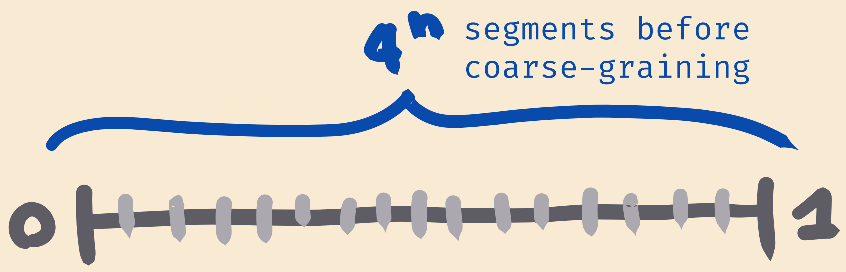
→ naive numerical kernel for [0,1]

$$\begin{aligned}\tilde{w}(\alpha_i, \beta_j) &= \tilde{J}_{\alpha_i, \beta_j} \\ &= \tilde{J}_{S(i), S(j)} \\ &\stackrel{\text{def}}{=} J_{ij} \\ &= w_U(\boldsymbol{v_i}, \boldsymbol{v_j})\end{aligned}$$

→ problem: **permutation invariance**
⇒ identical dynamics indep. of the mapping!

→ in 2D: 2^n bins per dimension
in 1D: 4^n bins per dimension

→ reduce to 2^n bins in 1D by **coarse-graining**



↑ coarse-graining averages 2^n consecutive populations in 1D, forming effective bins of size 2^{-n}

coarse-graining: how to simulate the 1-dimensional neural field

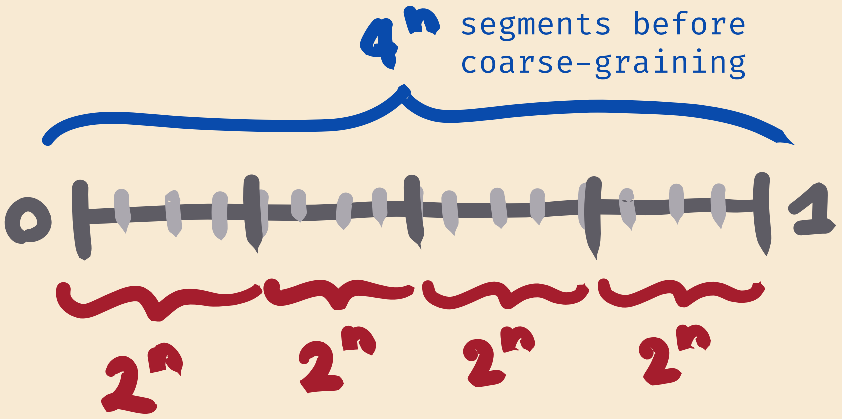
→ naive numerical kernel for [0,1]

$$\begin{aligned}\tilde{w}(\alpha_i, \beta_j) &= \tilde{J}_{\alpha_i, \beta_j} \\ &= \tilde{J}_{S(i), S(j)} \\ &\overset{\text{def}}{=} J_{ij} \\ &= w_U(\boldsymbol{v_i}, \boldsymbol{v_j})\end{aligned}$$

→ problem: **permutation invariance** ⇒ identical dynamics indep. of the mapping!

→ in 2D: 2^n bins per dimension
in 1D: 4^n bins per dimension

→ reduce to 2^n bins in 1D by **coarse-graining**



↑ coarse-graining averages 2^n consecutive populations in 1D, forming effective bins of size 2^{-n}

coarse-graining: how to simulate the 1-dimensional neural field

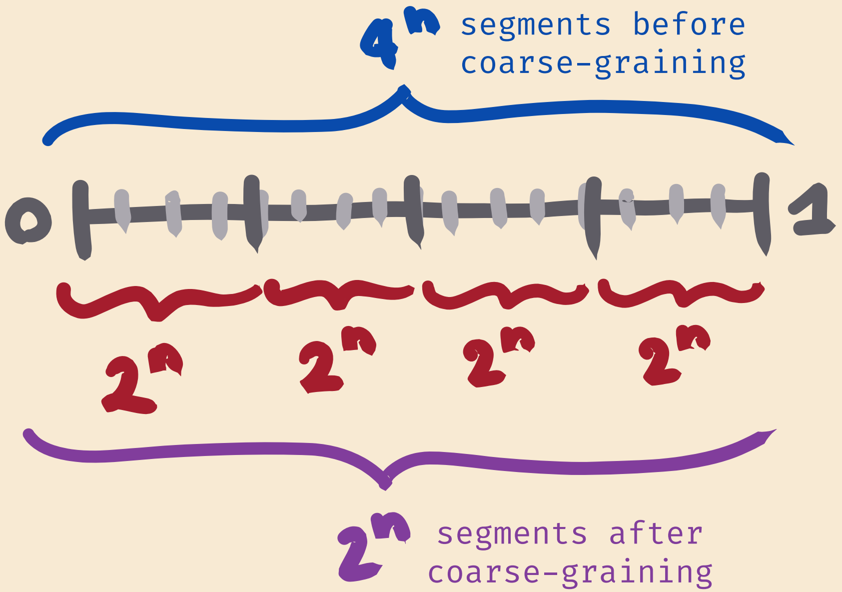
- → naive numerical kernel for [0,1]

$$\begin{aligned}\tilde{w}(\alpha_i, \beta_j) &= \tilde{J}_{\alpha_i,\beta_j} \\ &= \tilde{J}_{S(i),S(j)} \\ &\overset{\text{def}}{=} J_{ij} \\ &= w_U(\boldsymbol{v_i}, \boldsymbol{v_j})\end{aligned}$$

- → problem: **permutation invariance**
  ⇒ identical dynamics indep. of the mapping!

- → in 2D: 2^n bins per dimension
  in 1D: 4^n bins per dimension

- → reduce to 2^n bins in 1D by **coarse-graining**



↑ coarse-graining averages 2^n consecutive populations in 1D, forming effective bins of size 2^{-n}

let's simulate!

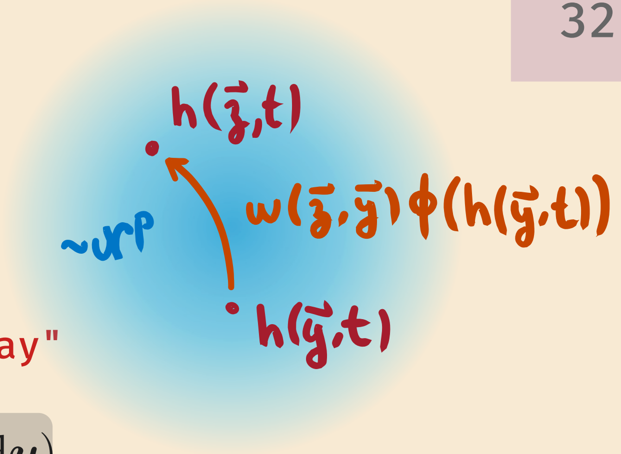
let's simulate!

…, but first, we'll need a toy model

let's simulate!

…, but first, we'll need a toy model

→ low-rank cycling gaussian neural field



"delay"

$$\partial_t h(\boldsymbol{z},t) = -h(\boldsymbol{z},t) + \sum_{\mu=1}^{p} \int_{\mathbb{R}^p} z_{\mu+1}\tilde{\phi}(y_\mu)\phi(h(\boldsymbol{y},t-\delta))\mathcal{N}^p(\mathrm{d}\boldsymbol{y})$$

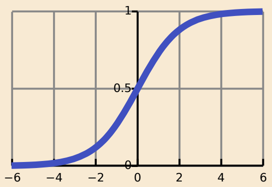
kernel has rank p

"rolling" induces recurrent drive to the next pattern

population density follows a p-dimensional Gaussian (zero mean, no covariance)

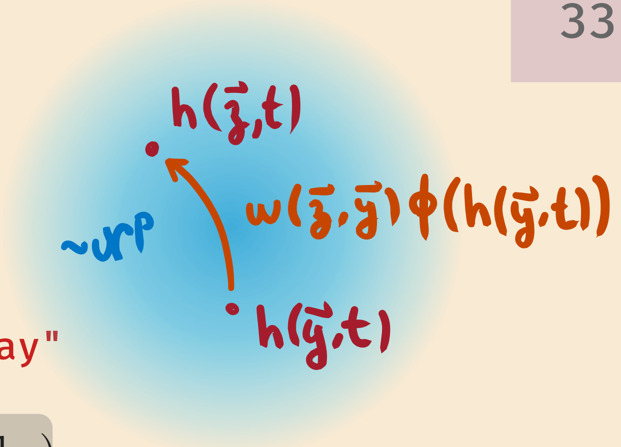
$$w(\boldsymbol{z},\boldsymbol{y}) = \sum_{\mu=1}^{p} z_\mu \tilde{\phi}(y_\mu)$$

$$\phi(h) = \frac{1}{1+\mathrm{e}^{-h}}$$



→ cycling behavior with "duration" δ

let's simulate!



..., but first, we'll need a toy model

→ low-rank cycling gaussian neural field

"delay"

$$\partial_t h(\boldsymbol{z},t) = -h(\boldsymbol{z},t) + \sum_{\mu=1}^{p} \int_{\mathbb{R}^p} z_{\mu+1}\tilde{\phi}(y_\mu)\phi(h(\boldsymbol{y},t-\delta))\mathcal{N}^p(\mathrm{d}\boldsymbol{y})$$

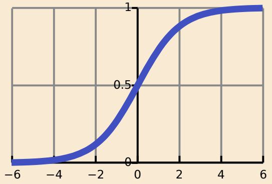
kernel has rank p

"rolling" induces recurrent drive to the next pattern

population density follows a p-dimensional Gaussian (zero mean, no covariance)

$$w(\boldsymbol{z},\boldsymbol{y}) = \sum_{\mu=1}^{p} z_\mu \tilde{\phi}(y_\mu)$$

$$\phi(h) = \frac{1}{1+\mathrm{e}^{-h}}$$

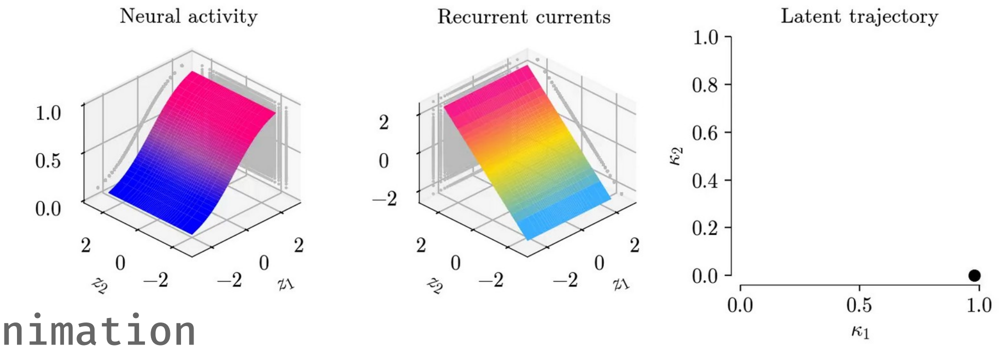


→ cycling behavior with "duration" δ

→ we get latent trajectories for free:

projection of the field onto the patterns

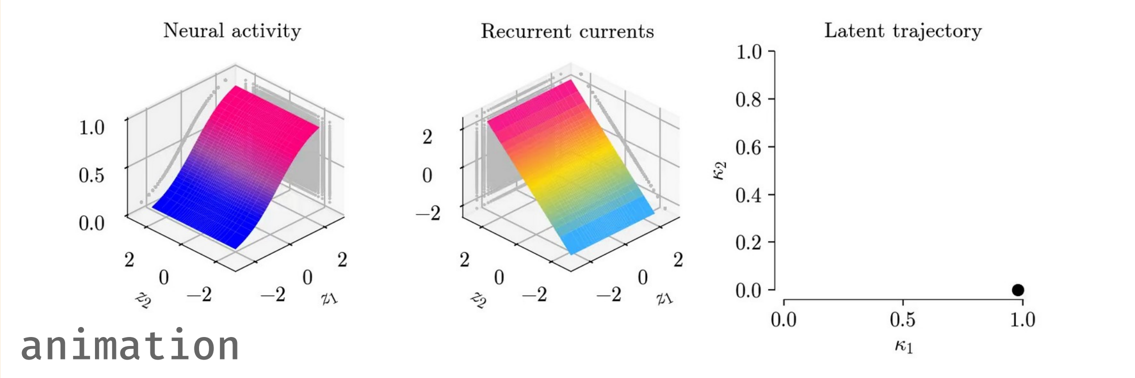
$$\kappa_\mu(t) = \int_{\mathbb{R}^p} y_\mu h(\boldsymbol{y},t)\mathcal{N}^p(\mathrm{d}\boldsymbol{y})$$



animation

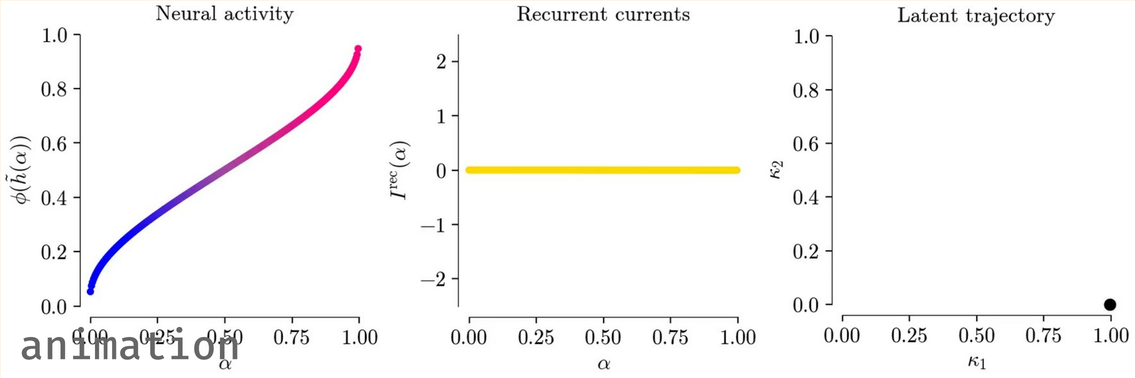
← original neural field for $p=2, \delta=6, h(z_1, z_2, 0) = z_1$

⇒ cycling behavior



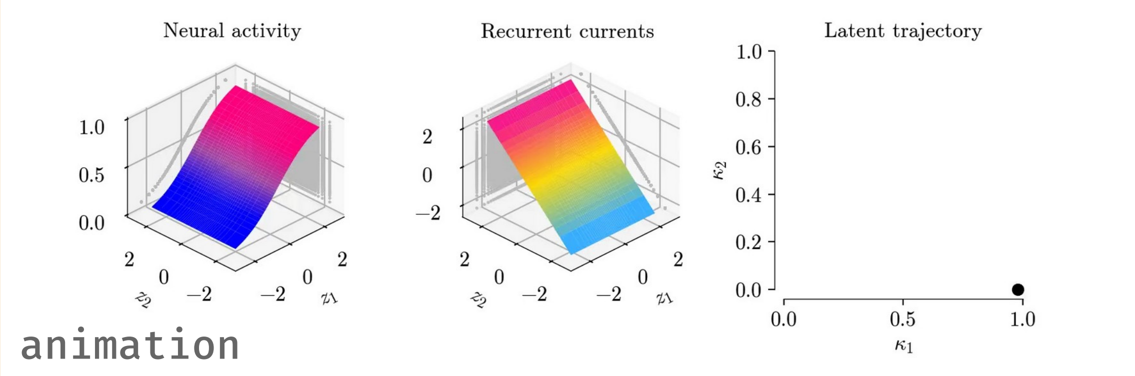
← original neural field for $p=2, \delta=6, h(z_1, z_2, 0) = z_1$

⇒ cycling behavior



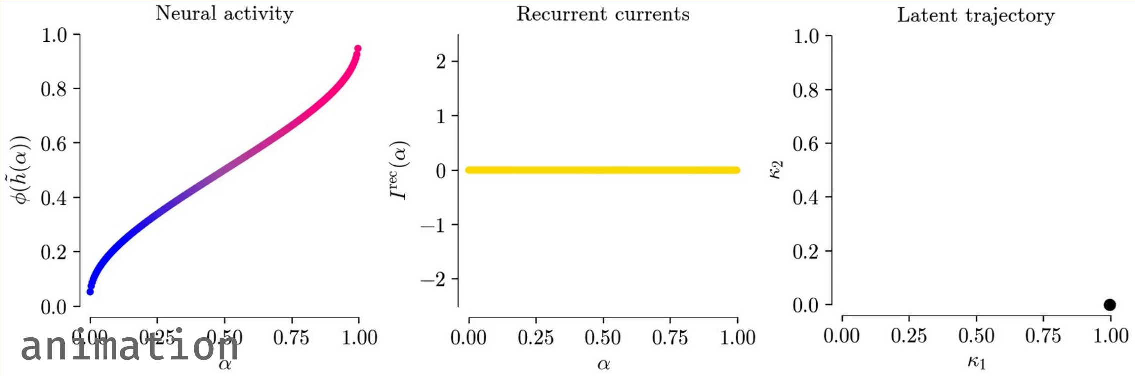
← mapped with Column mapping, then coarse-grained

⇒ quick decay



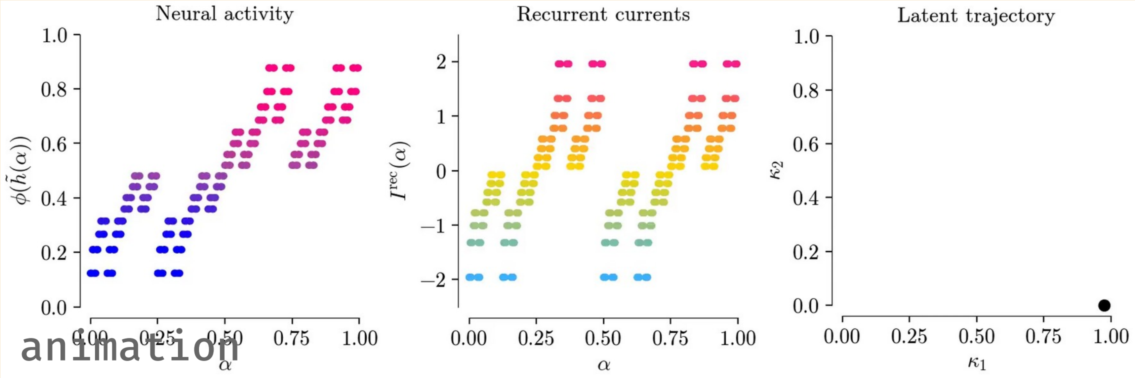
← original neural field for $p=2, \delta=6, h(z_1,z_2,0)=z_1$

⇒ cycling behavior



← mapped with Column mapping, then coarse-grained

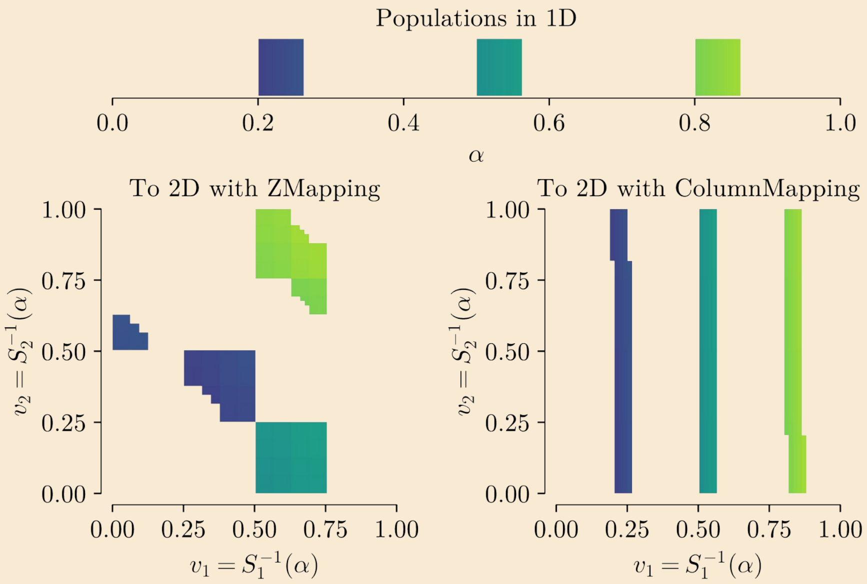
⇒ quick decay



← mapped with Z-mapping, then coarse-grained

⇒ identical cycling behavior

the notion of locality – motivation

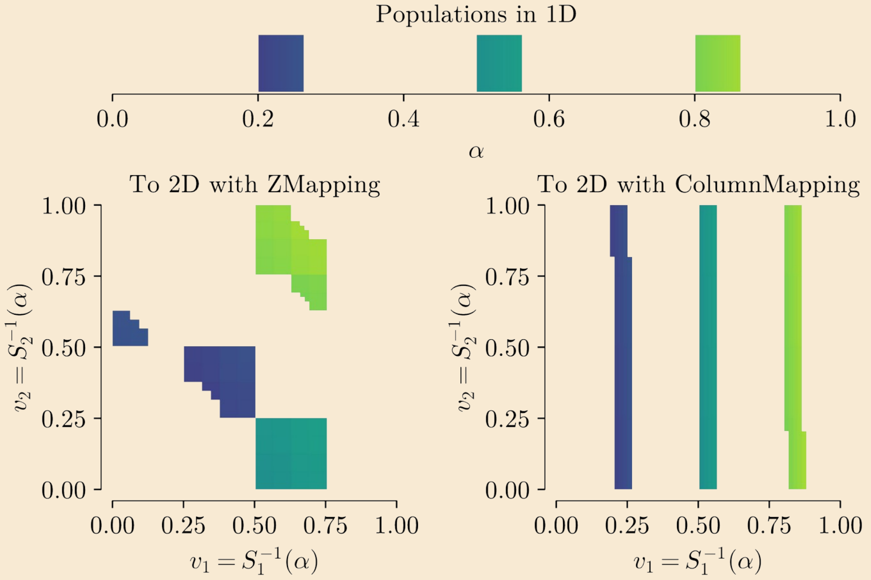


→ are populations close in 1D also close in 2D ?

recall: $\tilde{w}(\alpha, \beta) = w_U(S^{-1}(\alpha), S^{-1}(\beta))$

← 2D populations corresponding to three small segments (each of length 1/16) of 1D populations

the notion of locality - motivation



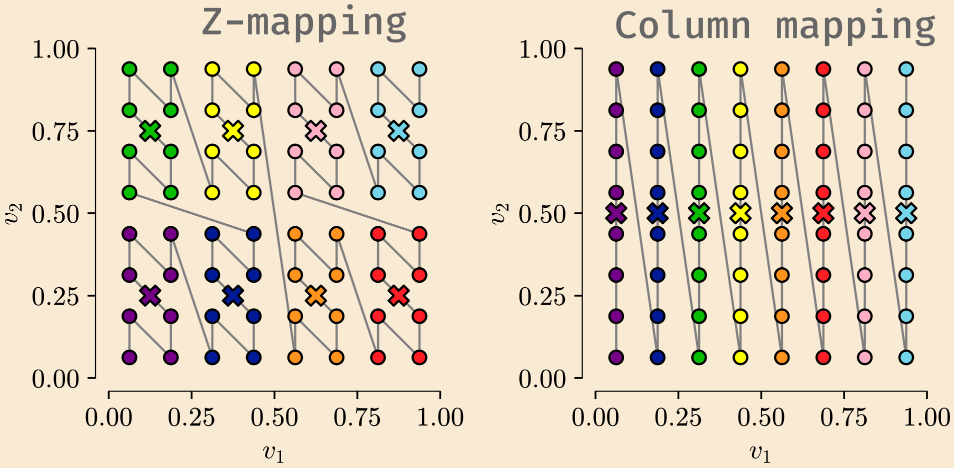
→ are populations close in 1D also close in 2D ?

recall: $\tilde{w}(\alpha, \beta) = w_U(S^{-1}(\alpha), S^{-1}(\beta))$

← 2D populations corresponding to three small segments (each of length 1/16) of 1D populations

→ what positions in 2D get "averaged together" ?

colour-coded square populations corresponding to binned segment populations before coarse-graining →



the notion of locality - formalism

→ average variation inside each coarse-graining bin

$$V_n(S^{-1}) = \frac{1}{2^n}\sum_{i=1}^{2^n} \sup_{\alpha,\alpha' \in \left[\frac{i-1}{2^n}, \frac{i}{2^n}\right]^2} \|S^{-1}(\alpha') - S^{-1}(\alpha)\|_1$$

average over all bins

every pair of positions inside each bin

distance in the p-dimensional embedding

→ if f is continuous, then $V_n(f) \to 0$

the notion of locality – formalism

→ average variation inside each coarse-graining bin

$$V_n(S^{-1}) = \frac{1}{2^n}\sum_{i=1}^{2^n} \sup_{\alpha,\alpha' \in \left[\frac{i-1}{2^n}, \frac{i}{2^n}\right]^2} \|S^{-1}(\alpha') - S^{-1}(\alpha)\|_1$$

average over all bins

every pair of positions inside each bin

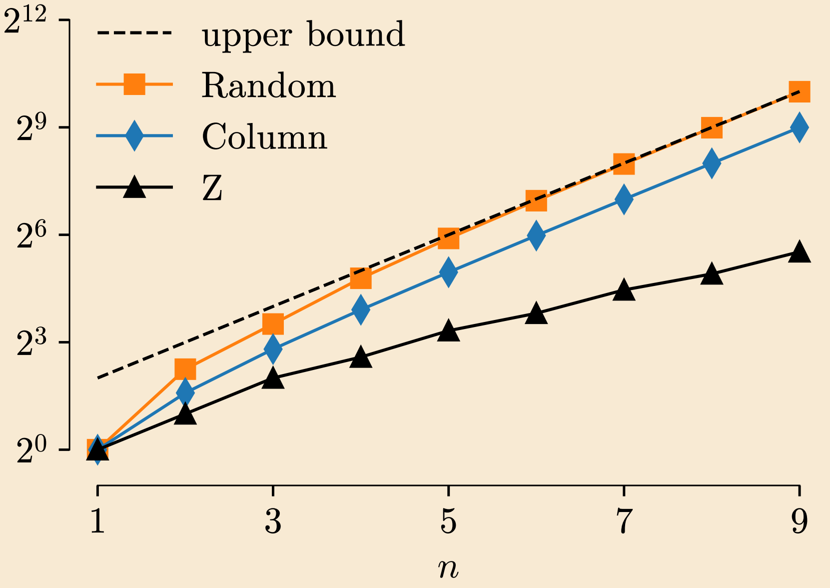
distance in the p-dimensional embedding

→ if f is continuous, then $V_n(f) \to 0$

→ scaling behavior of numerator

- Random mapping saturates bound
- Column mapping ~ 2^n ⇒ $V_n(C^{-1}) > 0$
- Z-mapping ~ $2^{n/2}$ ⇒ $V_n(Z^{-1}) \to 0$

↓ (numerator of the) average binned variation as a function of n



locality guarantees simulation accuracy

→ $\tilde{w}$ is regular enough that the integral on [0,1] can be numerically evaluated. sketch of proof:

locality guarantees simulation accuracy

- $\tilde{w}$ is regular enough that the integral on [0,1] can be numerically evaluated. sketch of proof:
  0) define numerical and analytical integrals

$$\mathrm{NI}_n = \frac{1}{2^n}\sum_{i=1}^{2^n} \tilde{w}(\alpha,\beta_i)\phi(\tilde{h}_i(t)), \quad \mathrm{AI} = \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)$$

locality guarantees simulation accuracy

→ $\tilde{w}$ is regular enough that the integral on [0,1] can be numerically evaluated. sketch of proof:

0) define numerical and analytical integrals

$$\mathrm{NI}_n = \frac{1}{2^n}\sum_{i=1}^{2^n} \tilde{w}(\alpha,\beta_i)\phi(\tilde{h}_i(t)), \quad \mathrm{AI} = \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)$$

1) compare the integrals on each coarse-grained bin

$$|\mathrm{NI}_n - \mathrm{AI}| \leq \sum_{i=1}^{2^n}\left|\int_{\frac{i-1}{2^n}}^{\frac{i}{2^n}} \tilde{w}(\alpha,\beta_i)\phi(\tilde{h}_i(t)) - \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)\right|$$

locality guarantees simulation accuracy

→ $\tilde{w}$ is regular enough that the integral on [0,1] can be numerically evaluated. sketch of proof:

0) define numerical and analytical integrals

$$\mathrm{NI}_n = \frac{1}{2^n}\sum_{i=1}^{2^n} \tilde{w}(\alpha,\beta_i)\phi(\tilde{h}_i(t)), \quad \mathrm{AI} = \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)$$

1) compare the integrals on each coarse-grained bin

$$|\mathrm{NI}_n - \mathrm{AI}| \leq \sum_{i=1}^{2^n}\left|\int_{\frac{i-1}{2^n}}^{\frac{i}{2^n}} \tilde{w}(\alpha,\beta_i)\phi(\tilde{h}_i(t)) - \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)\right|$$

2) assume Lipschiz continuity of the original neural field with cst $L(\alpha)$

$$|\mathrm{NI}_n - \mathrm{AI}| \leq L(\alpha)\sum_{i=1}^{2^n}\left|\int_{\frac{i-1}{2^n}}^{\frac{i}{2^n}} \|S^{-1}(\beta_i) - S^{-1}(\beta)\|\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)\right|$$

locality guarantees simulation accuracy

→ $\tilde{w}$ is regular enough that the integral on [0,1] can be numerically evaluated. sketch of proof:

0) define numerical and analytical integrals

$$\mathrm{NI}_n = \frac{1}{2^n}\sum_{i=1}^{2^n} \tilde{w}(\alpha,\beta_i)\phi(\tilde{h}_i(t)), \quad \mathrm{AI} = \int_{[0,1]} \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)$$

1) compare the integrals on each coarse-grained bin

$$|\mathrm{NI}_n - \mathrm{AI}| \leq \sum_{i=1}^{2^n}\left|\int_{\frac{i-1}{2^n}}^{\frac{i}{2^n}} \tilde{w}(\alpha,\beta_i)\phi(\tilde{h}_i(t)) - \tilde{w}(\alpha,\beta)\phi(\tilde{h}(\beta,t))\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)\right|$$

2) assume Lipschiz continuity of the original neural field with cst $L(\alpha)$

$$|\mathrm{NI}_n - \mathrm{AI}| \leq L(\alpha)\sum_{i=1}^{2^n}\left|\int_{\frac{i-1}{2^n}}^{\frac{i}{2^n}} \|S^{-1}(\beta_i) - S^{-1}(\beta)\|\left[\lambda\circ S^{-1}\right](\mathrm{d}\beta)\right|$$

3) bound integrand by supremum and use locality

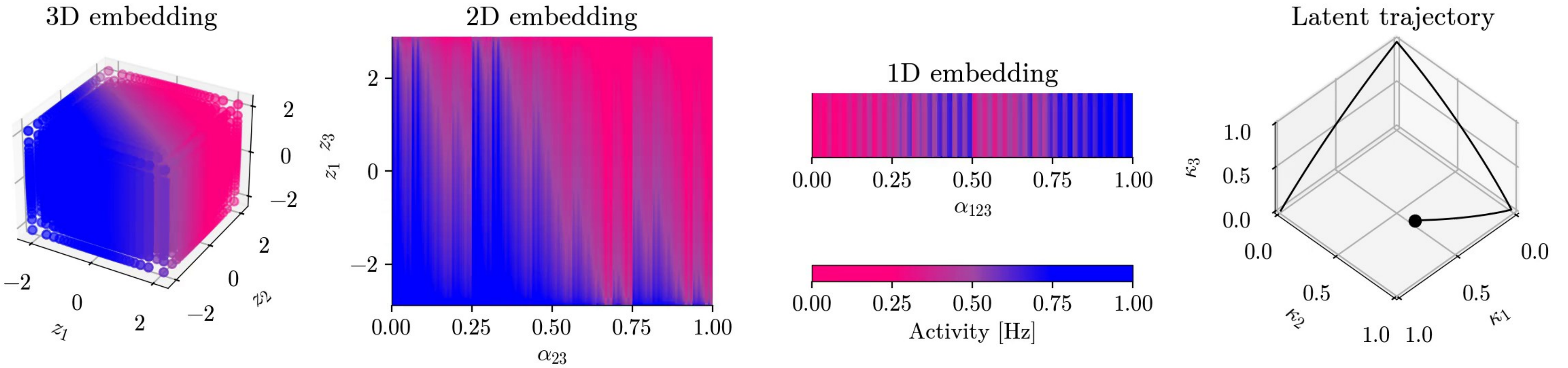
$$|\mathrm{NI}_n - \mathrm{AI}| \leq L(\alpha)V_n(S^{-1}) \xrightarrow{n\to\infty} 0$$

(bonus) iterating mapping and coarse-graining from p-dim to 1-dim

- $\tilde{w}$ is regular enough that we can again apply a mapping and do coarse-graining
- **"local mappings conserve regularity"**

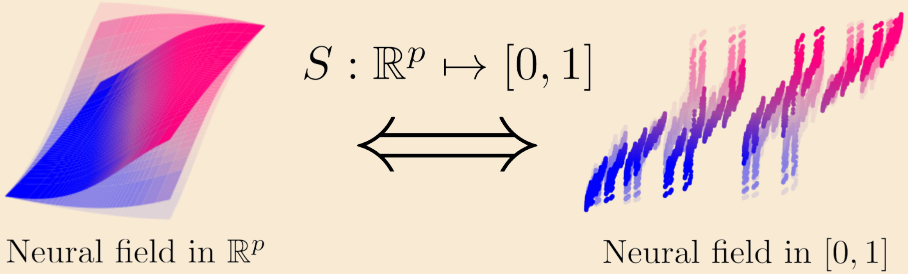
(bonus) iterating mapping and coarse-graining from p-dim to 1-dim

- $\tilde{w}$ is regular enough that we can again apply a mapping and do coarse-graining
- **"local mappings conserve regularity"**
- numerical demo with p=3: from $[0,1]^3$ to $[0,1]^2$ to $[0,1]$



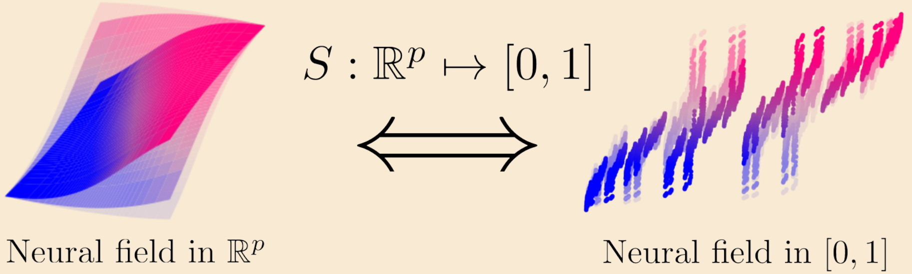
summary & conclusion

- example of neural fields in p dimensions that can be mapped to **equivalent neural fields** in [0,1]
- locality quantifies the **conservation of "regularity"**
- **coarse-graining** numerically enforces the notion of locality: non local mappings are destroyed
- the Z-mapping can be **approximated numerically**, and the numerical integral on [0,1] approaches the analytical integral on [0,1]



summary & conclusion

- example of neural fields in p dimensions that can be mapped to **equivalent neural fields** in [0,1]
- locality quantifies the **conservation of "regularity"**
- **coarse-graining** numerically enforces the notion of locality: non local mappings are destroyed
- the Z-mapping can be **approximated numerically**, and the numerical integral on [0,1] approaches the analytical integral on [0,1]



Thank you!