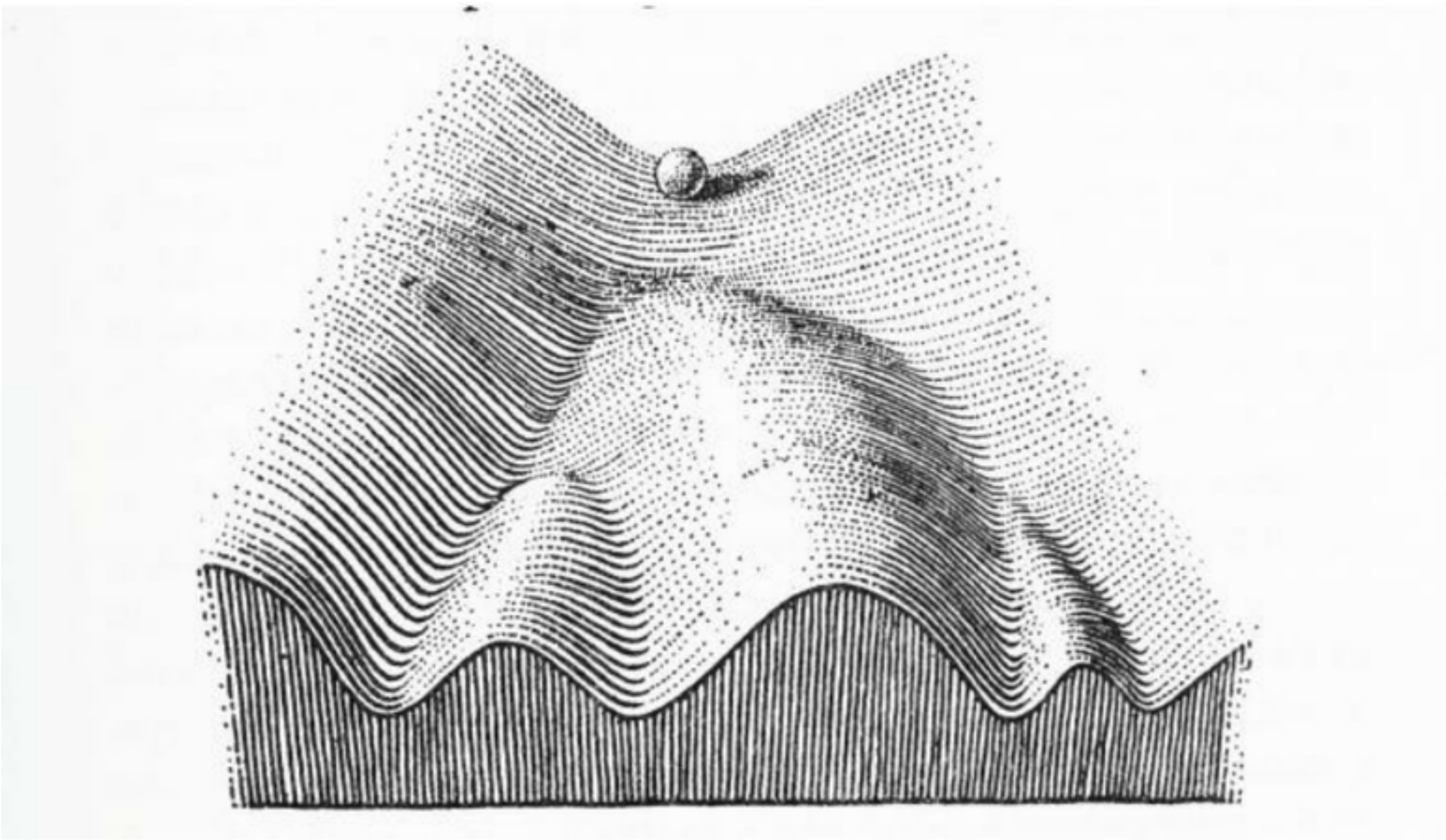
Finding *separatrices* of dynamical flows with *Deep Koopman Eigenfunctions*

Kabir V. Dabholkar and Omri Barak

multistability in nature

multistability in nature

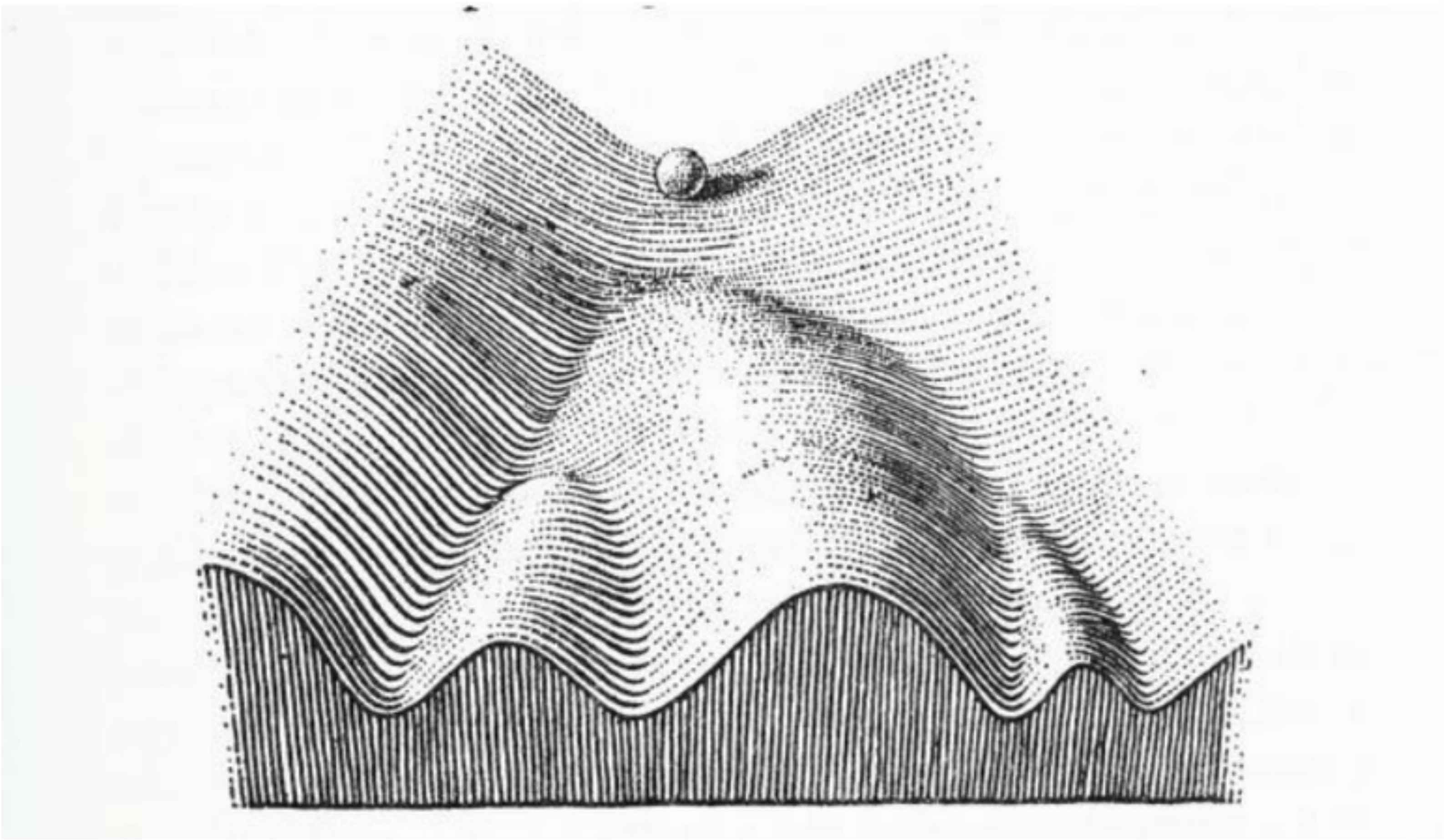
Cell biology



Waddington 1957

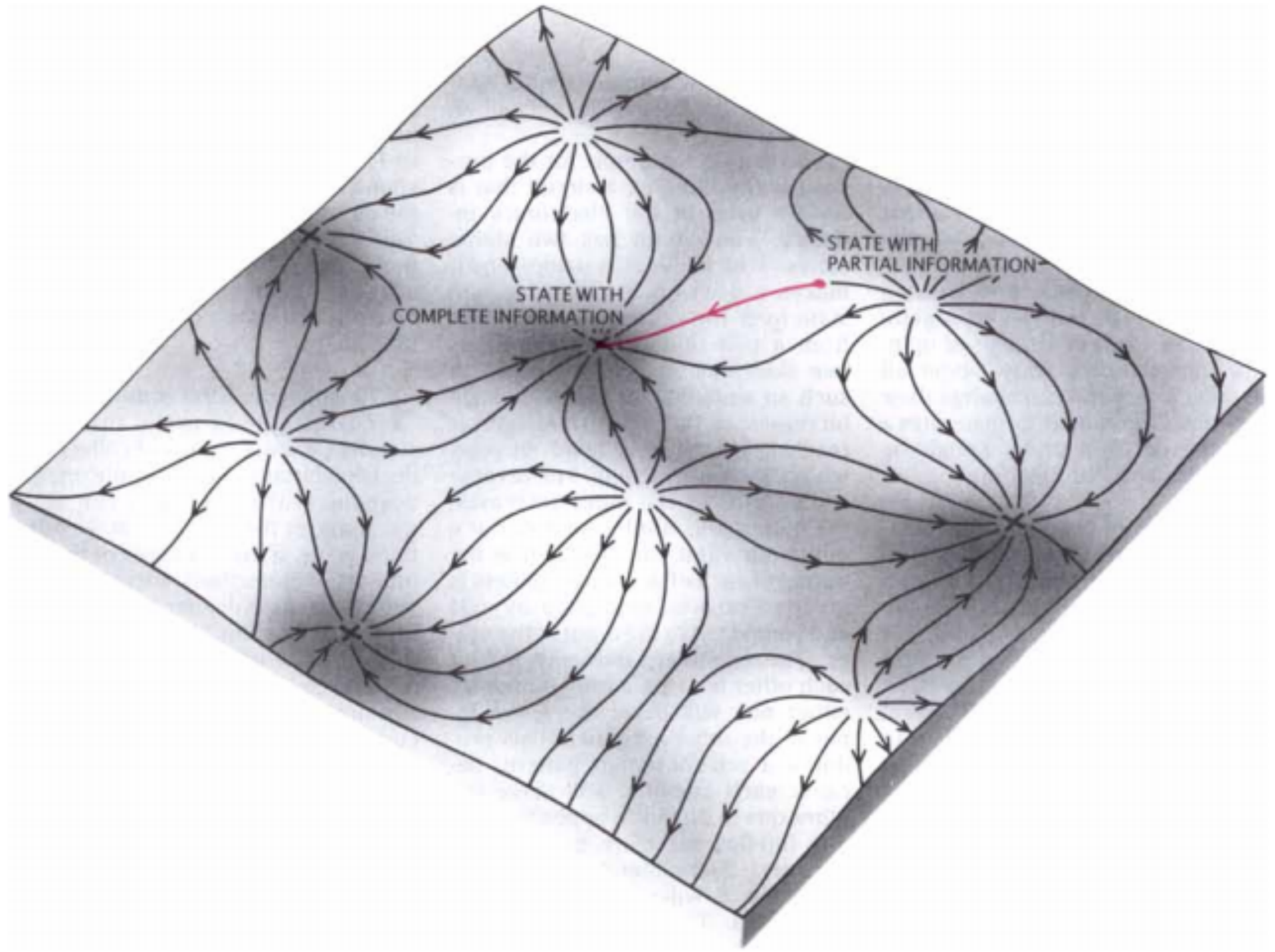
multistability in nature

Cell biology



Waddington 1957

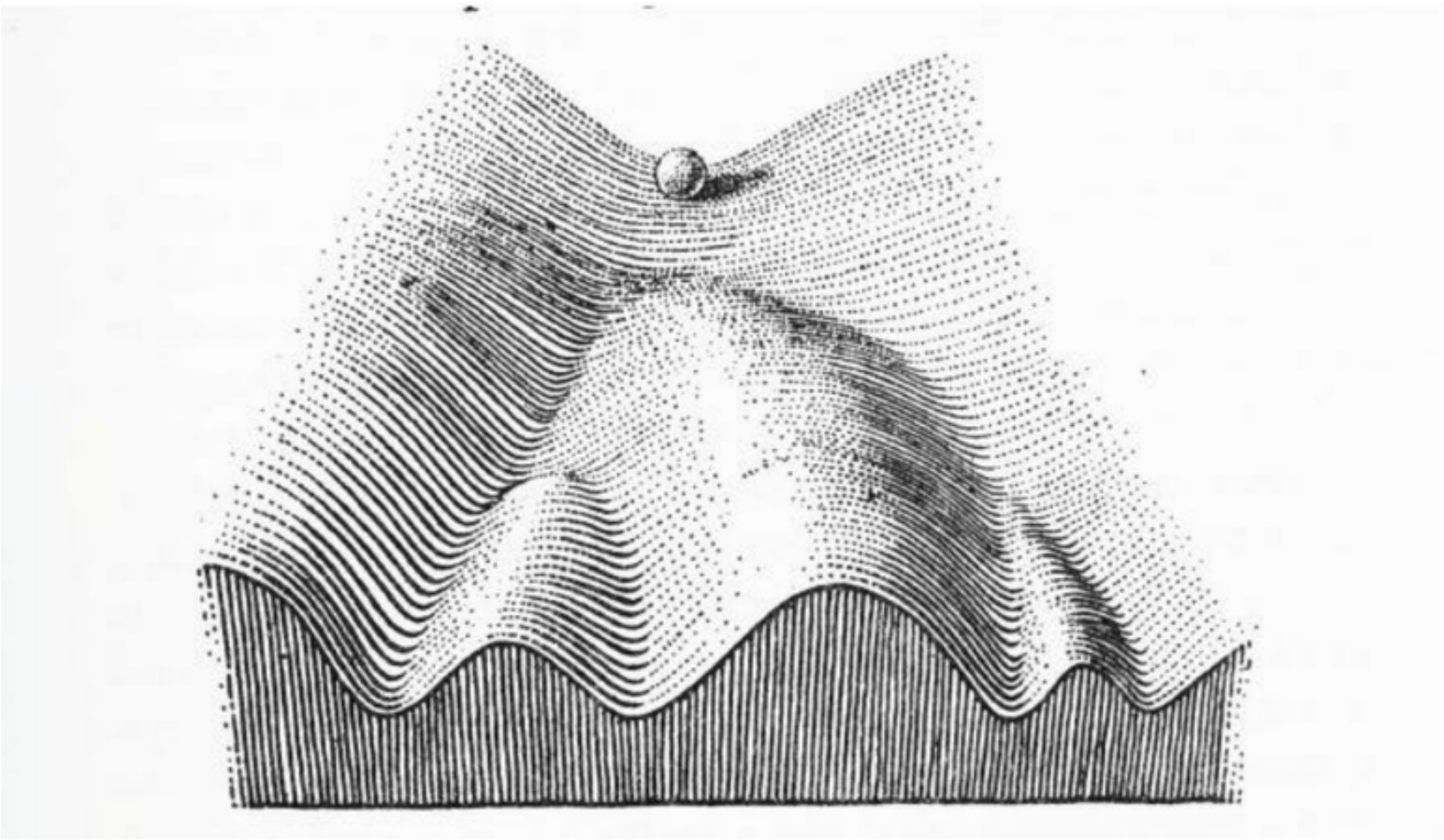
Neuroscience



Tank and Hopfield 1987

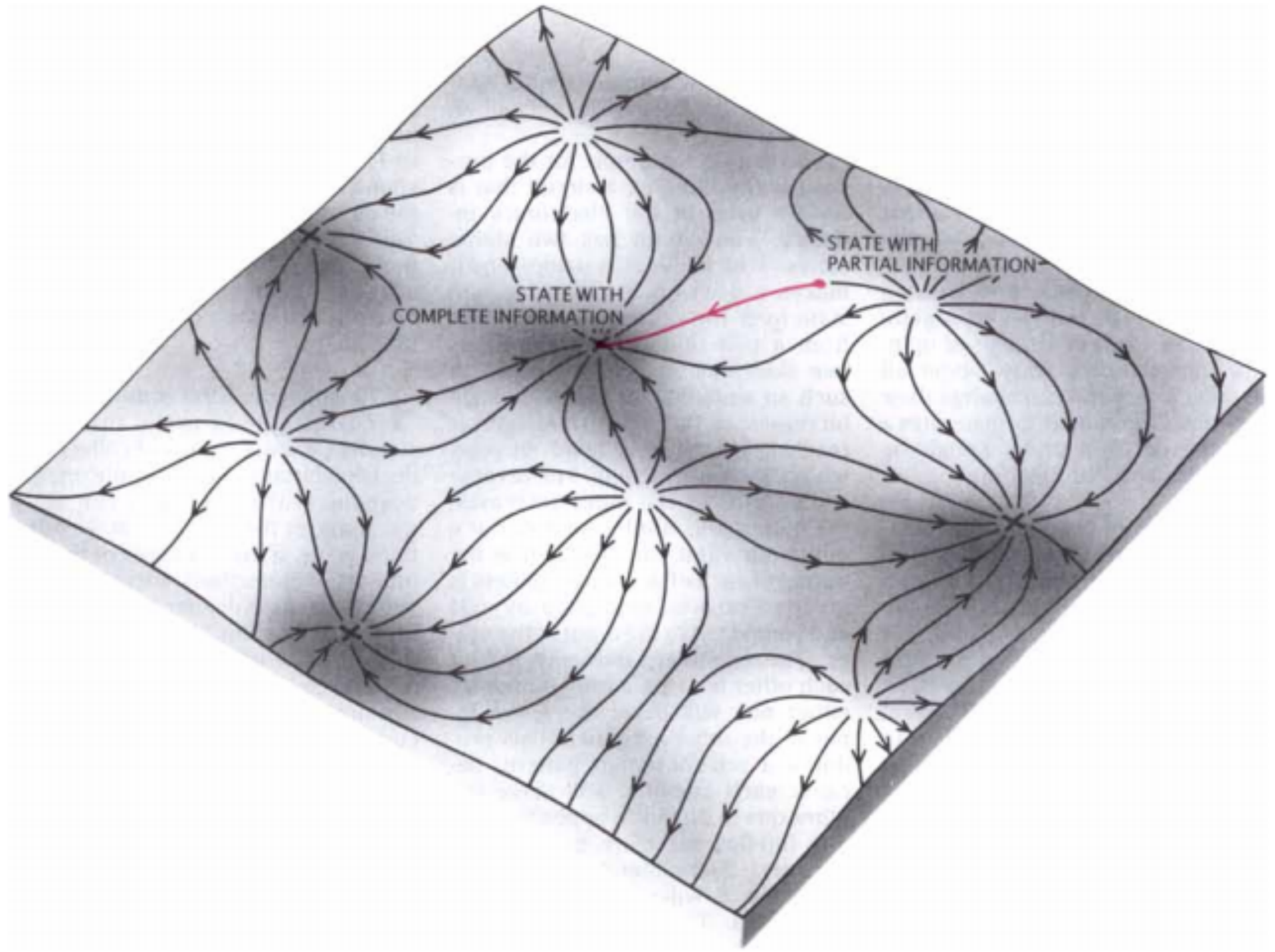
multistability in nature

Cell biology



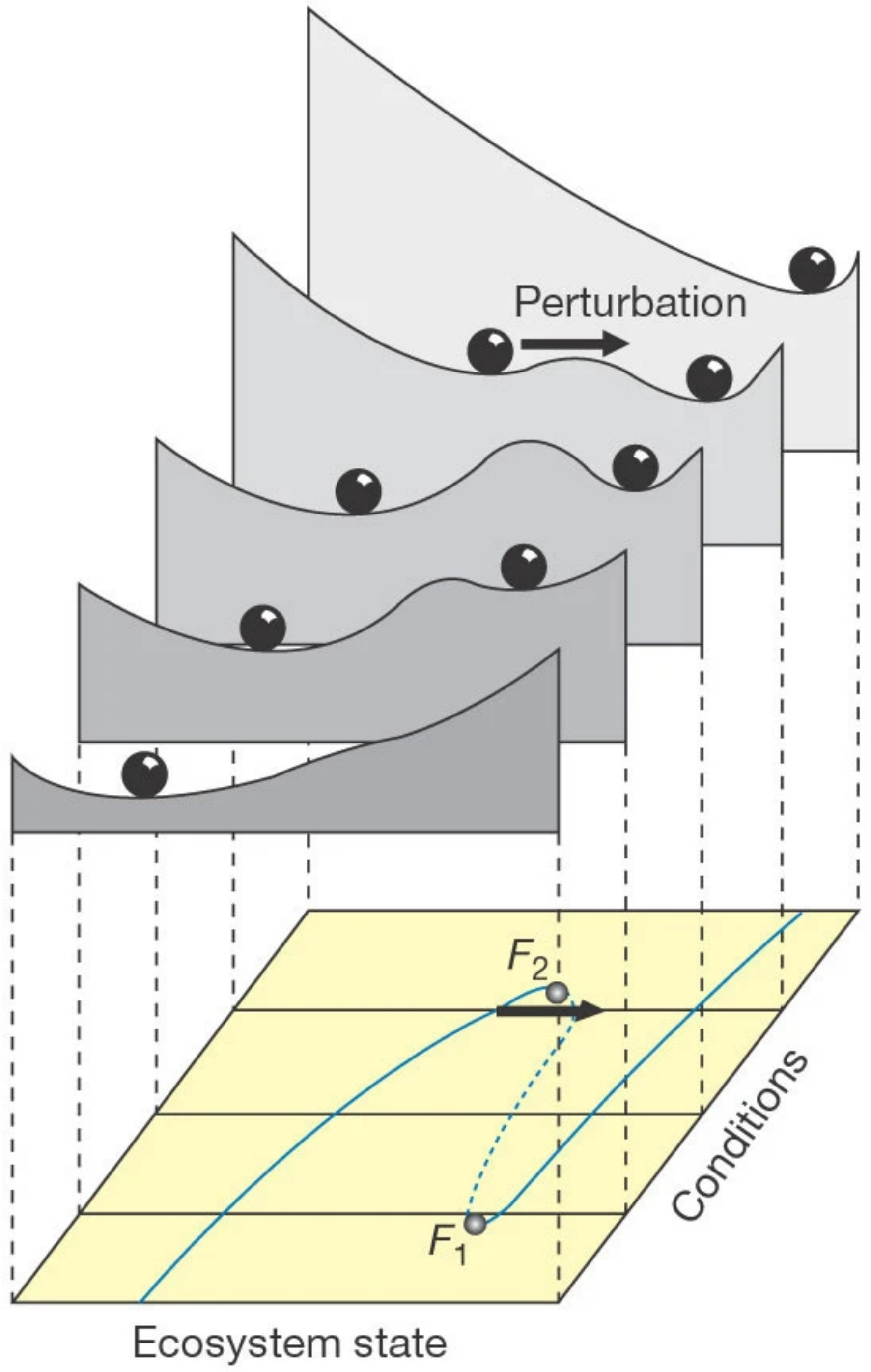
Waddington 1957

Neuroscience



Tank and Hopfield 1987

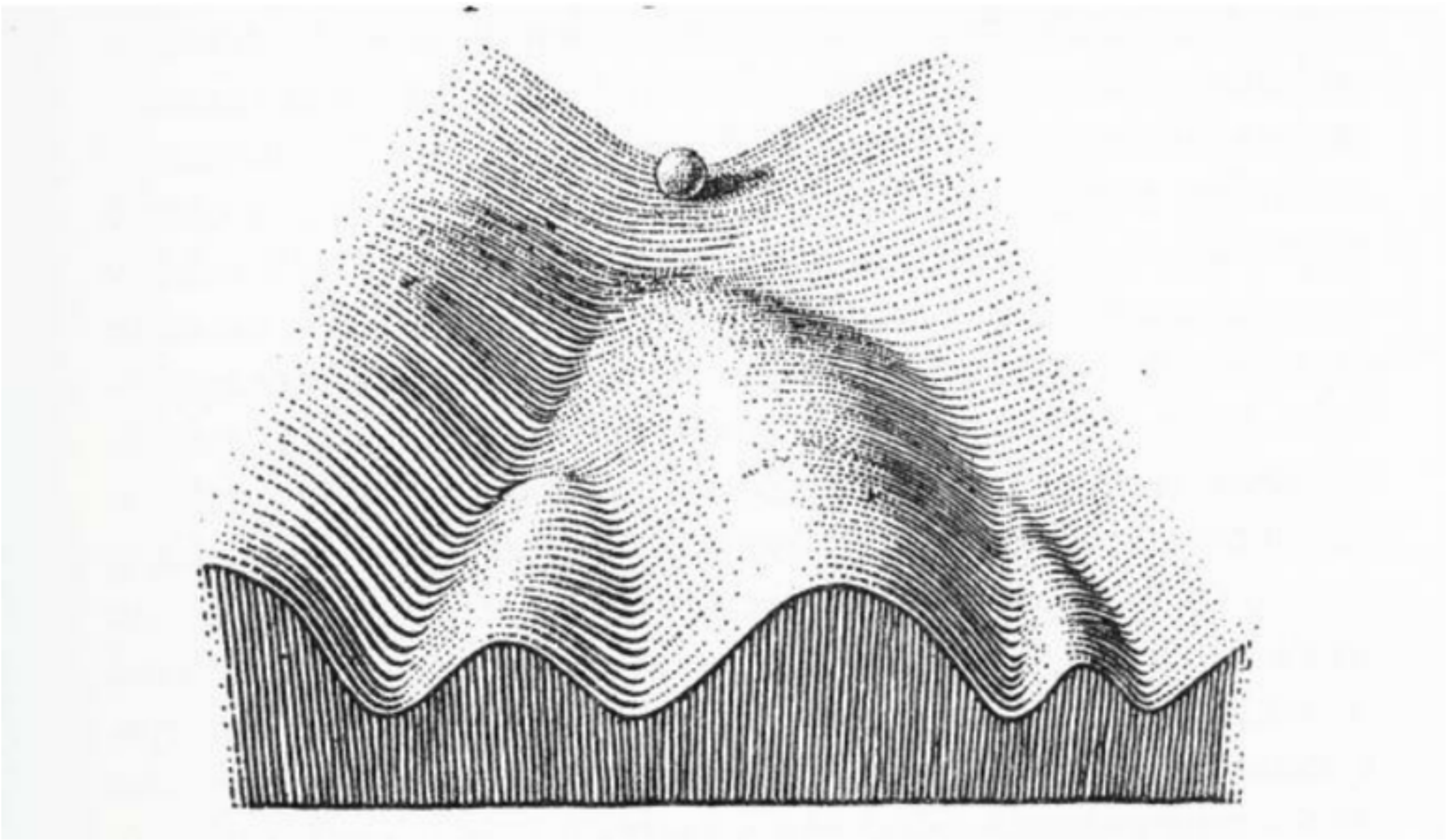
Ecology



Scheffer et al 2001

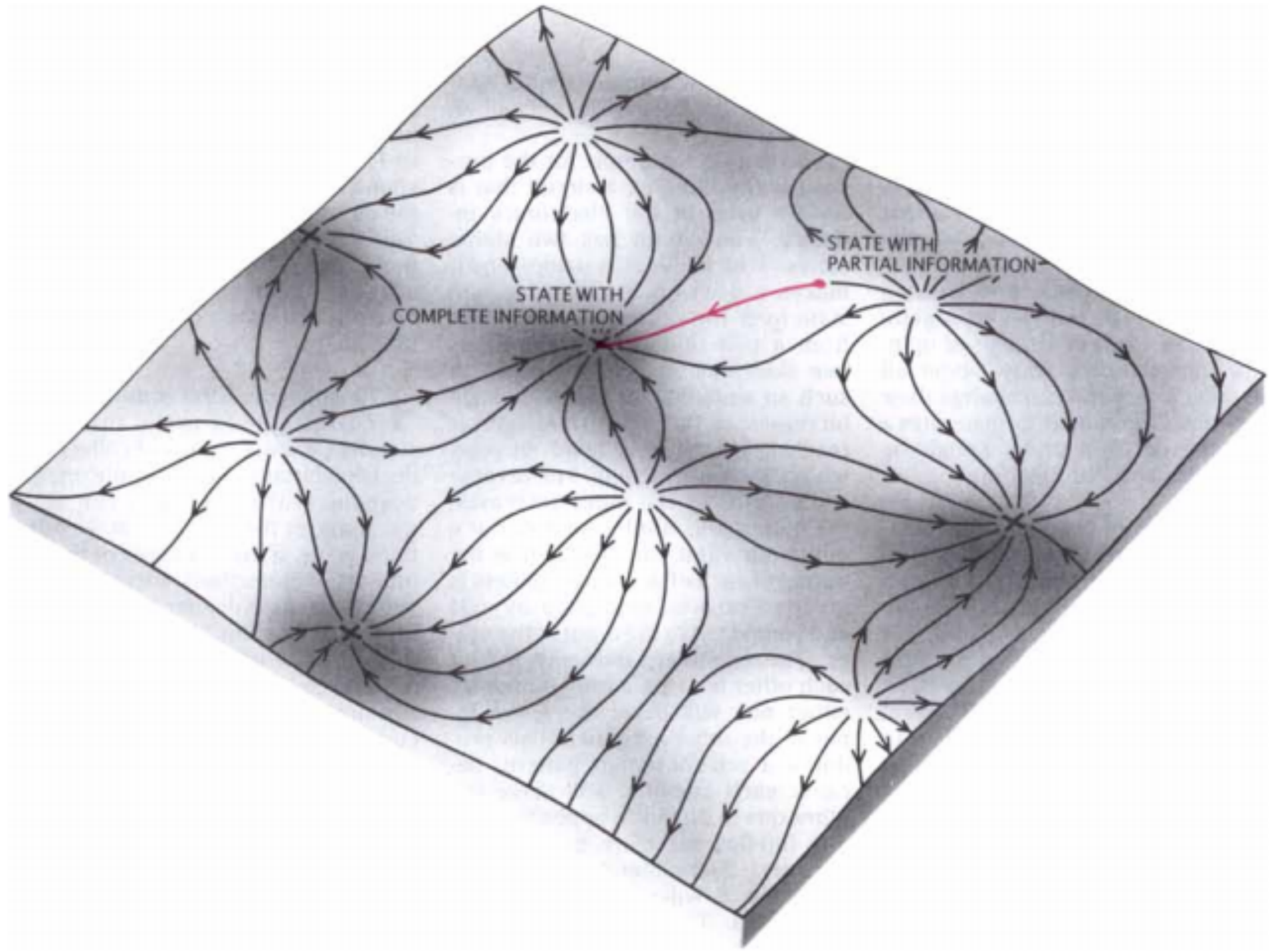
multistability in nature

Cell biology



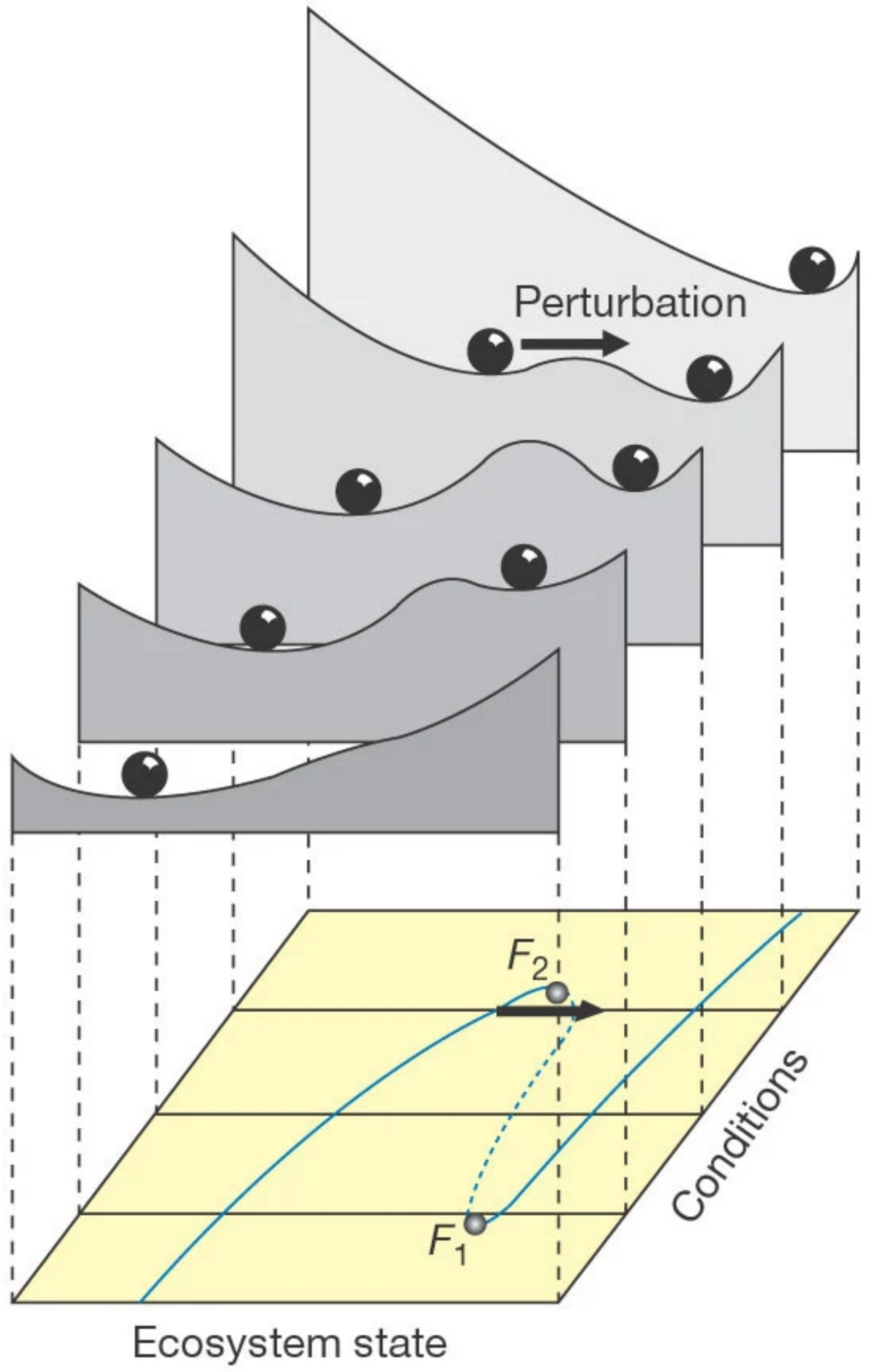
Waddington 1957

Neuroscience



Tank and Hopfield 1987

Ecology

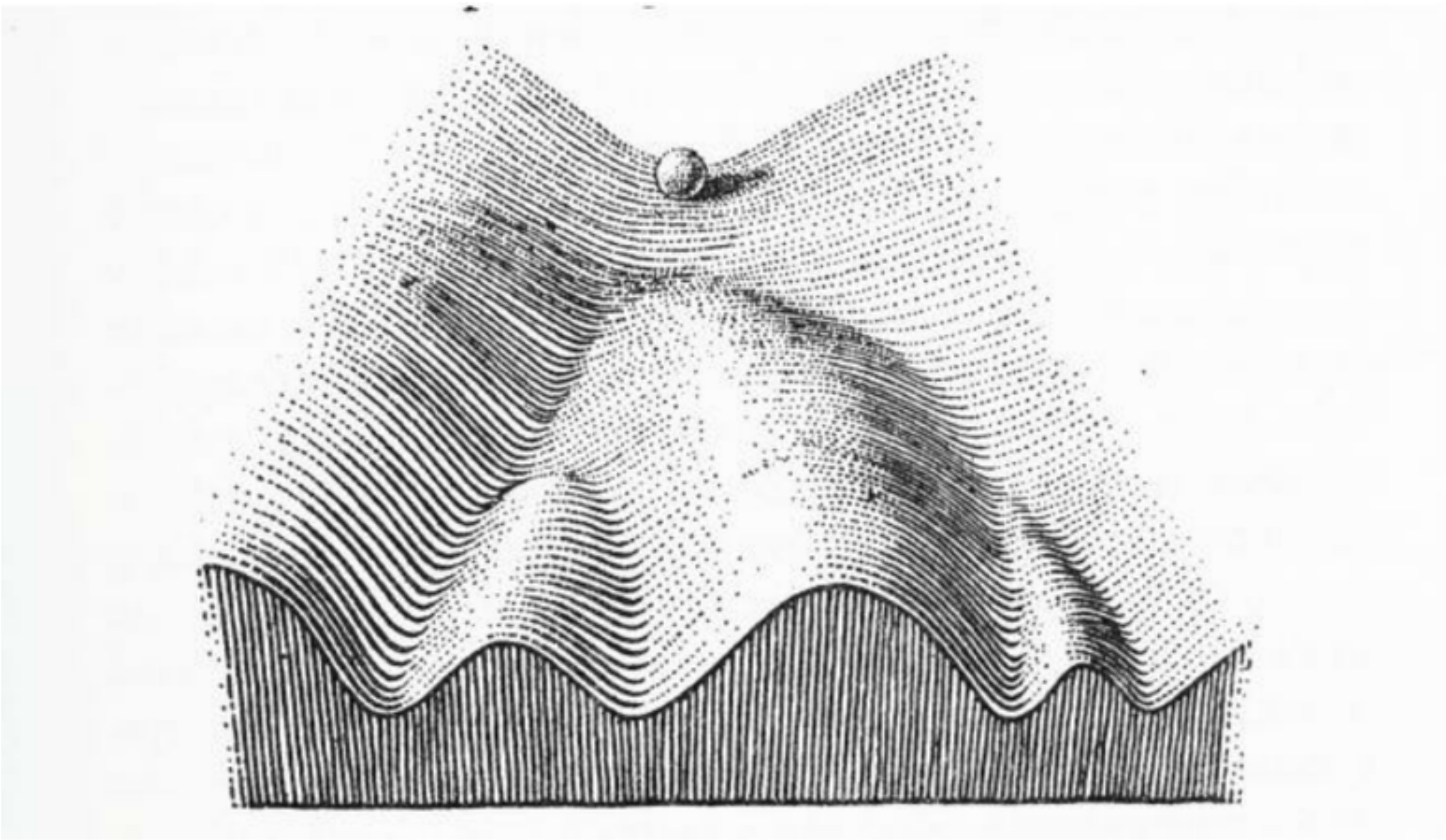


Scheffer et al 2001

complex, high-dimensional dynamics

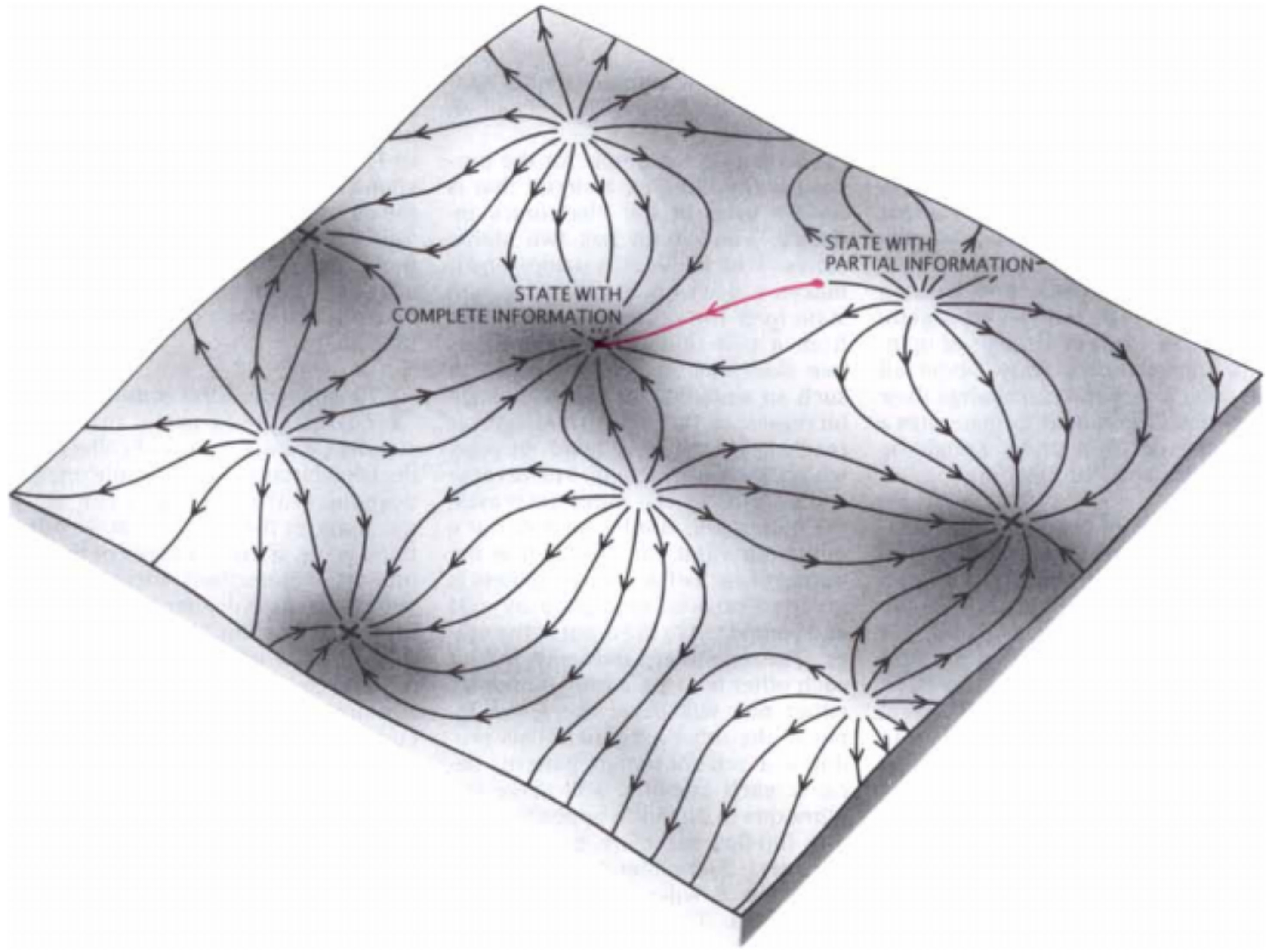
multistability in nature

Cell biology



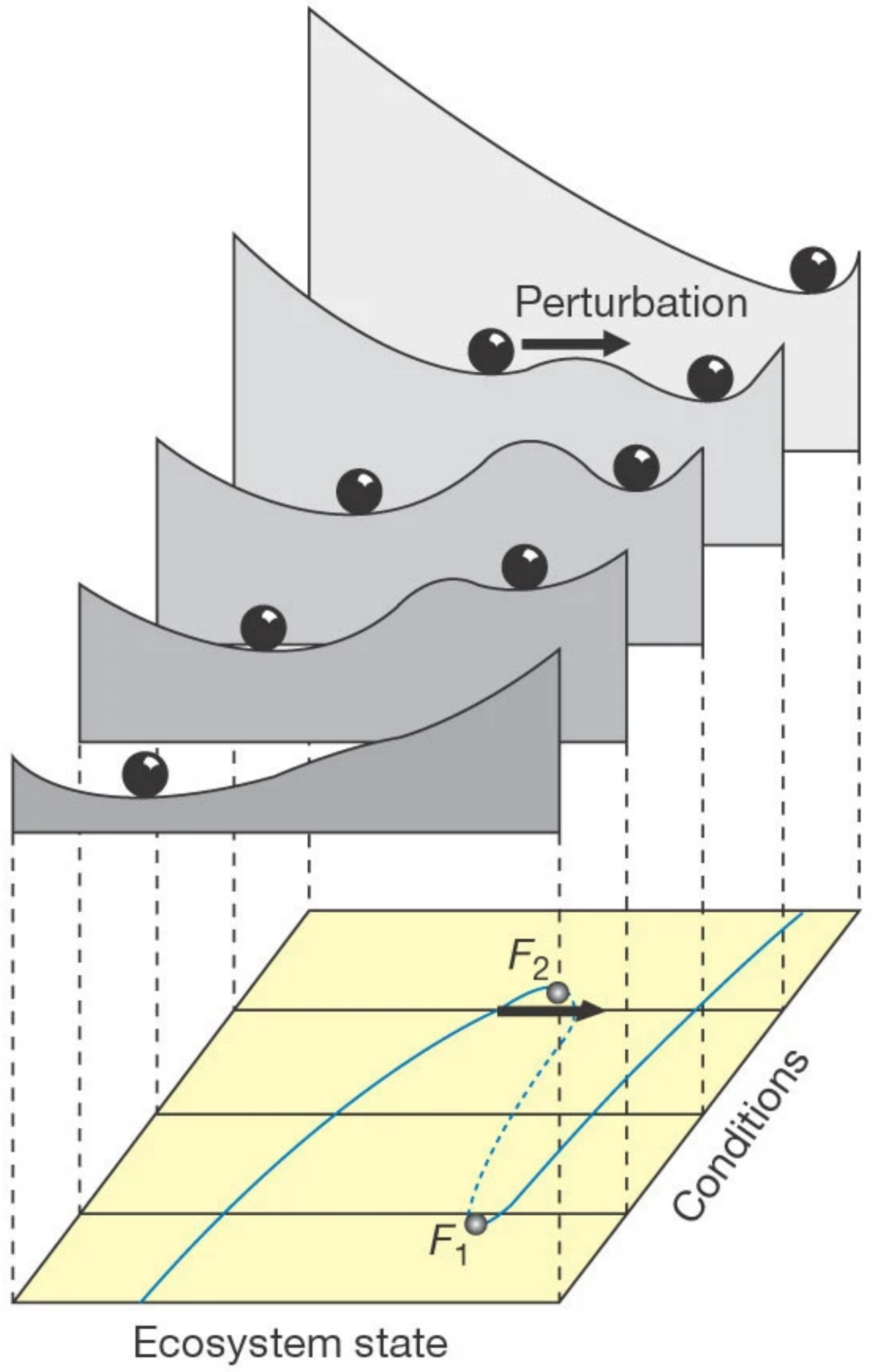
Waddington 1957

Neuroscience



Tank and Hopfield 1987

Ecology

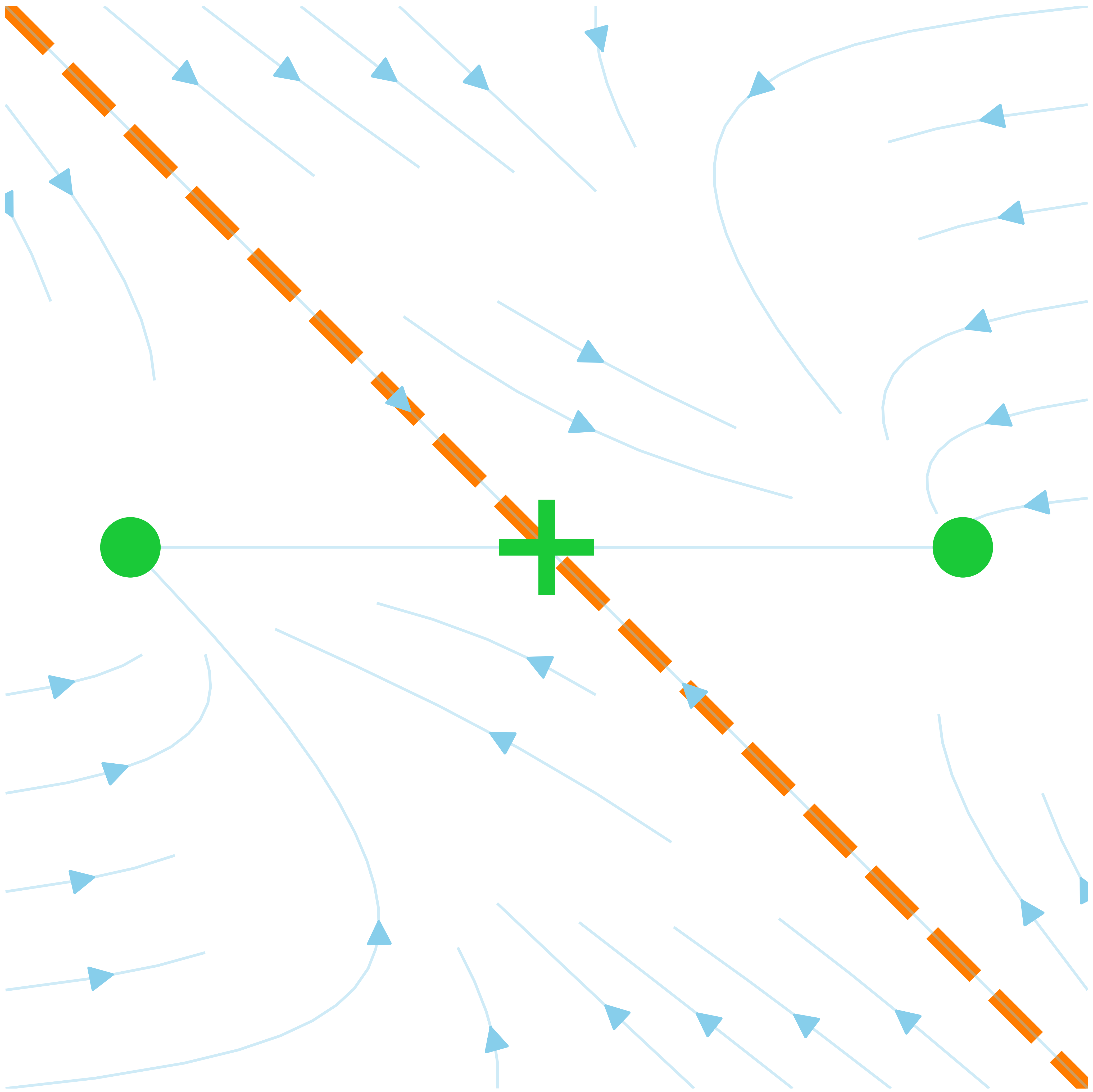


Scheffer et al 2001

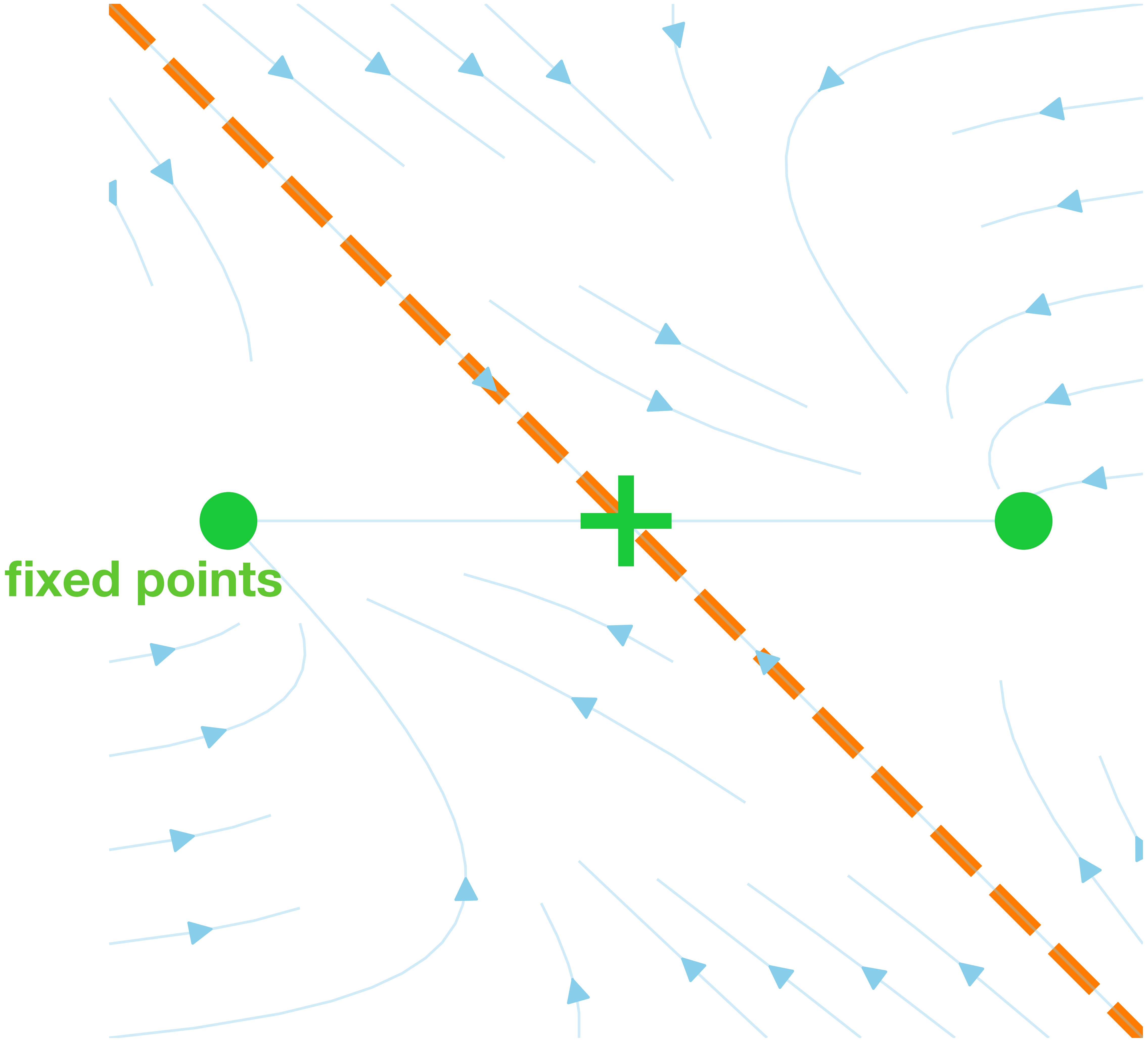
complex, high-dimensional dynamics

need for better tools to understand models

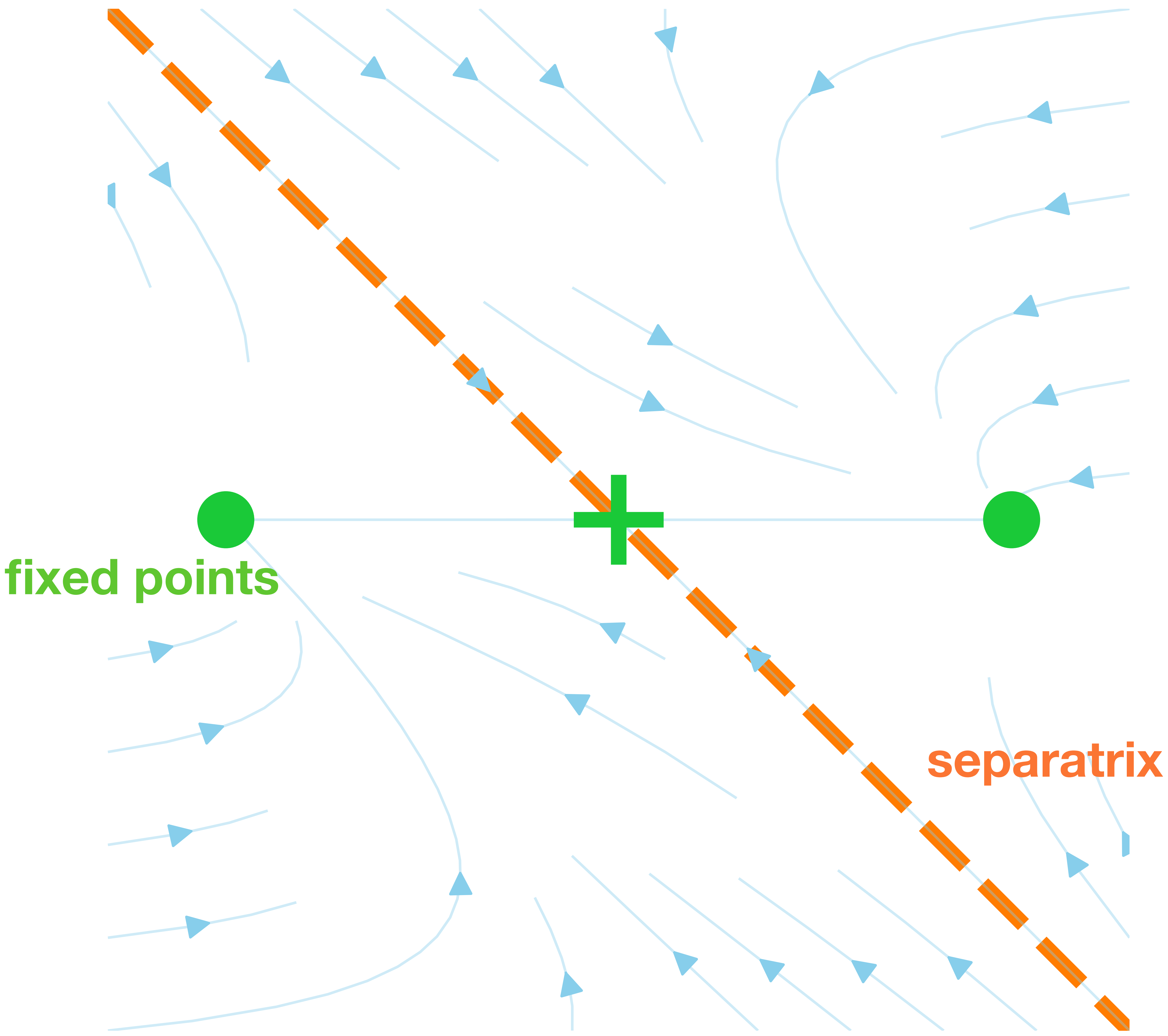
$$\dot{x} = f(x)$$



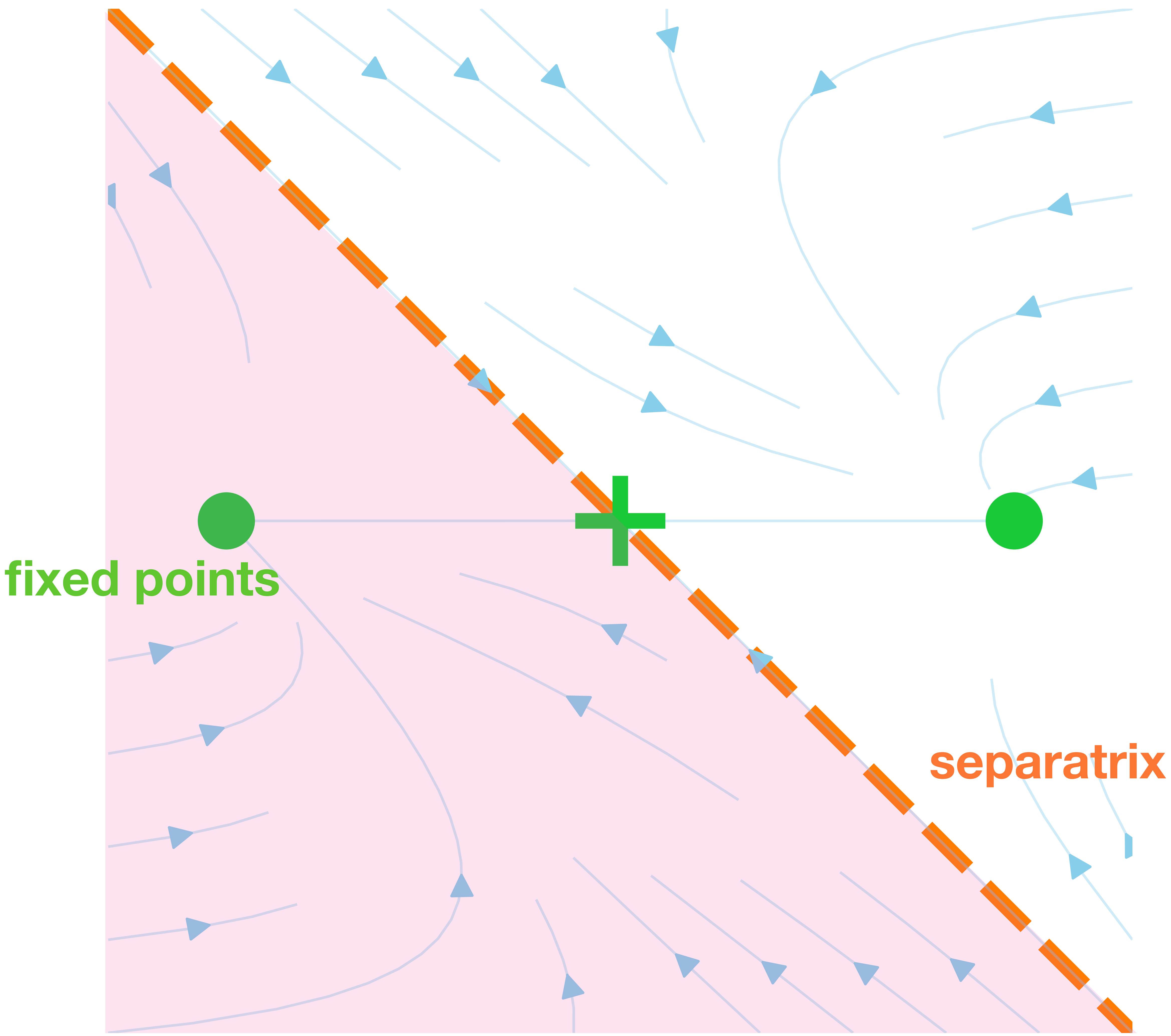
$$\dot{x} = f(x)$$



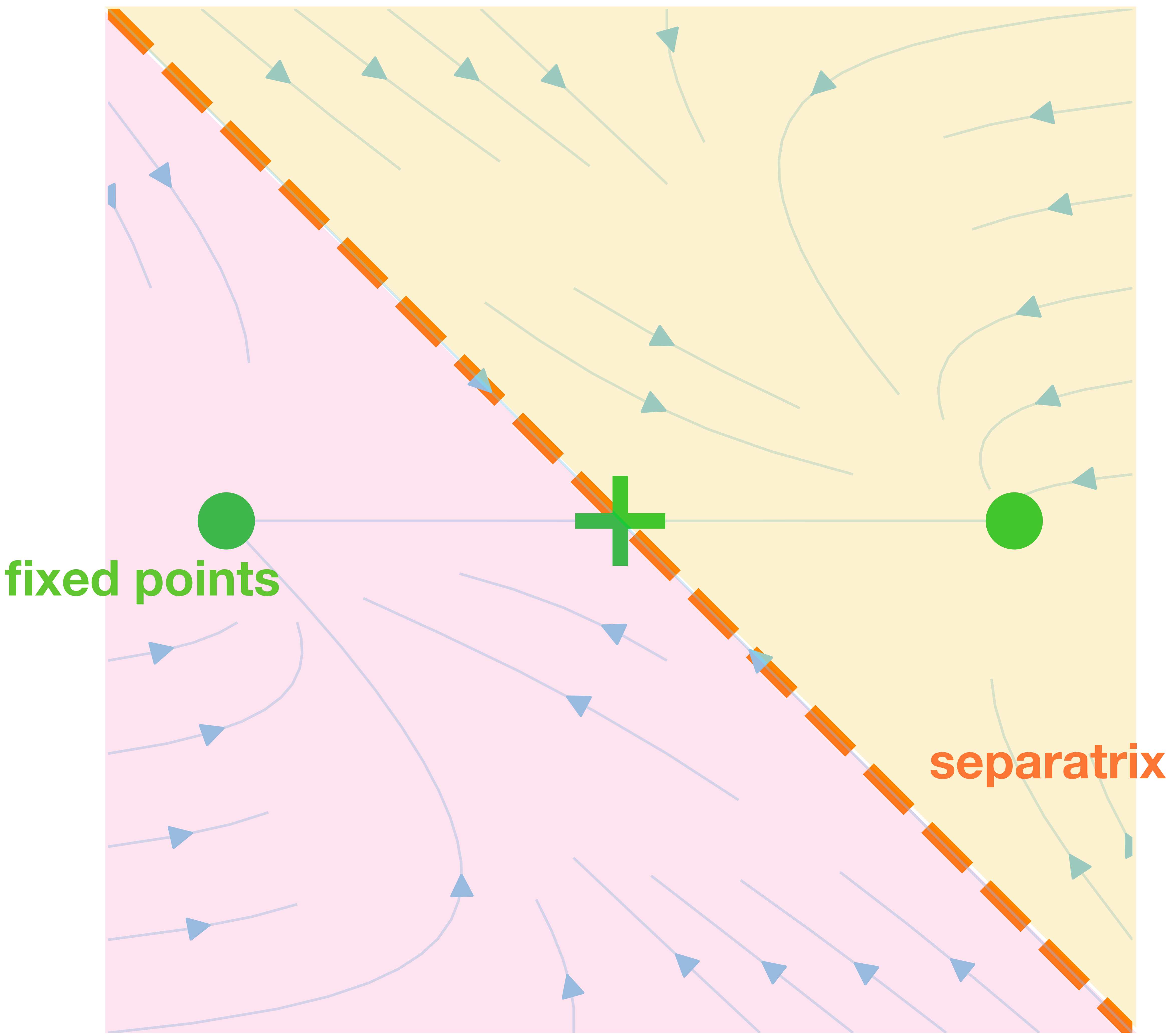
$$\dot{x} = f(x)$$



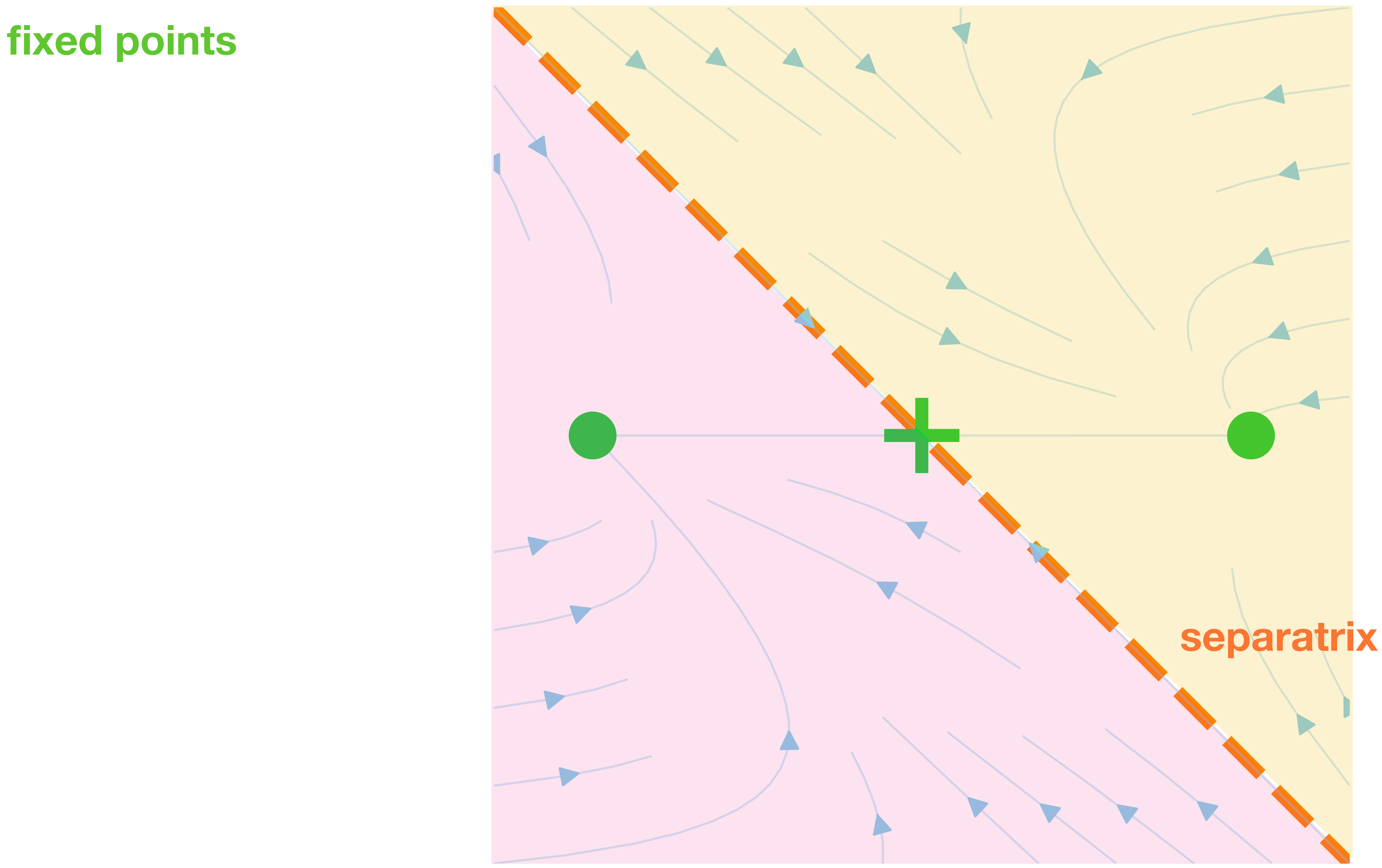
$$\dot{x} = f(x)$$



$$\dot{x} = f(x)$$



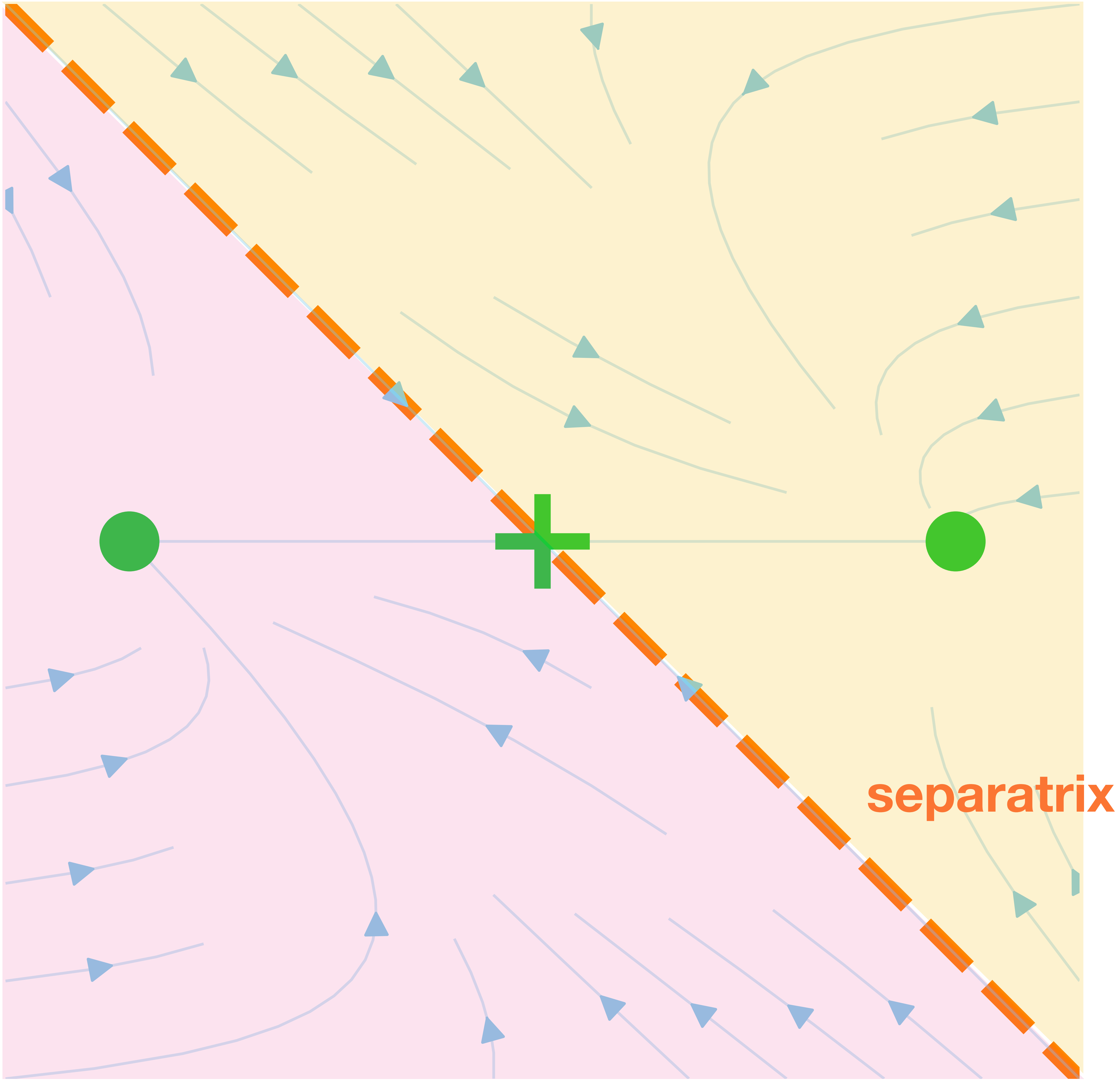
$$\dot{x} = f(x)$$



neighborhood is linear, easier to analyse

fixed points

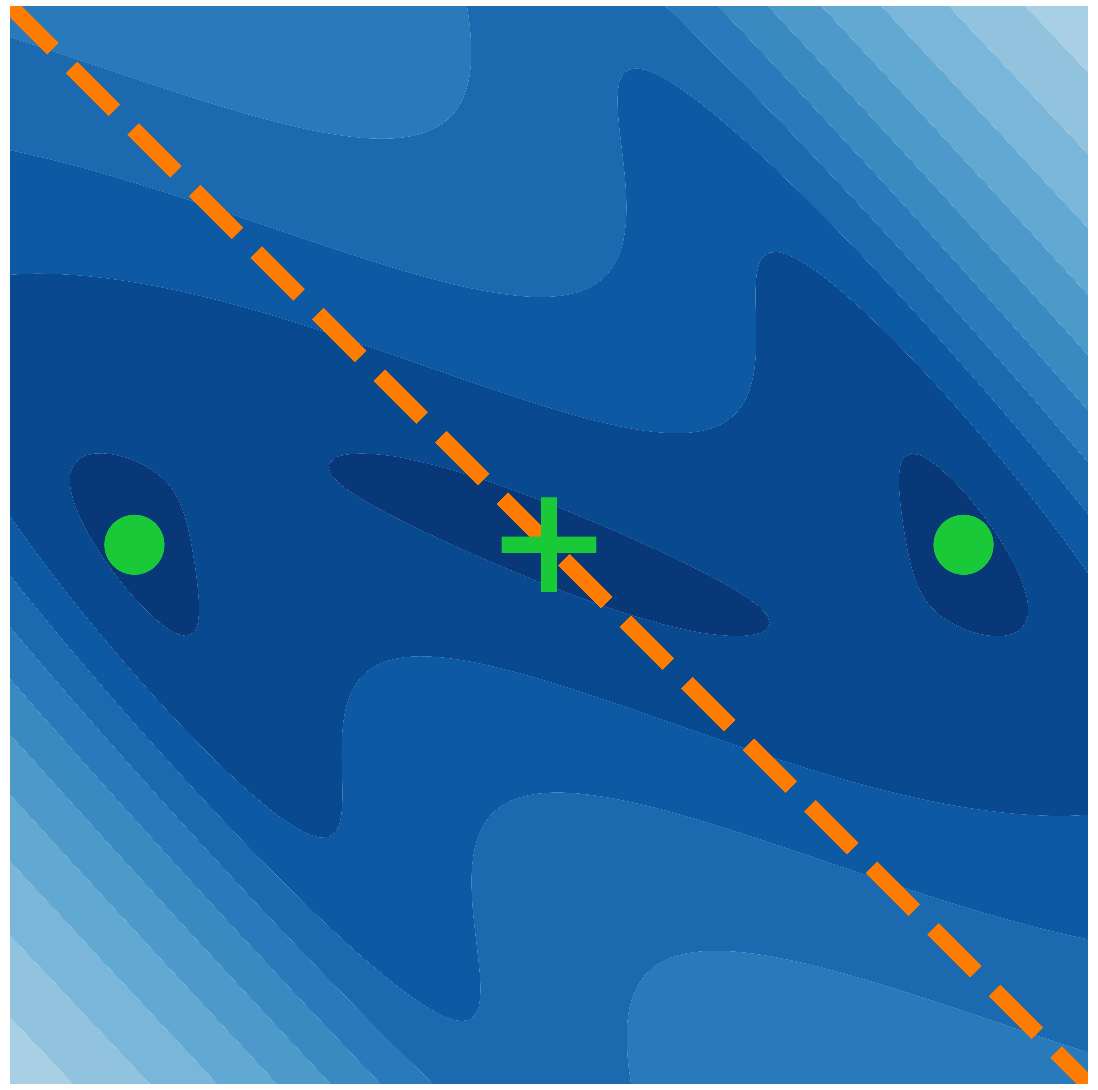
$$\dot{x} = f(x)$$



neighborhood is linear, easier to analyse

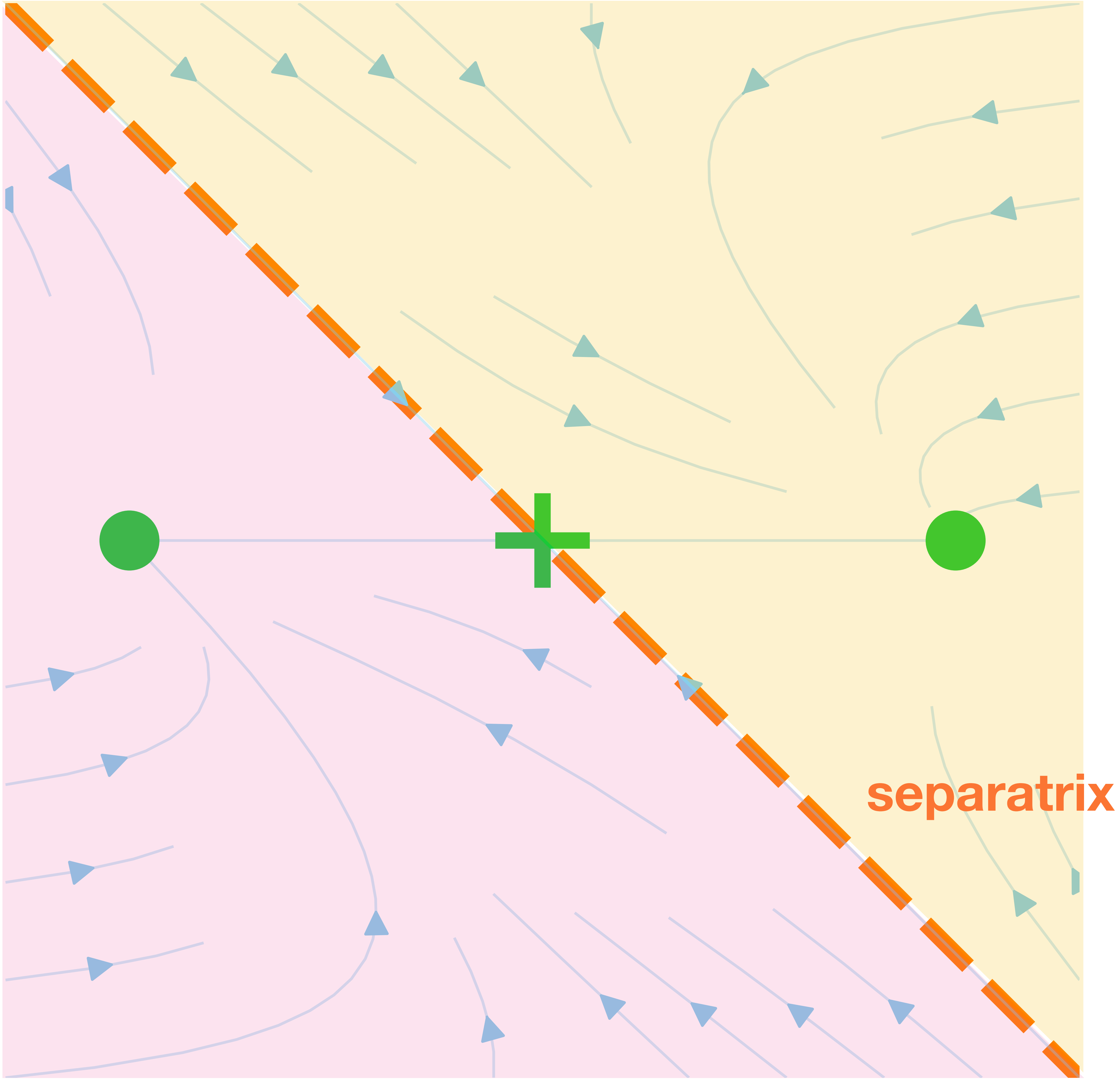
fixed points

$$q(x) := \|f(x)\|^2$$



Sussillo and Barak 2013

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$

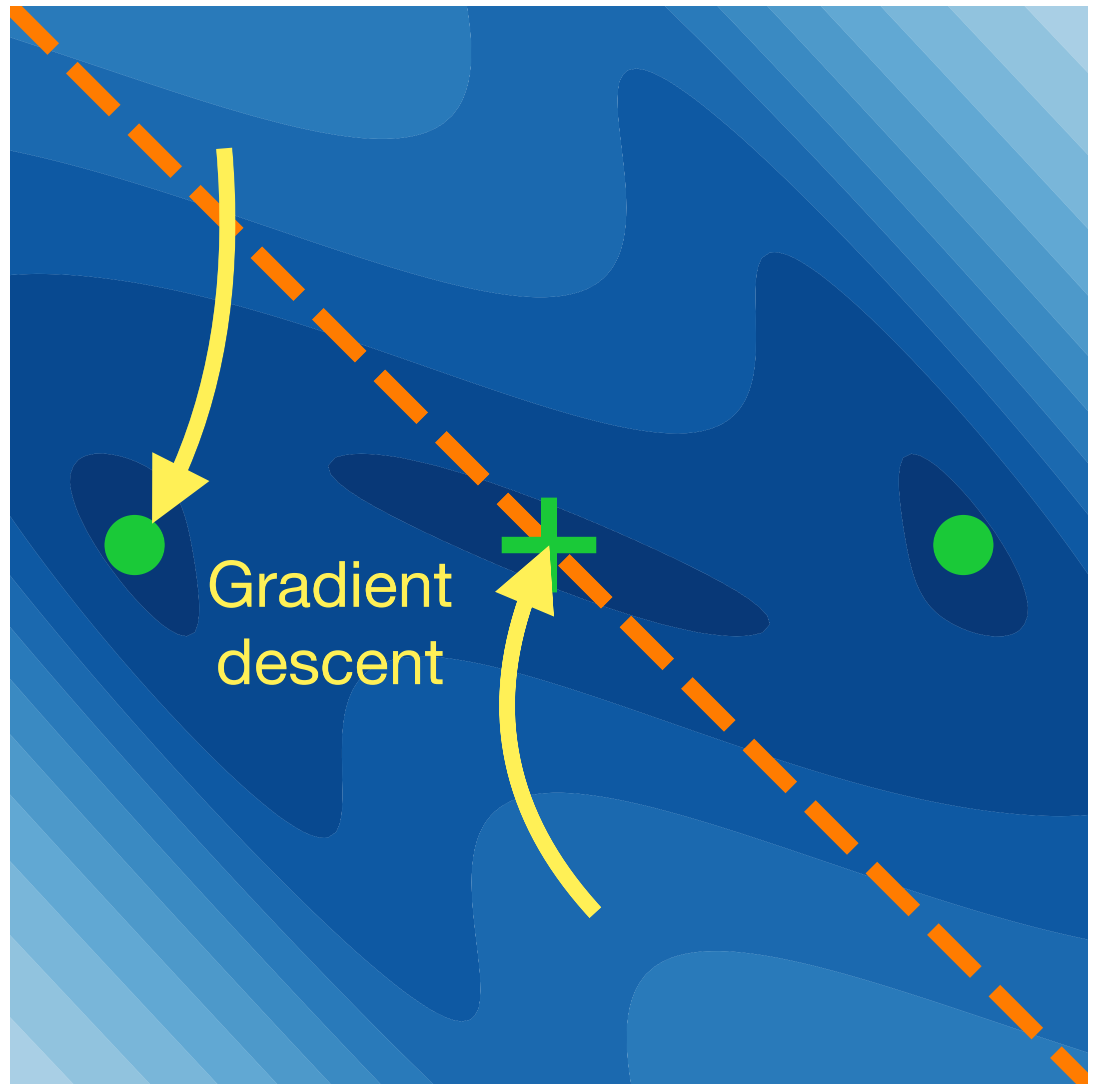


neighborhood is linear, easier to analyse

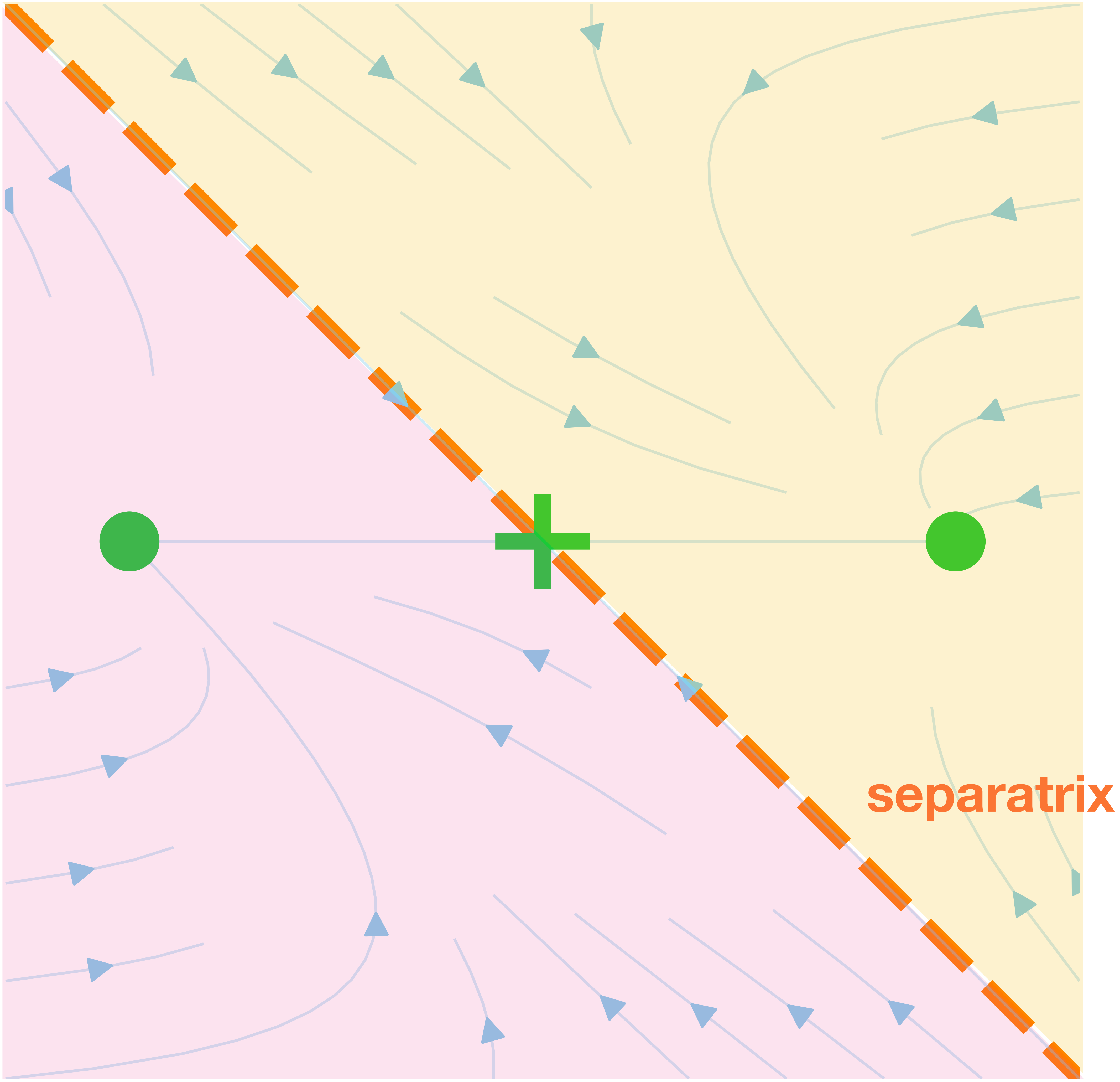
$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$

fixed points

$$q(x) := \|f(x)\|^2$$



Sussillo and Barak 2013

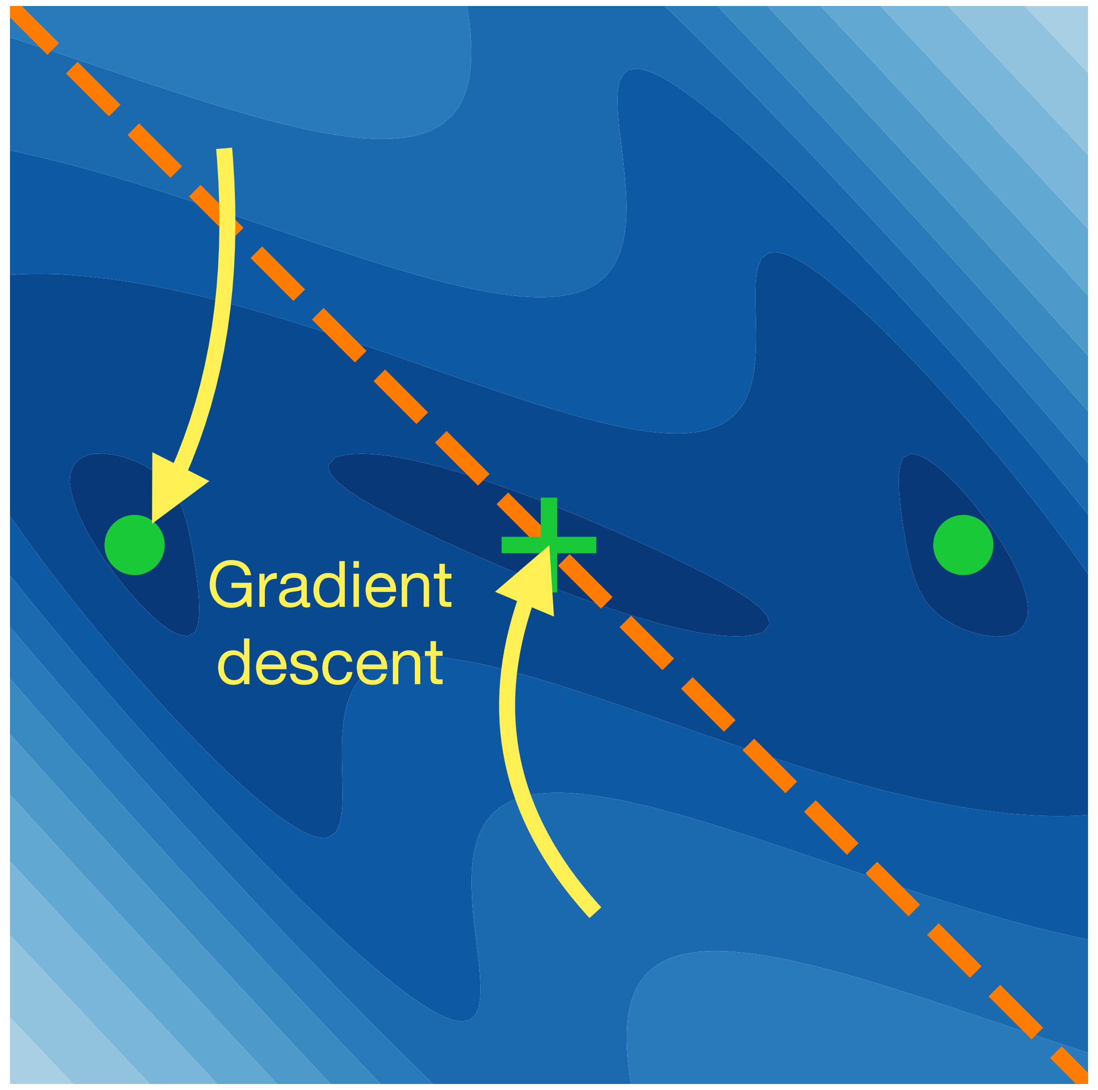


neighborhood is linear, easier to analyse

$$\dot{x} = f(x)$$

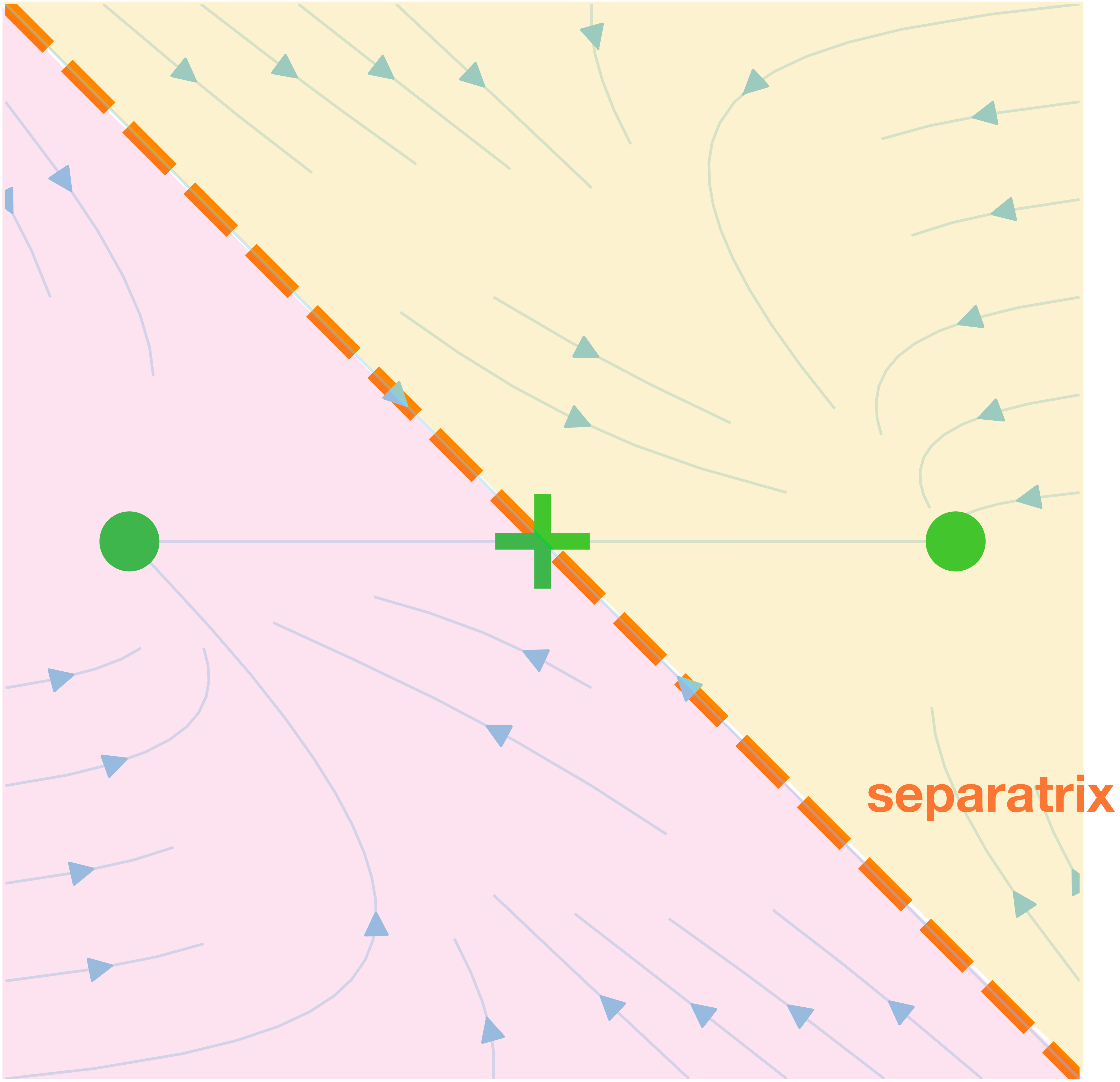
fixed points

$$q(x) := \|f(x)\|^2$$



Sussillo and Barak 2013

Works in high-dimensional Recurrent Neural Networks!

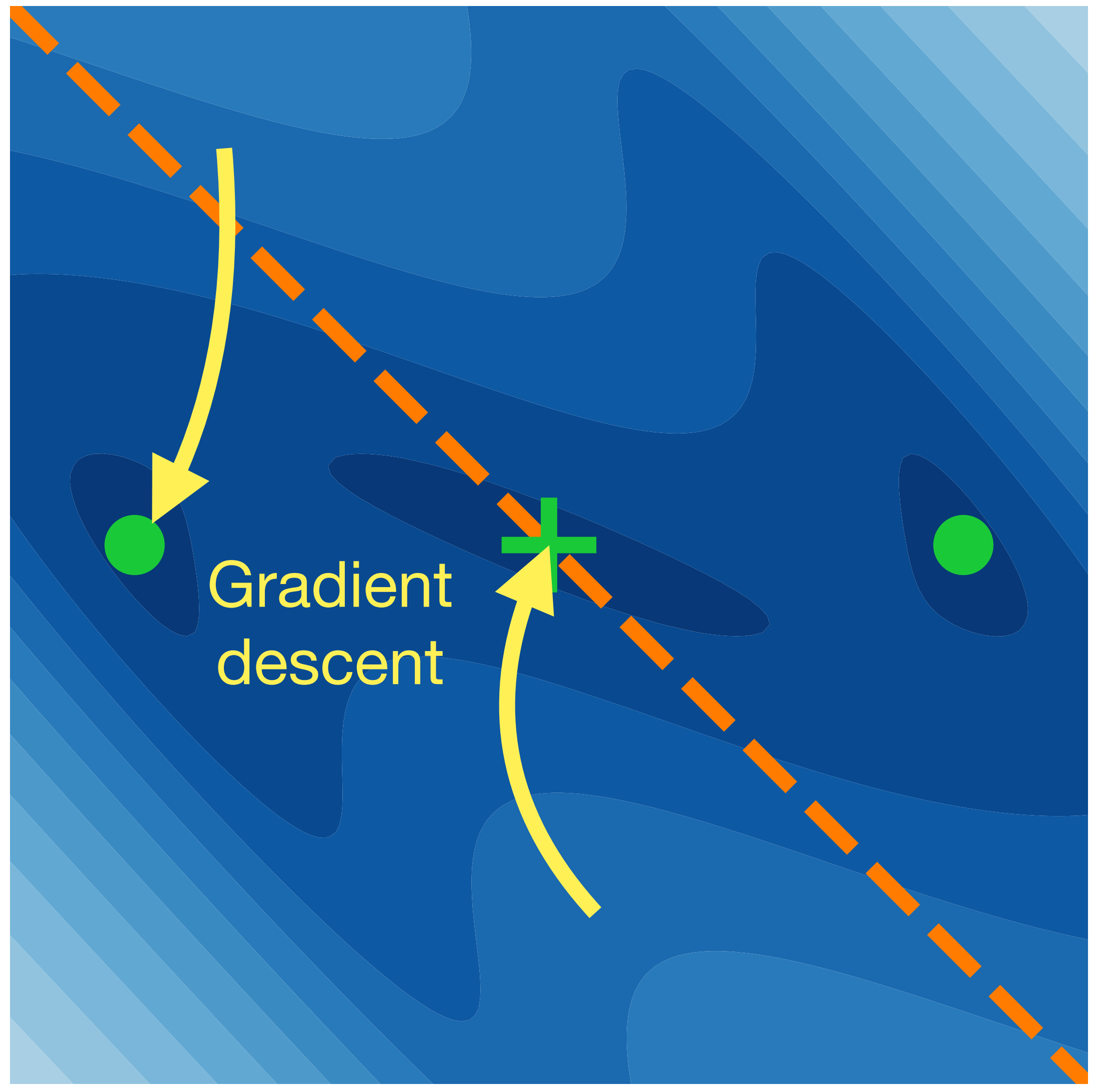


neighborhood is linear, easier to analyse

$$\dot{x} = f(x)$$

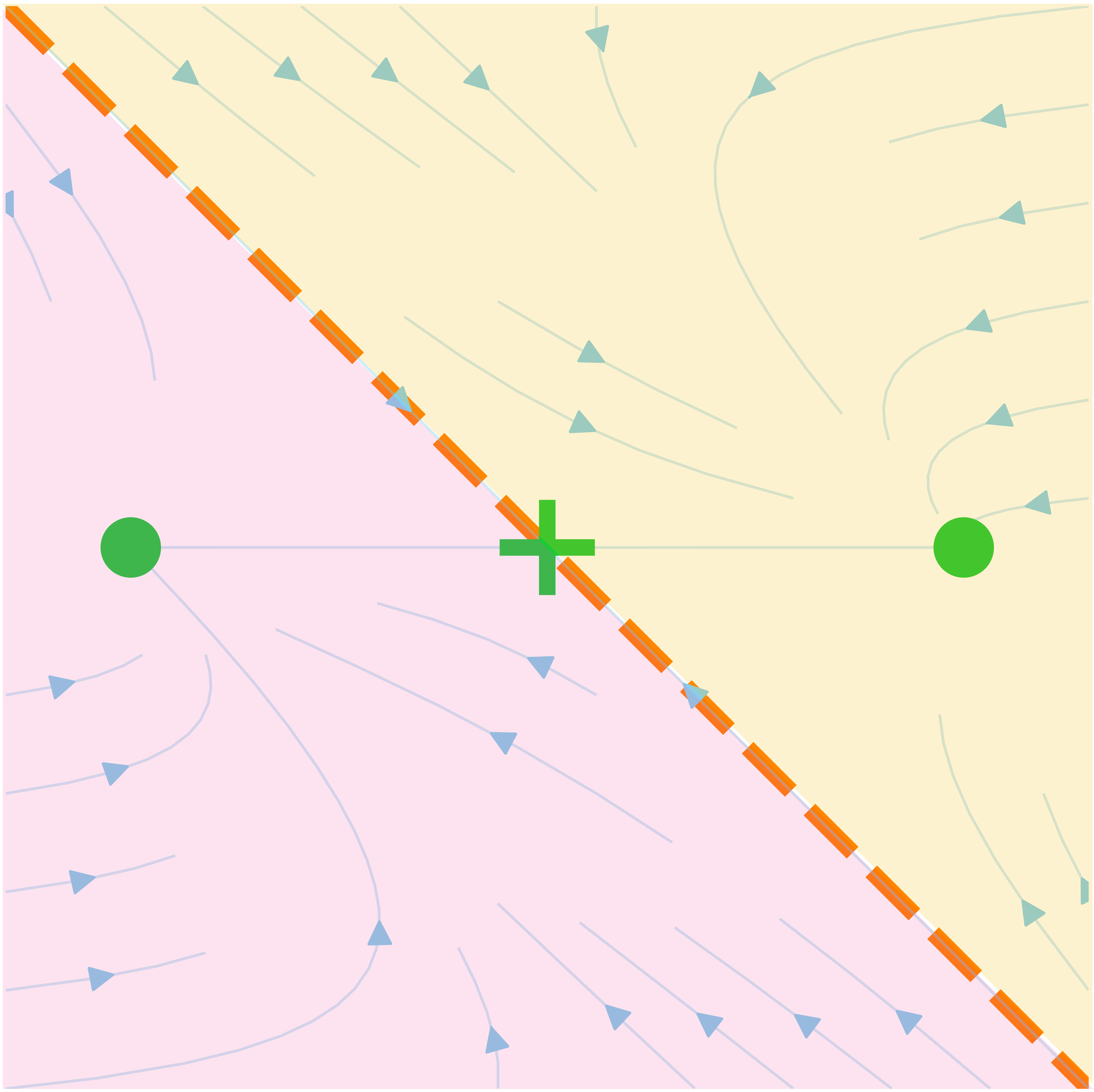
fixed points

$$q(x) := \|f(x)\|^2$$



Sussillo and Barak 2013

Works in high-dimensional Recurrent Neural Networks!



separatrix

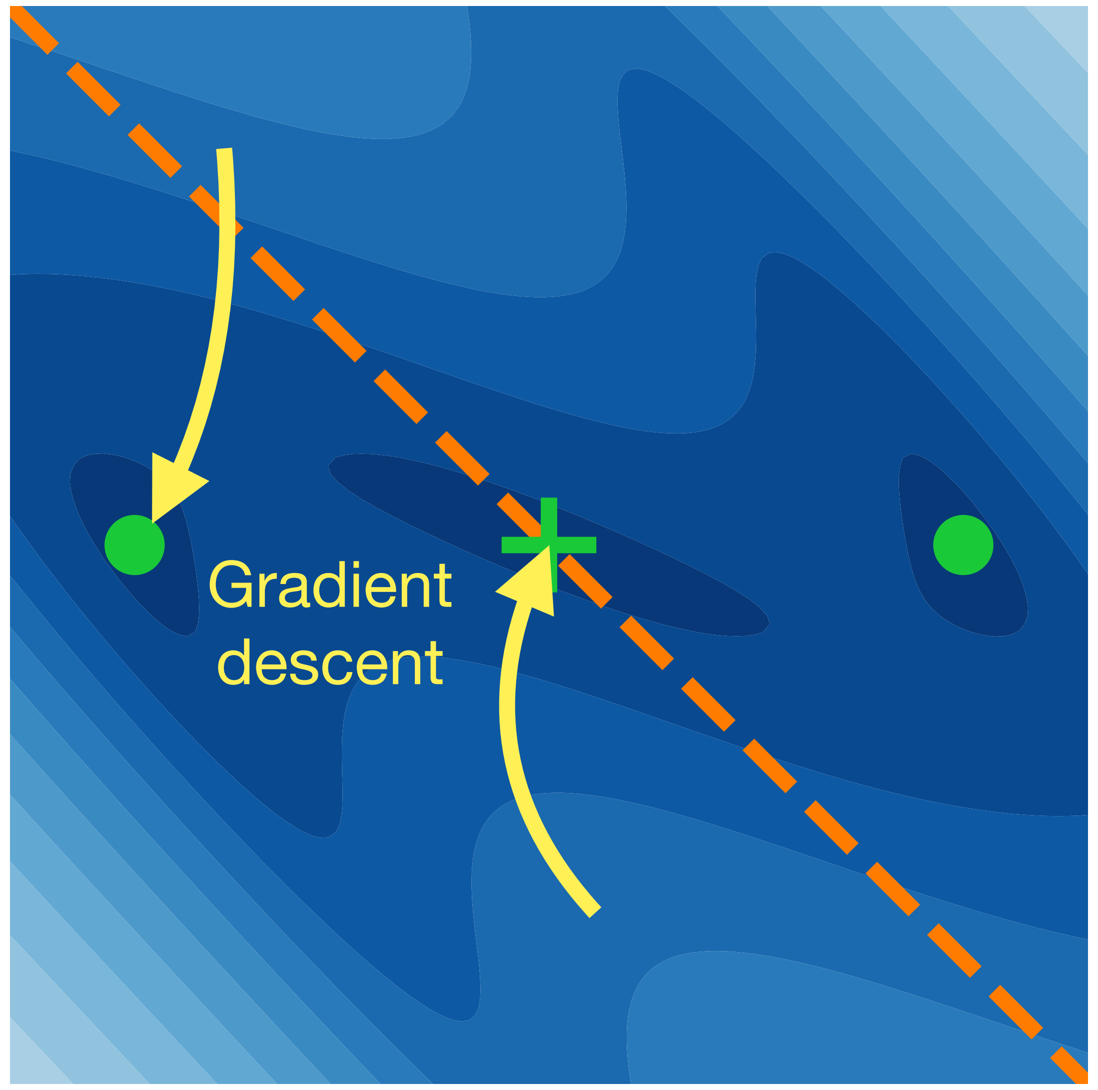
neighborhood is linear, easier to analyse

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$

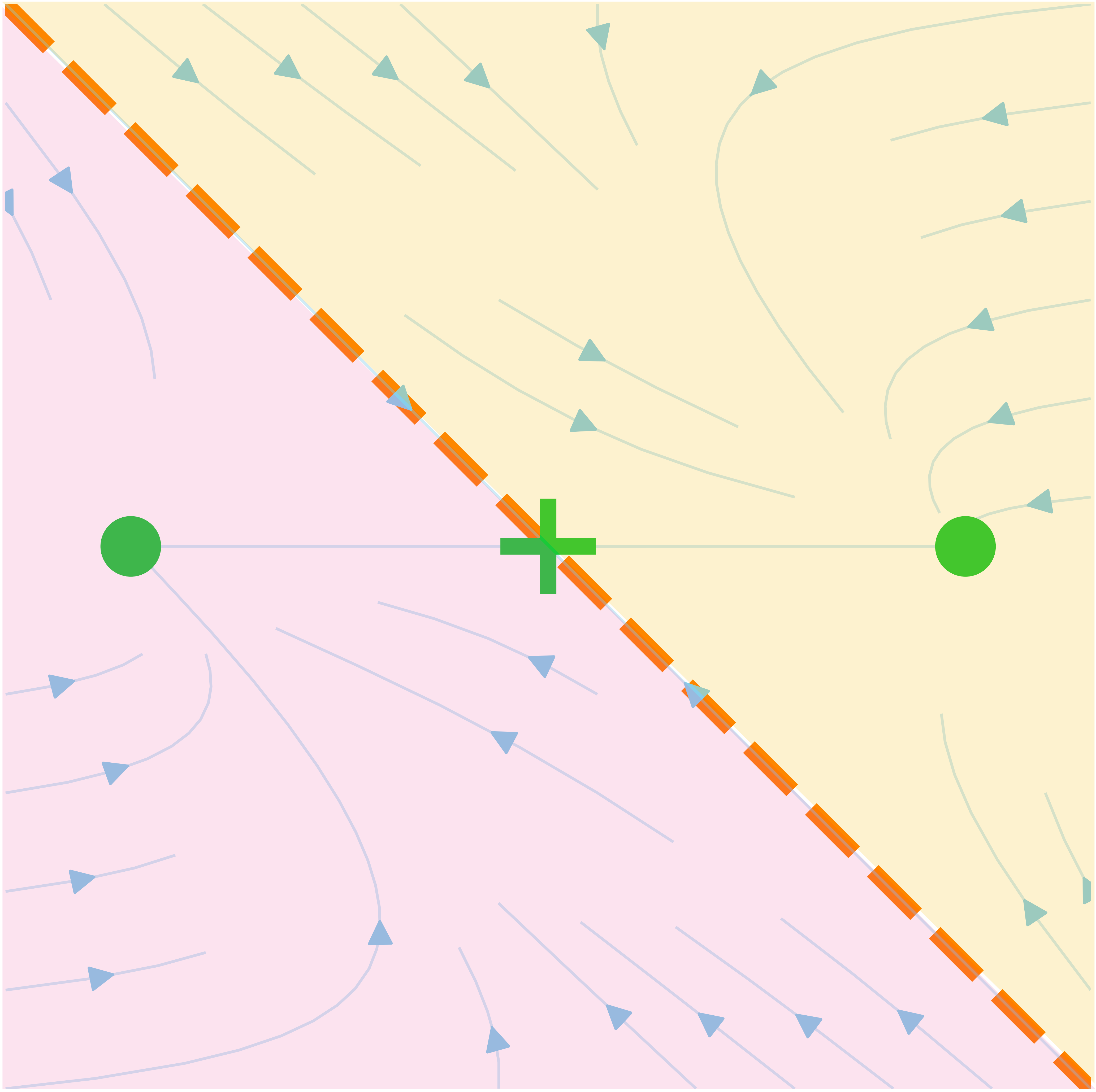
decision boundaries, optimal perturbations

fixed points

$$q(x) := \|f(x)\|^2$$



Sussillo and Barak 2013



separatrix

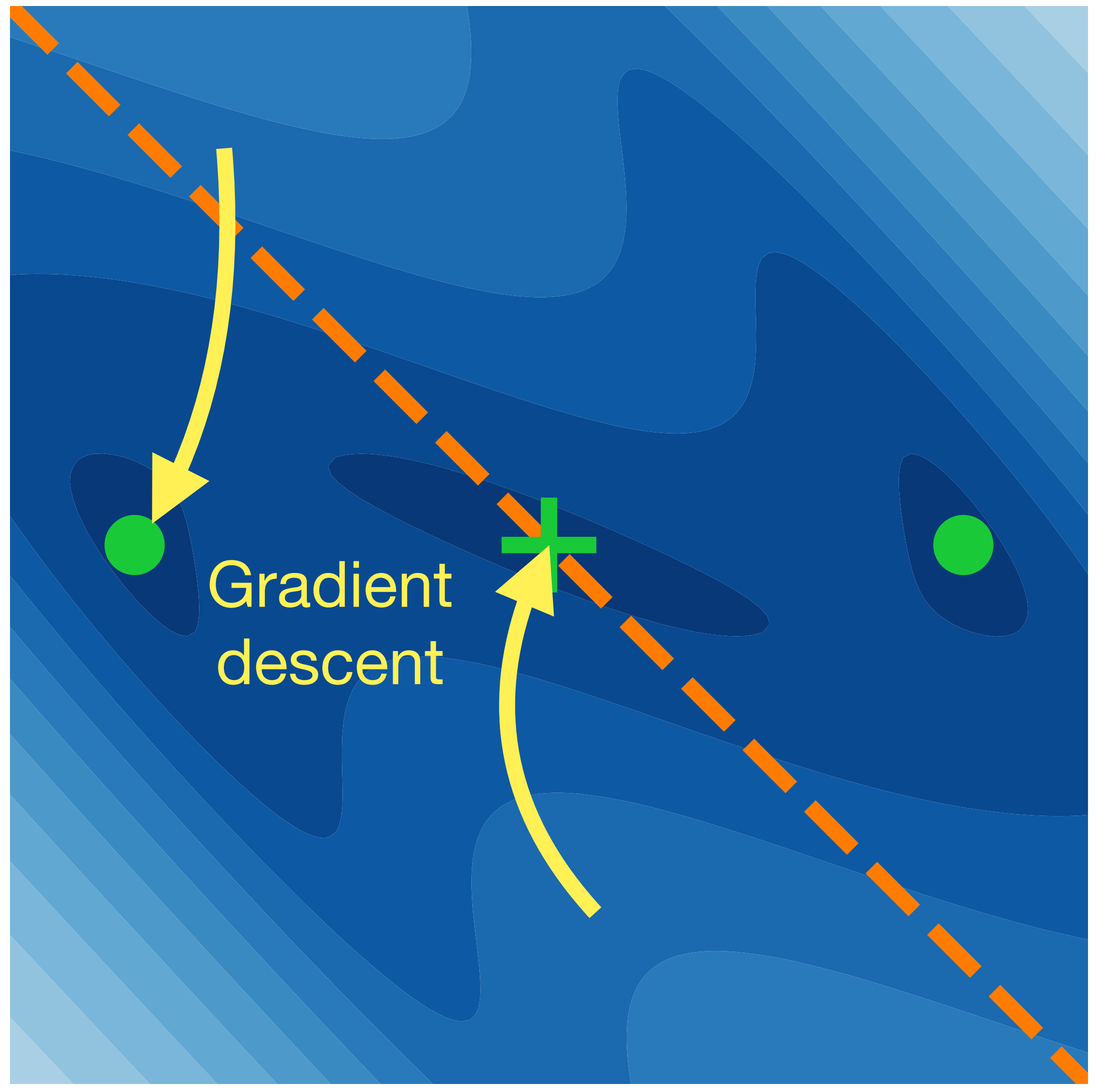
Works in high-dimensional Recurrent Neural Networks!

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$

neighborhood is linear, easier to analyse

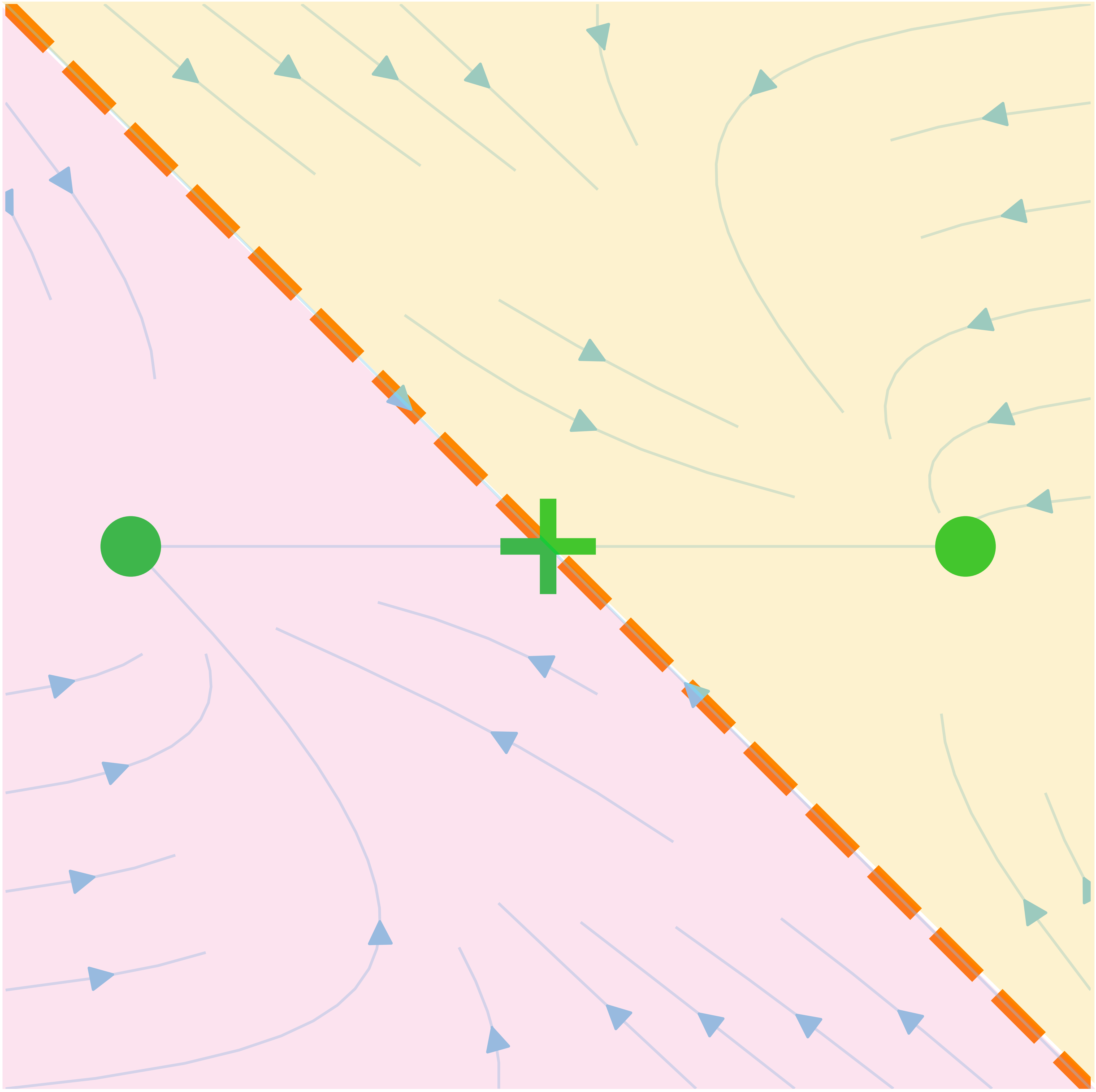
fixed points

$$q(x) := \|f(x)\|^2$$



Sussillo and Barak 2013

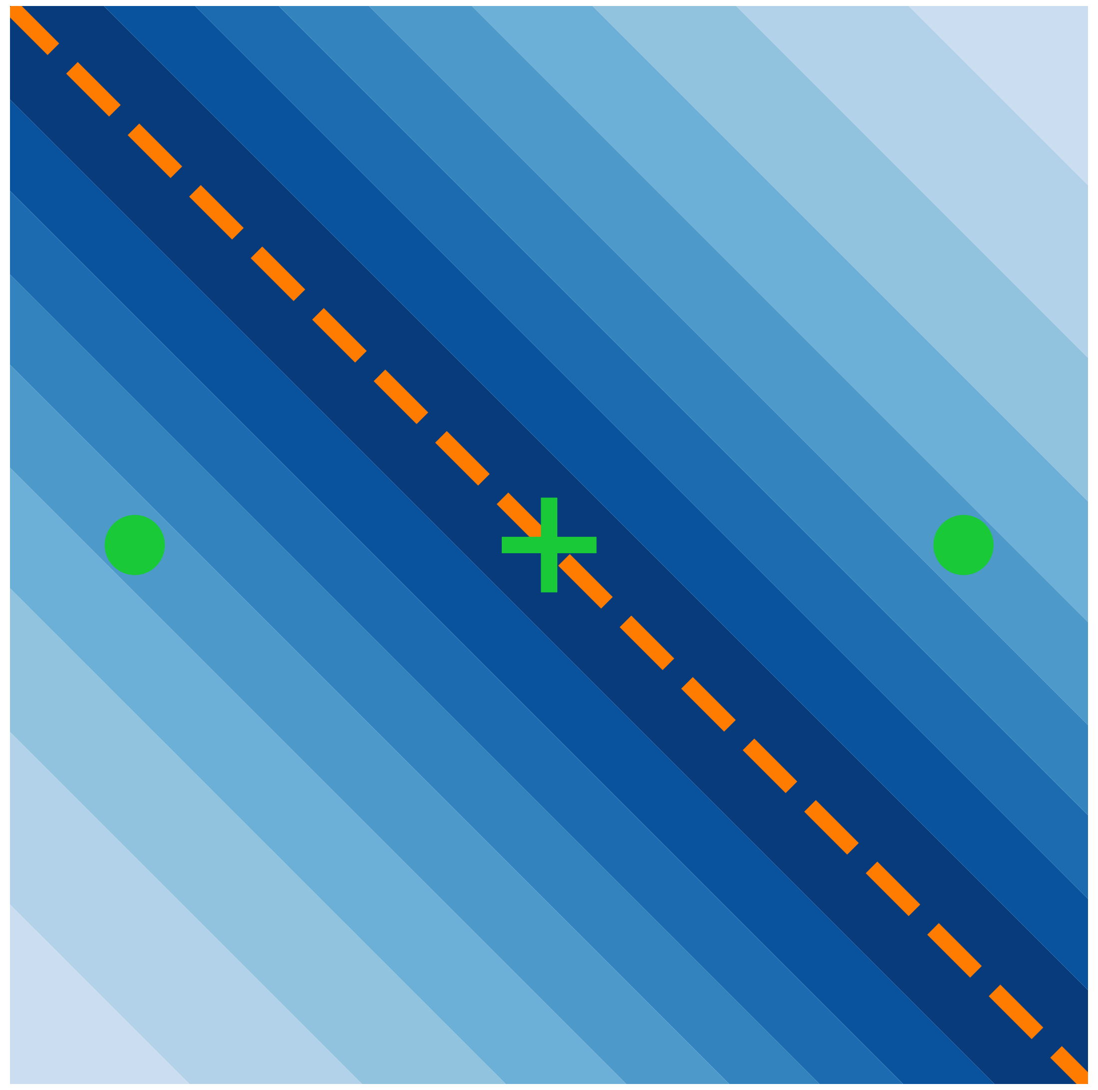
Works in high-dimensional
Recurrent Neural Networks!



decision boundaries, optimal perturbations

separatrix

$$\psi(x) := ?$$



our work

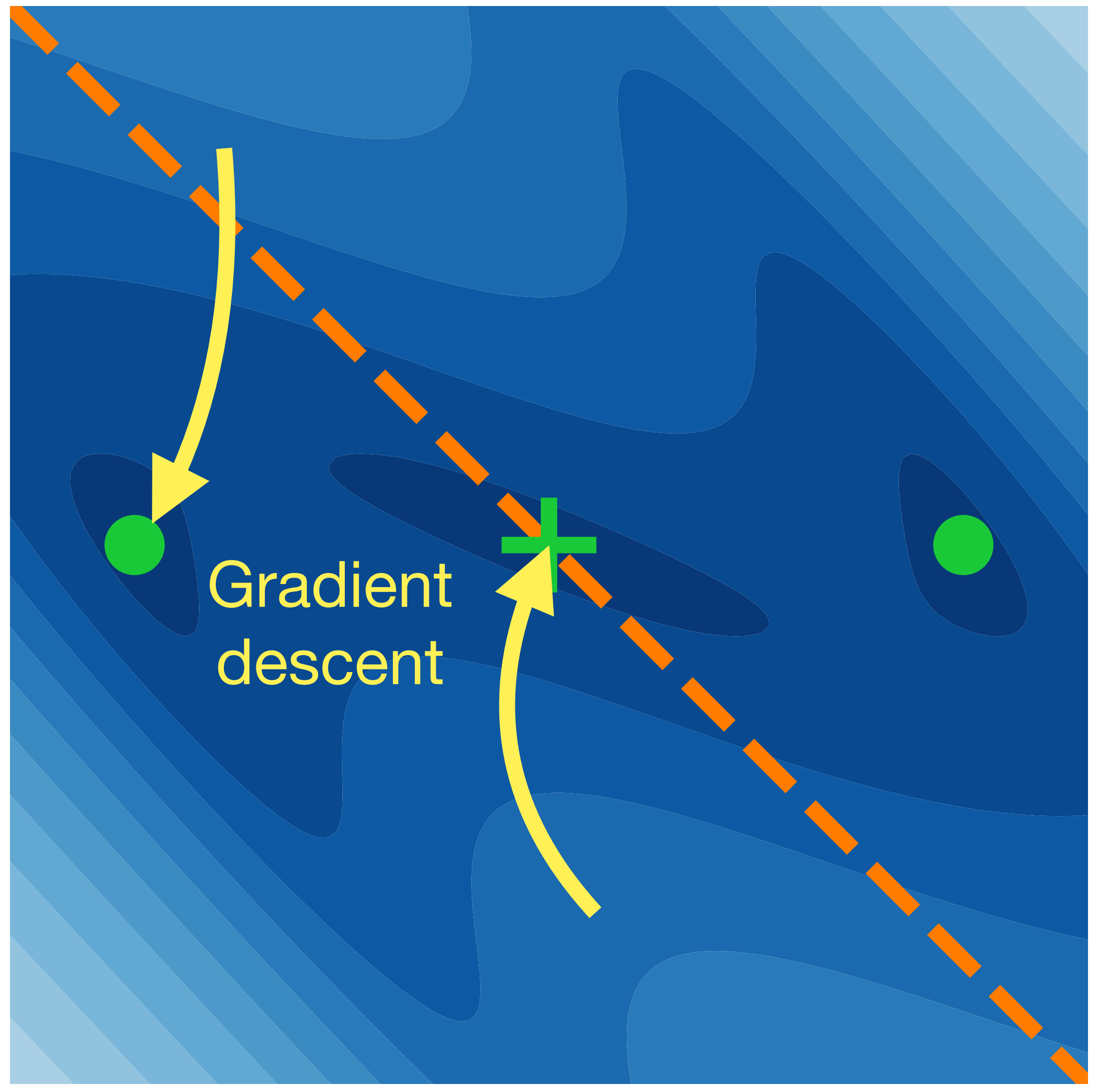
neighborhood is linear, easier to analyse

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$

decision boundaries, optimal perturbations

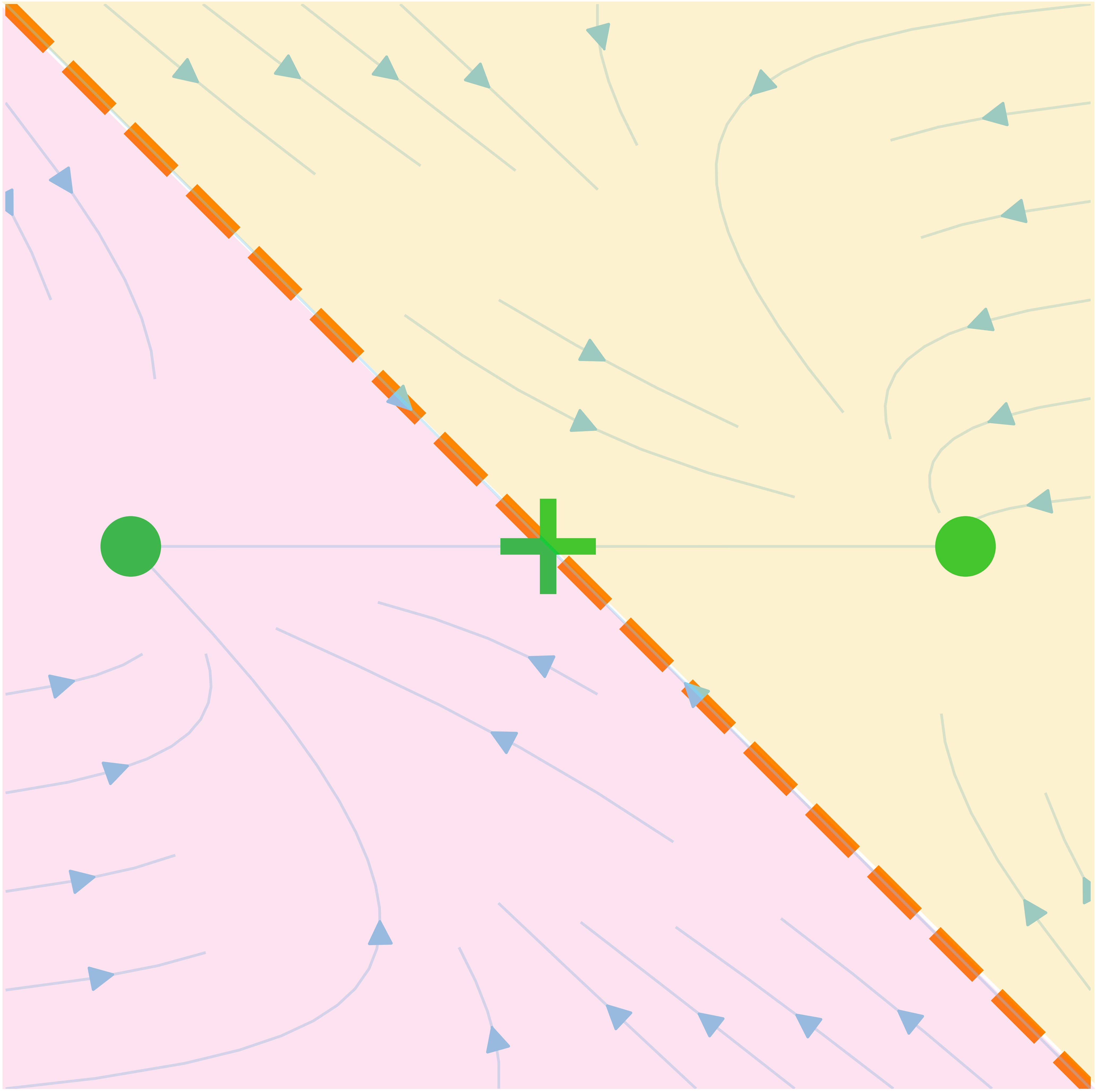
fixed points

$$q(x) := \|f(x)\|^2$$



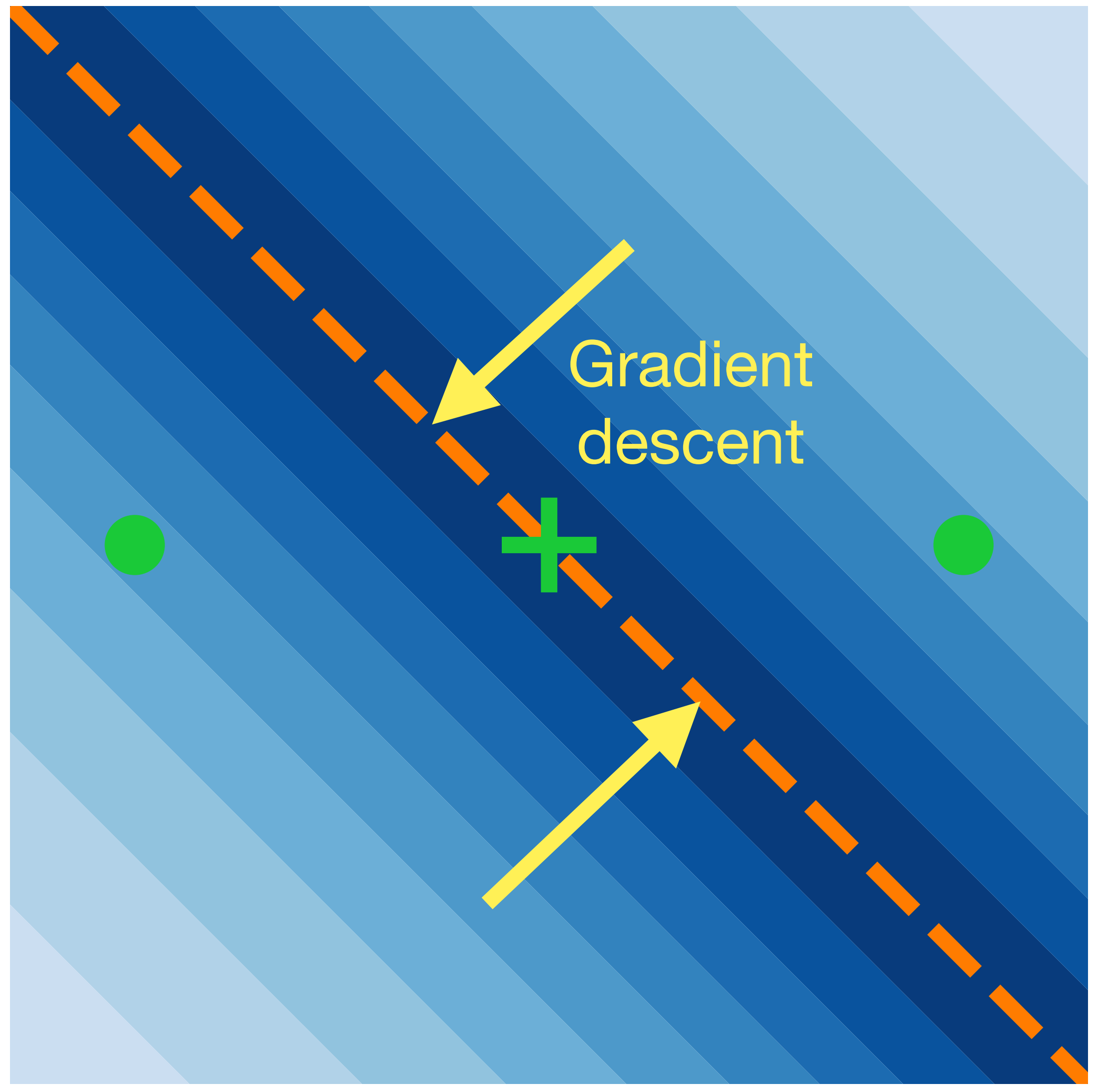
Sussillo and Barak 2013

Works in high-dimensional
Recurrent Neural Networks!



separatrix

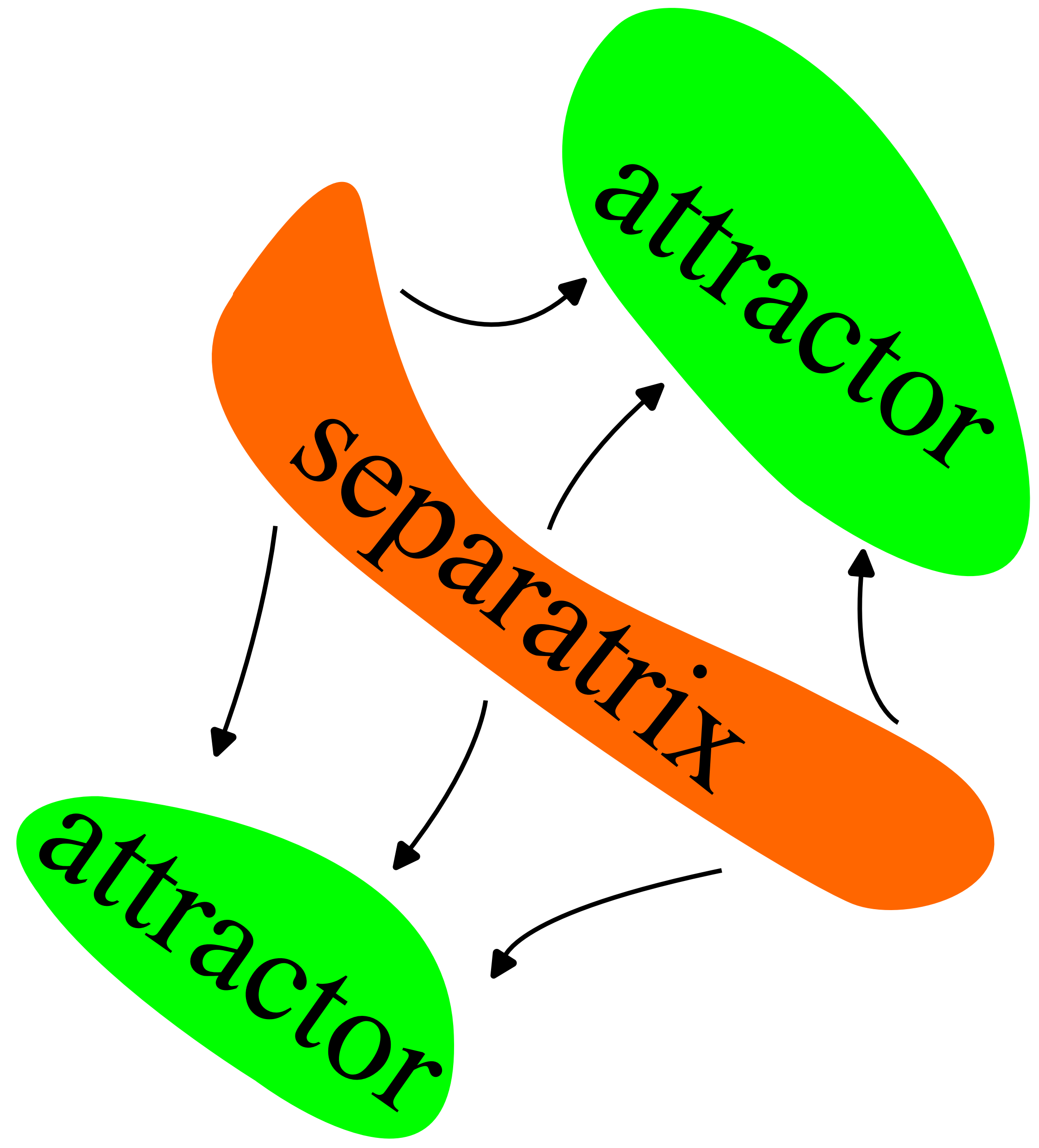
$$\psi(x) := ?$$



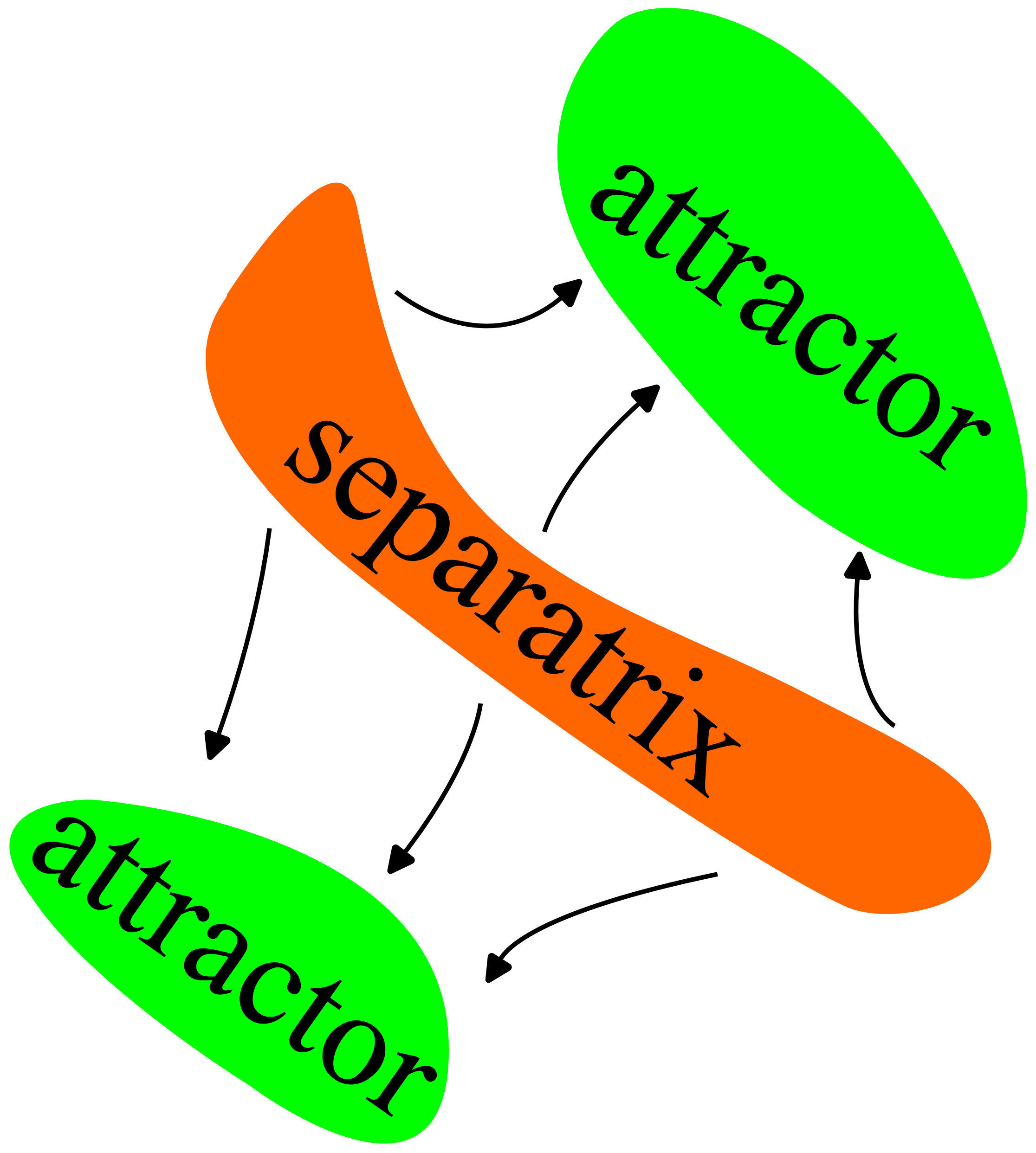
our work

How do we find such a function?

How do we find such a function?

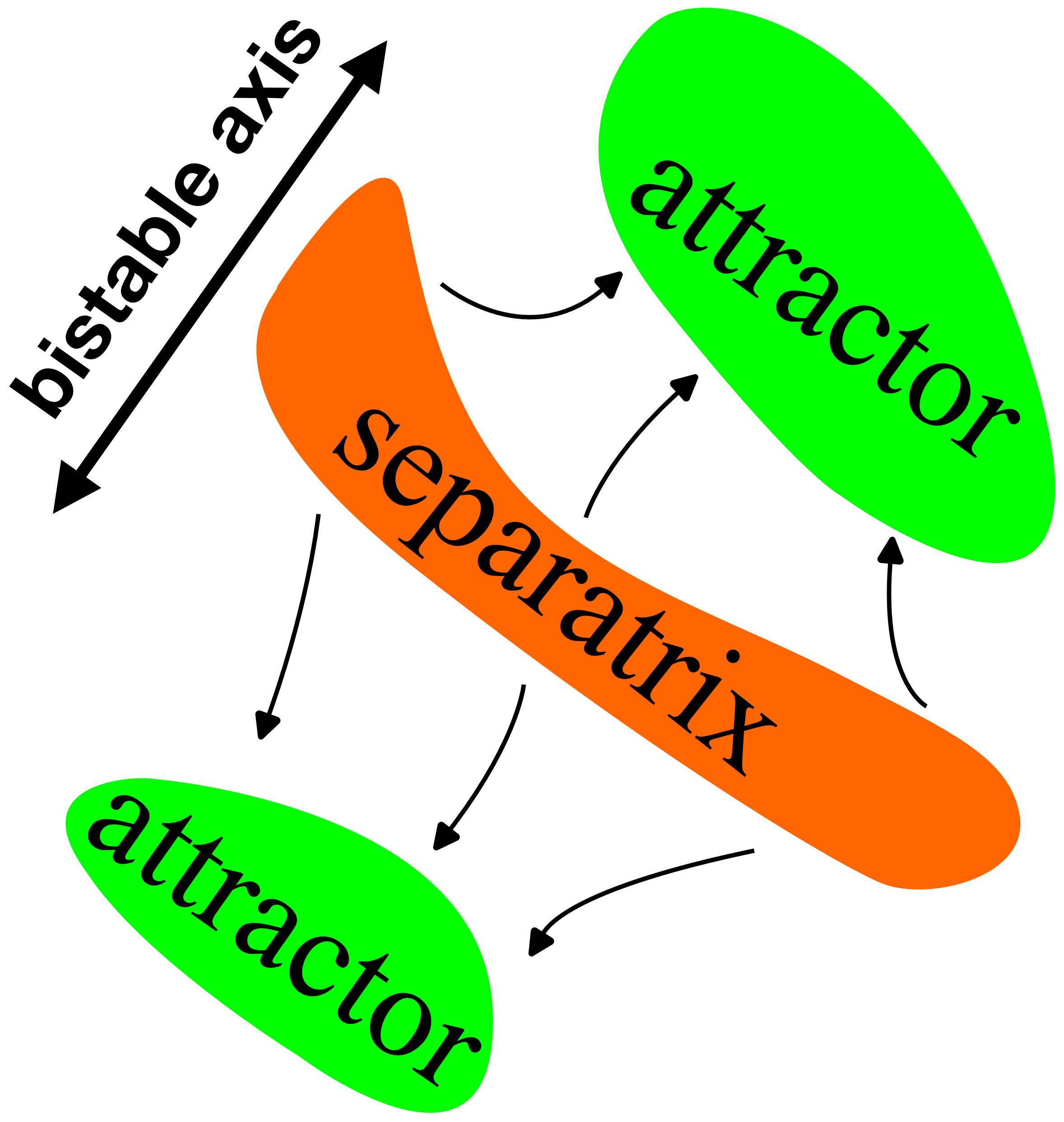


How do we find such a function?



$$\dot{x} = f(x)$$
$$x \in \mathcal{X}$$

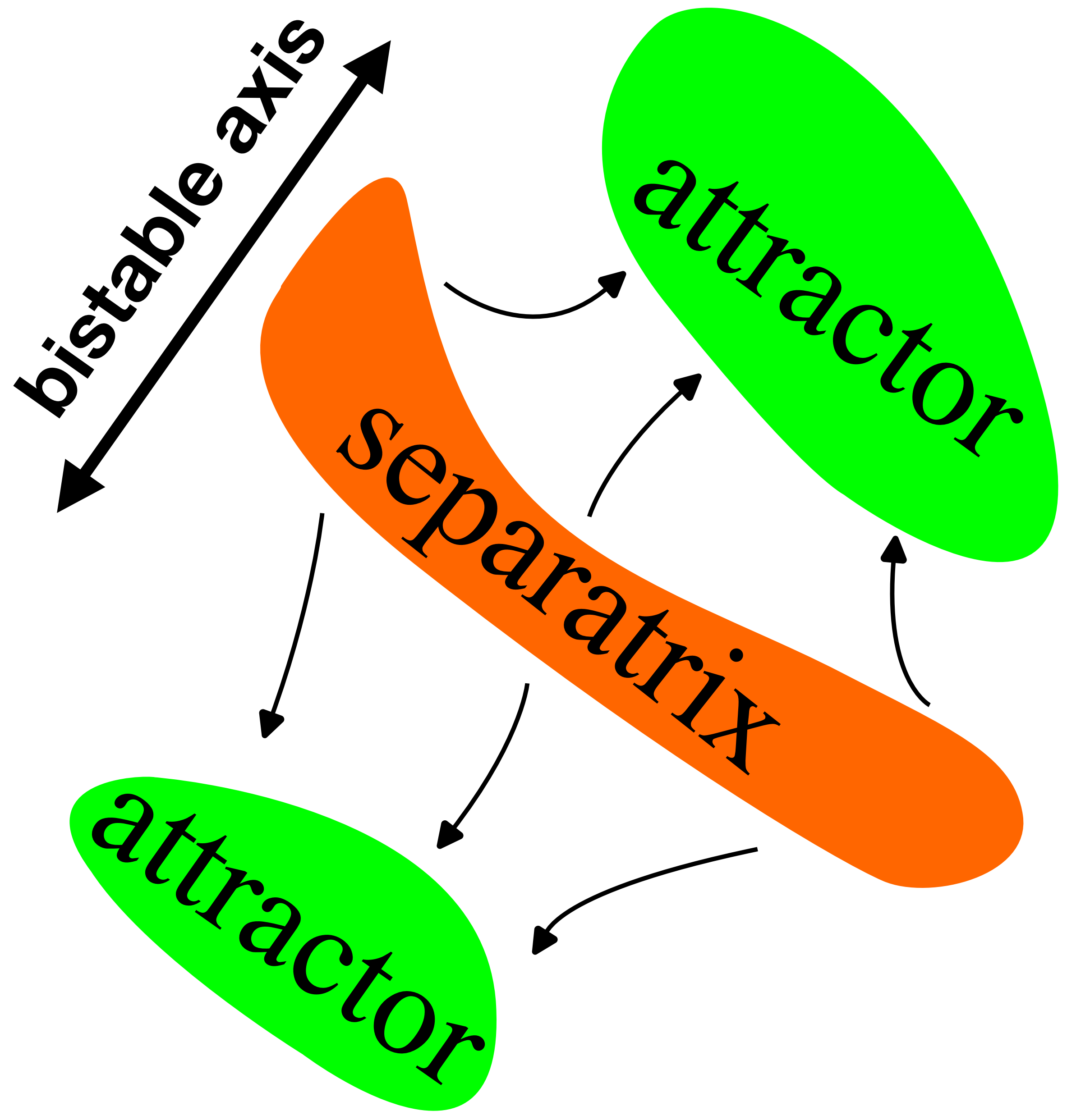
How do we find such a function?



$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$

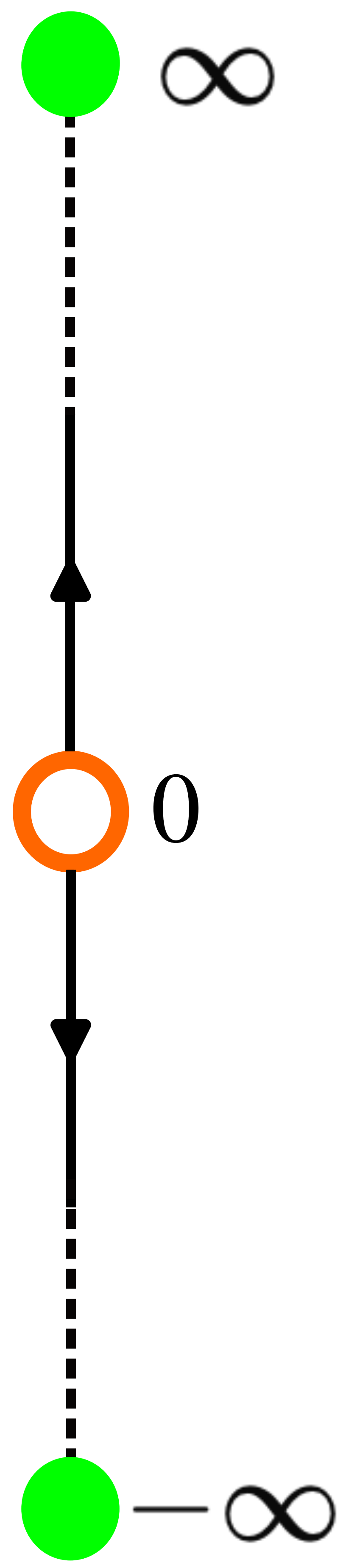
$$\boldsymbol{x} \in \mathcal{X}$$

How do we find such a function?

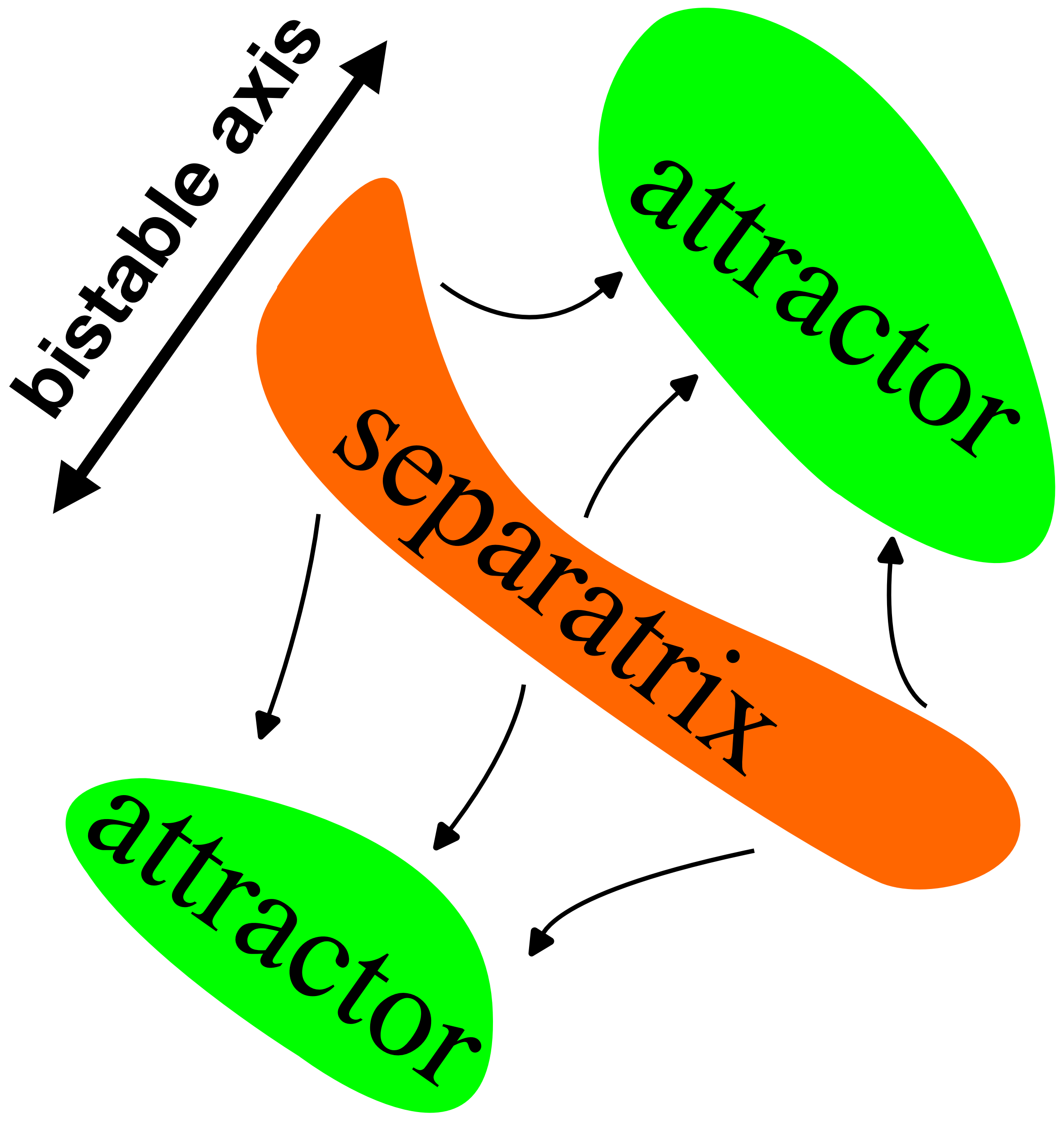


$$\dot{x} = f(x)$$
$$x \in \mathcal{X}$$

1D, linear, unstable dynamics

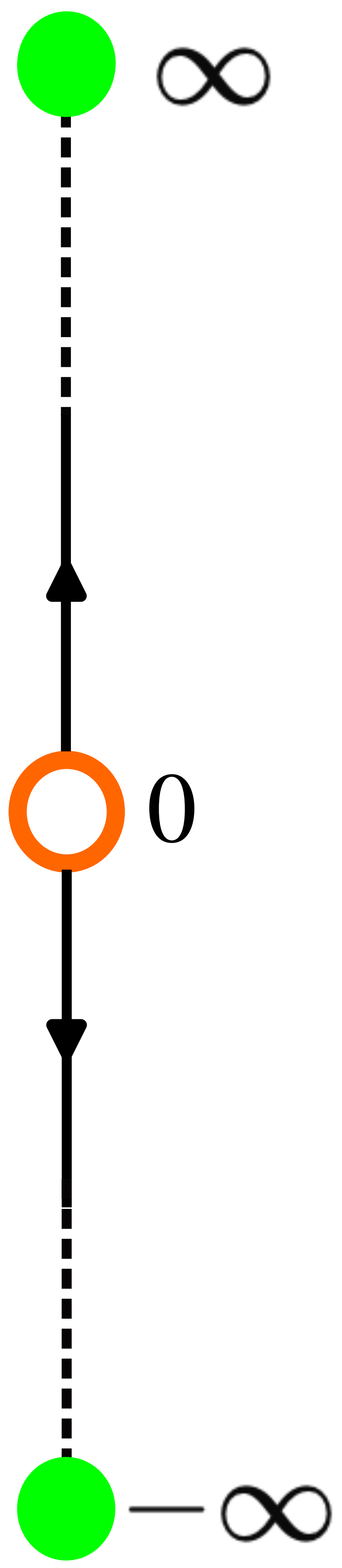


How do we find such a function?



$$\dot{x} = f(x)$$
$$x \in \mathcal{X}$$

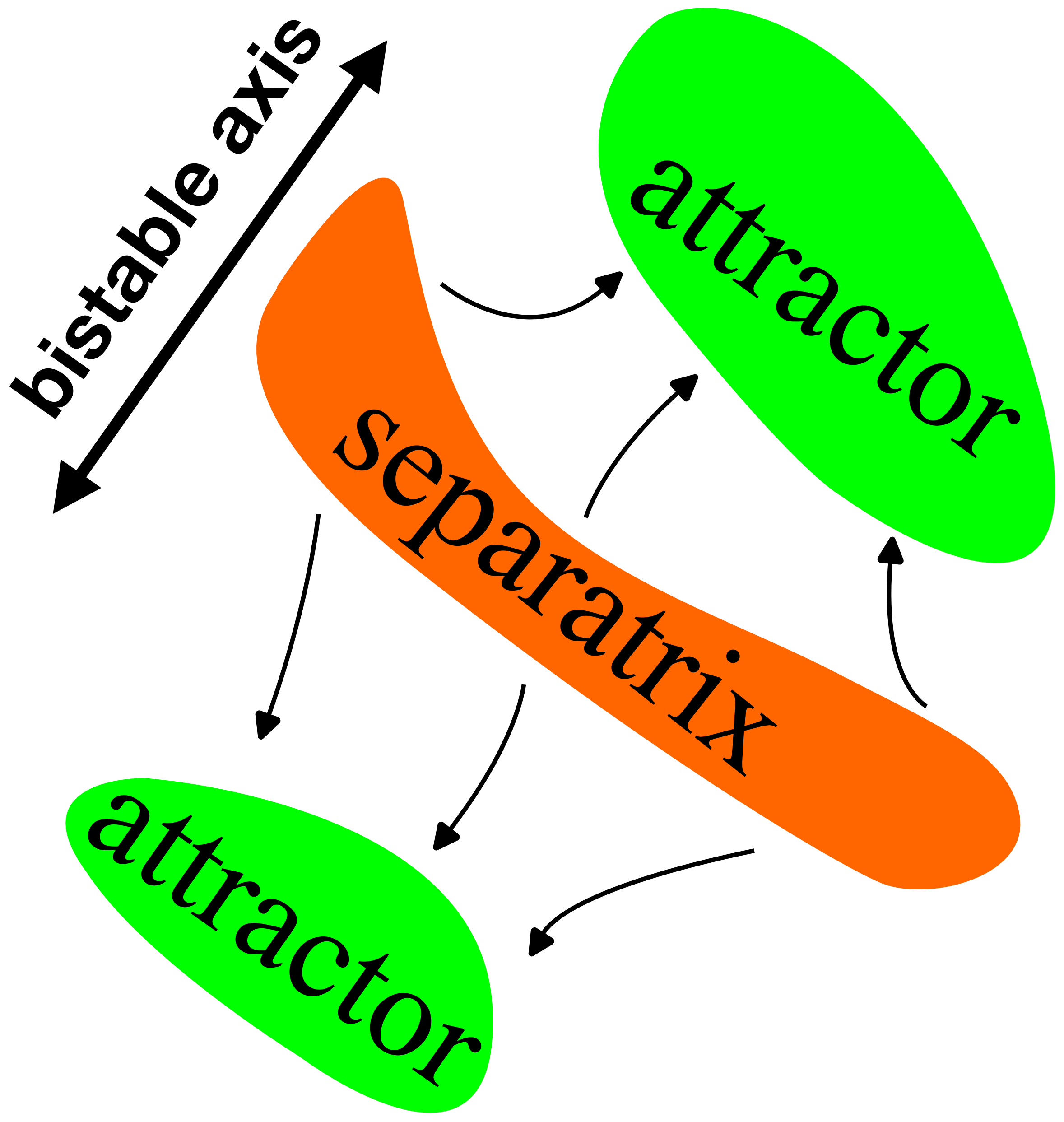
1D, linear, unstable dynamics

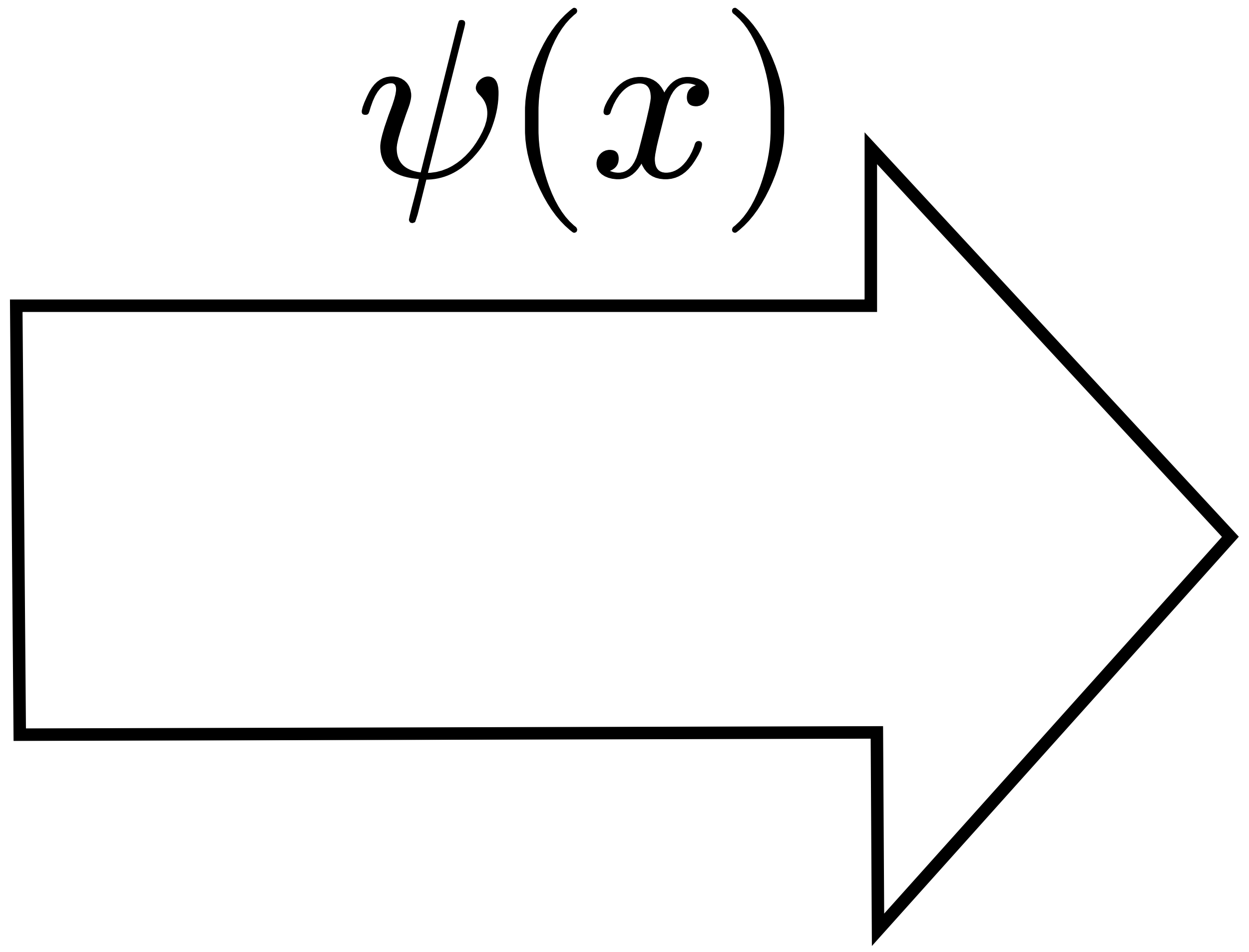


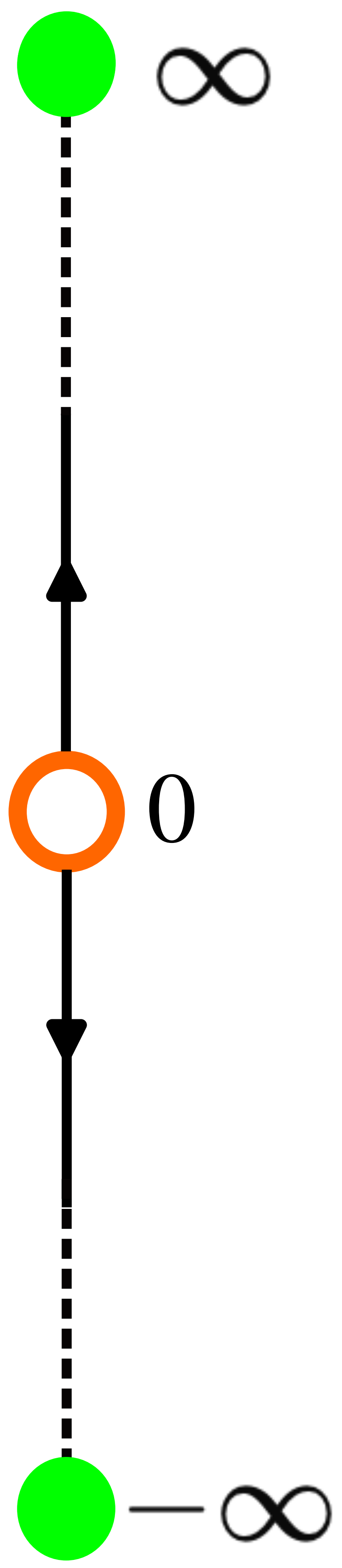
$$\dot{\psi} = \lambda \psi$$
$$\psi \in \mathbb{R}$$
$$\lambda > 0$$

How do we find such a function?

1D, linear, unstable dynamics





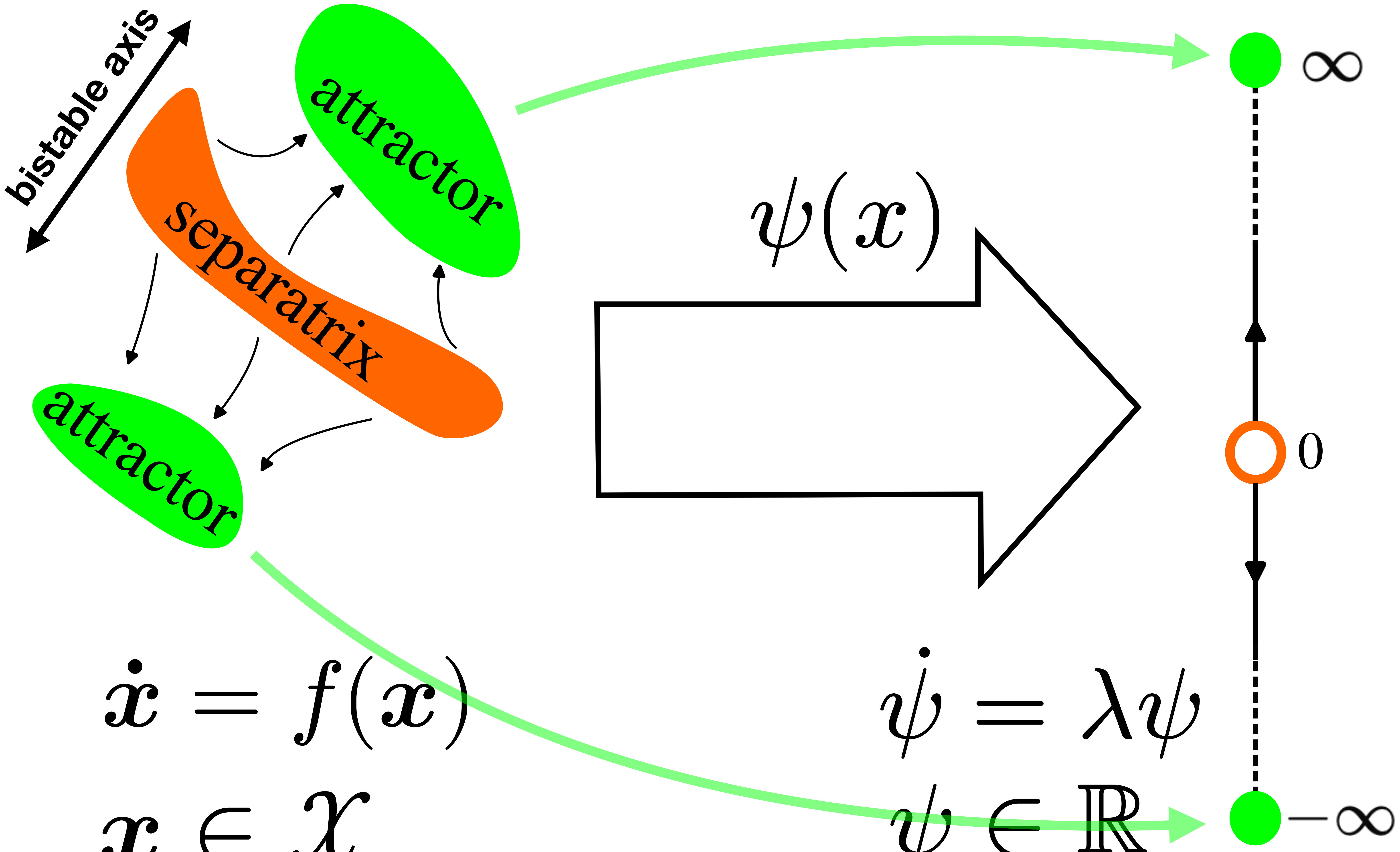


$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$
$$\boldsymbol{x} \in \mathcal{X}$$

$$\dot{\psi} = \lambda\psi$$
$$\psi \in \mathbb{R}$$
$$\lambda > 0$$

How do we find such a function?

1D, linear, unstable dynamics



$$\dot{x} = f(x)$$

$$x \in \mathcal{X}$$

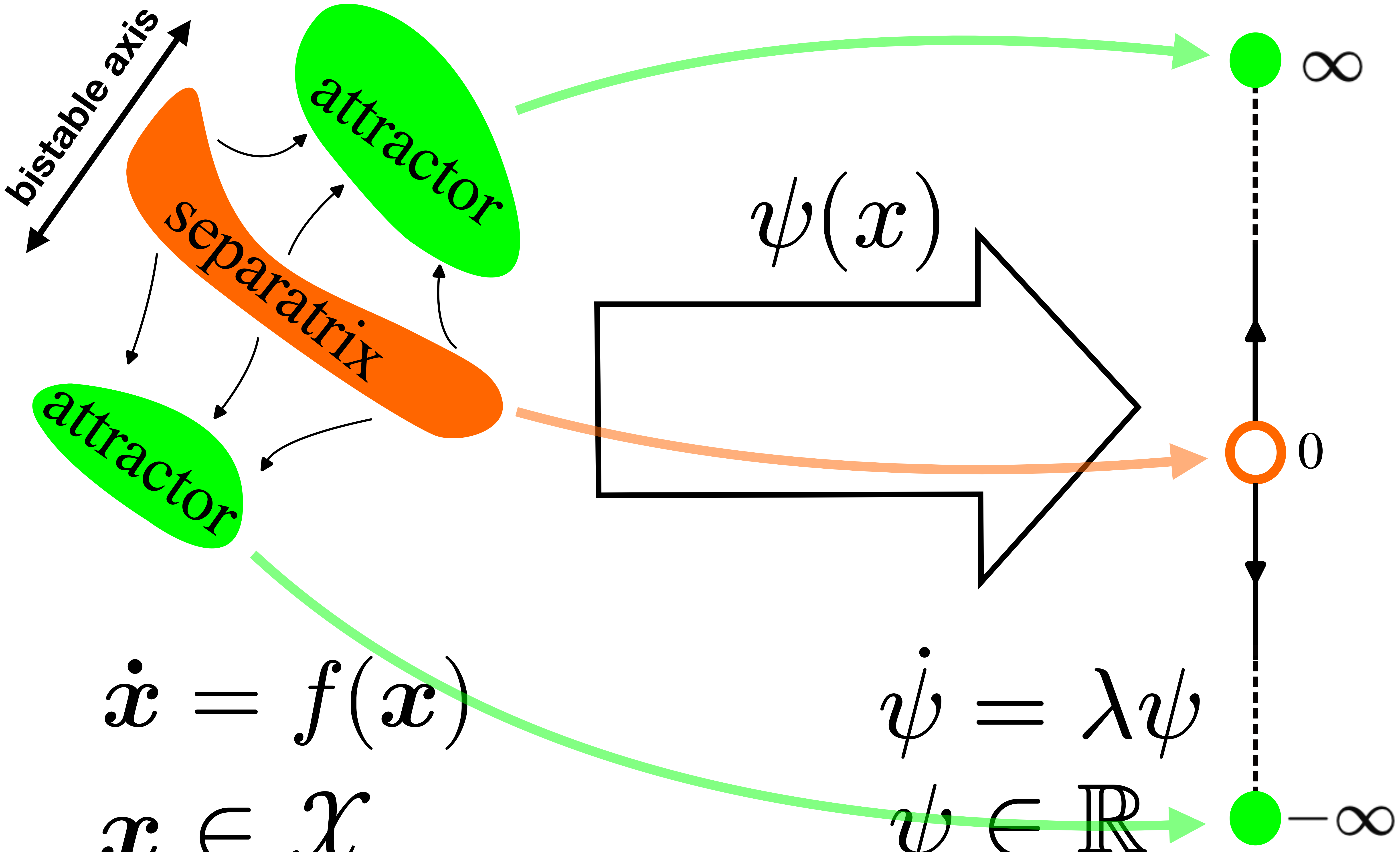
$$\dot{\psi} = \lambda \psi$$

$$\psi \in \mathbb{R}$$

$$\lambda > 0$$

How do we find such a function?

1D, linear, unstable dynamics

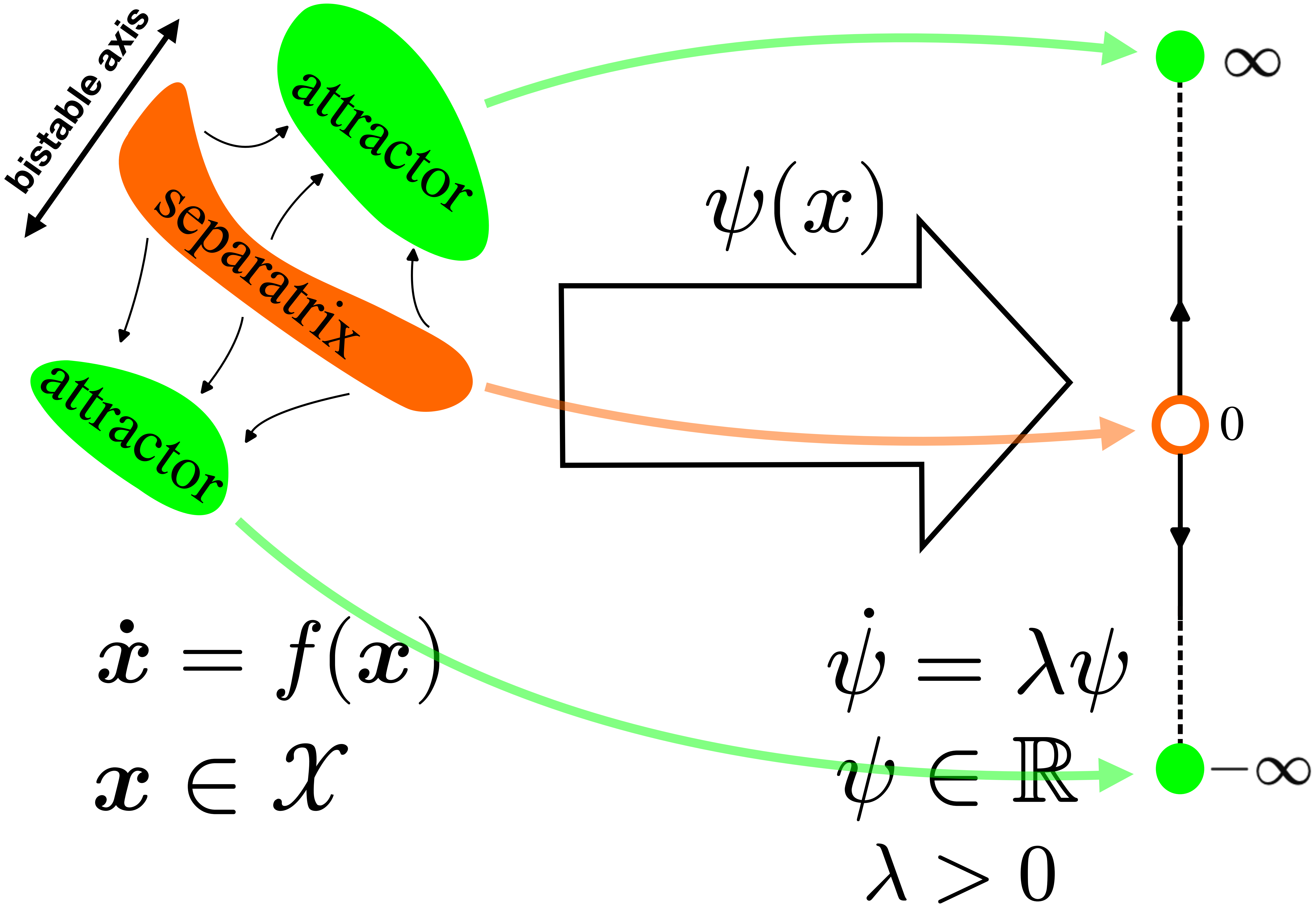


$$\dot{\boldsymbol{x}} = f(\boldsymbol{x})$$
$$\boldsymbol{x} \in \mathcal{X}$$

$$\dot{\psi} = \lambda \psi$$
$$\psi \in \mathbb{R}$$
$$\lambda > 0$$

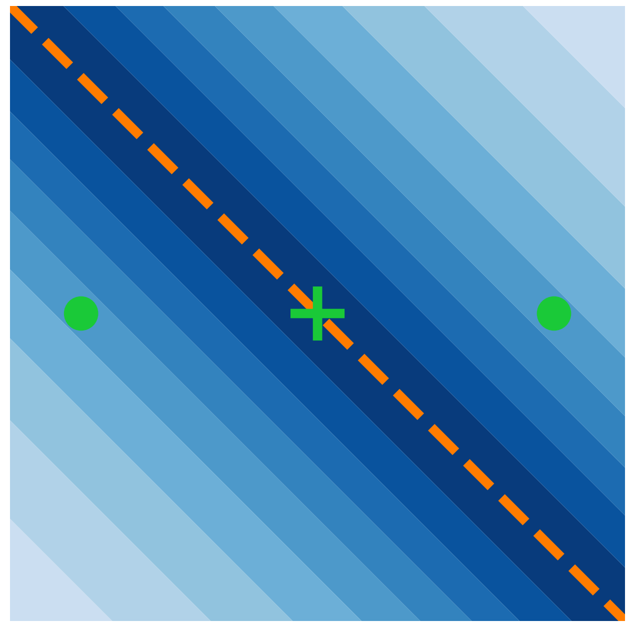
How do we find such a function?

1D, linear, unstable dynamics



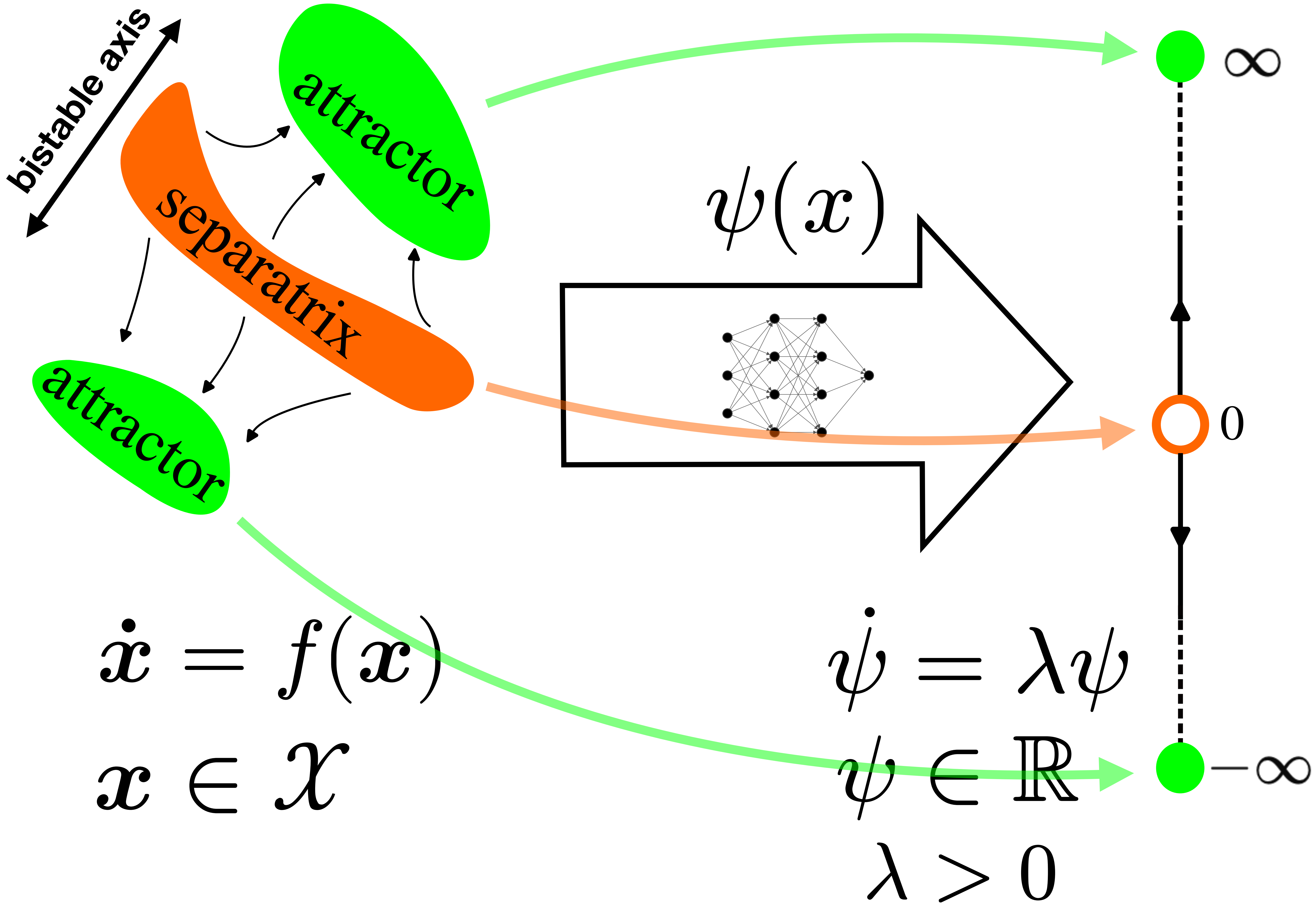
exactly what we were looking for!

$\psi(x) := ?$



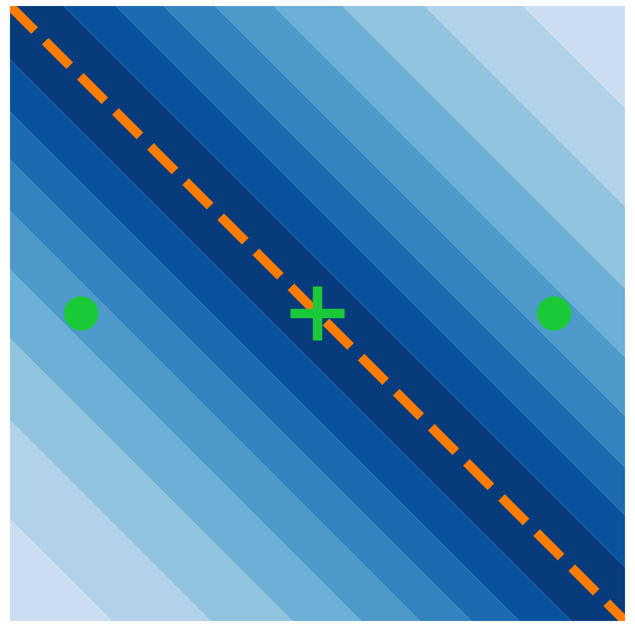
How do we find such a function?

1D, linear, unstable dynamics



exactly what we were looking for!

$\psi(x) := \,?$



we want

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$

$\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$

$\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi = \lambda\psi$$

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$

$\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi \qquad = \lambda\psi$$

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$

$\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\big(\boldsymbol{x}(t)\big) = \lambda\psi\big(\boldsymbol{x}(t)\big)$$

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ $\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\Big(\boldsymbol{x}(t)\Big) = \lambda\psi\Big(\boldsymbol{x}(t)\Big)$$

Koopman Eigenfunction!

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$

$\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\big(\boldsymbol{x}(t)\big) = \lambda\psi\big(\boldsymbol{x}(t)\big)$$

Koopman Eigenfunction!

chain-rule

$$\nabla\psi(\boldsymbol{x})^T \frac{d\boldsymbol{x}}{dt}$$

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$

$\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\big(\boldsymbol{x}(t)\big) = \lambda\psi\big(\boldsymbol{x}(t)\big)$$

Koopman Eigenfunction!

chain-rule

$$\nabla\psi(\boldsymbol{x})^T f(\boldsymbol{x})$$

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ $\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\Big(\boldsymbol{x}(t)\Big) = \lambda\psi\Big(\boldsymbol{x}(t)\Big)$$

Koopman Eigenfunction!

chain-rule

$$\nabla\psi(\boldsymbol{x})^T f(\boldsymbol{x}) = \lambda\,\psi(\boldsymbol{x})$$

partial **differential equation (PDE)**

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ $\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\Big(\boldsymbol{x}(t)\Big) = \lambda\psi\Big(\boldsymbol{x}(t)\Big)$$

Koopman Eigenfunction!

chain-rule

$$\nabla\psi(\boldsymbol{x})^T f(\boldsymbol{x}) - \lambda\,\psi(\boldsymbol{x})$$

***partial* differential equation (PDE)**

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ $\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\big(\boldsymbol{x}(t)\big) = \lambda\psi\big(\boldsymbol{x}(t)\big)$$

Koopman Eigenfunction!

chain-rule

$$\left[\nabla\psi(\boldsymbol{x})^T f(\boldsymbol{x}) - \lambda\,\psi(\boldsymbol{x})\right]^2$$

partial **differential equation (PDE)**

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ $\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\big(\boldsymbol{x}(t)\big) = \lambda\psi\big(\boldsymbol{x}(t)\big)$$

Koopman Eigenfunction!

chain-rule

$$\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\nabla\psi(\boldsymbol{x})^T f(\boldsymbol{x}) - \lambda\,\psi(\boldsymbol{x})\right]^2$$

***partial* differential equation (PDE)**

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ $\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\Big(\boldsymbol{x}(t)\Big) = \lambda\psi\Big(\boldsymbol{x}(t)\Big)$$

Koopman Eigenfunction!

chain-rule

$$\mathcal{L}_{\text{PDE}} = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\Big[\nabla\psi(\boldsymbol{x})^T f(\boldsymbol{x}) - \lambda\,\psi(\boldsymbol{x})\Big]^2$$

partial **differential equation (PDE)**

Loss function

we want

$\boldsymbol{x}(t)$ **evolves as** $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ $\psi\big(\boldsymbol{x}(t)\big)$ **evolves as:**

$$\frac{d}{dt}\psi\Big(\boldsymbol{x}(t)\Big) = \lambda\psi\Big(\boldsymbol{x}(t)\Big)$$

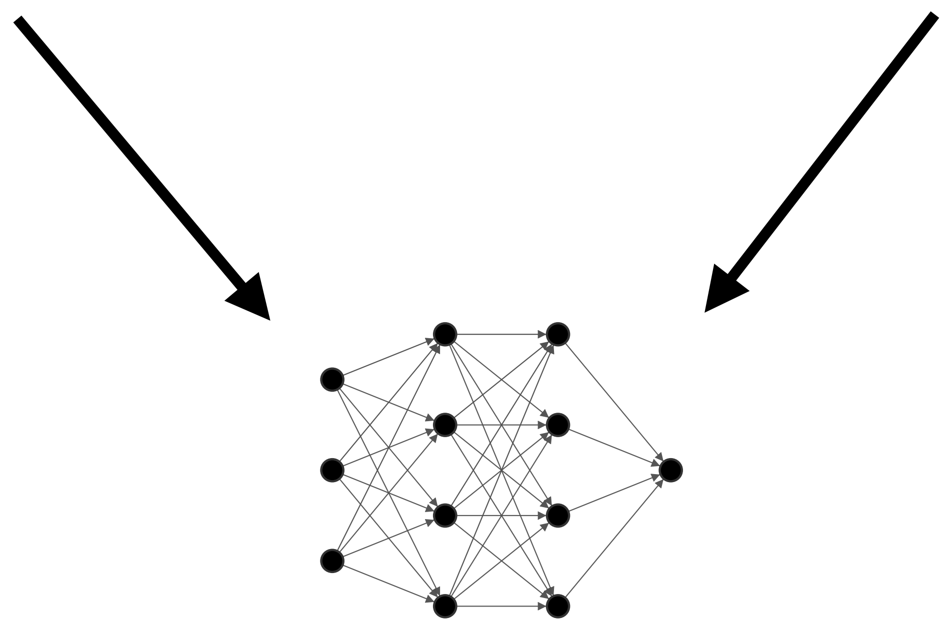
Koopman Eigenfunction!

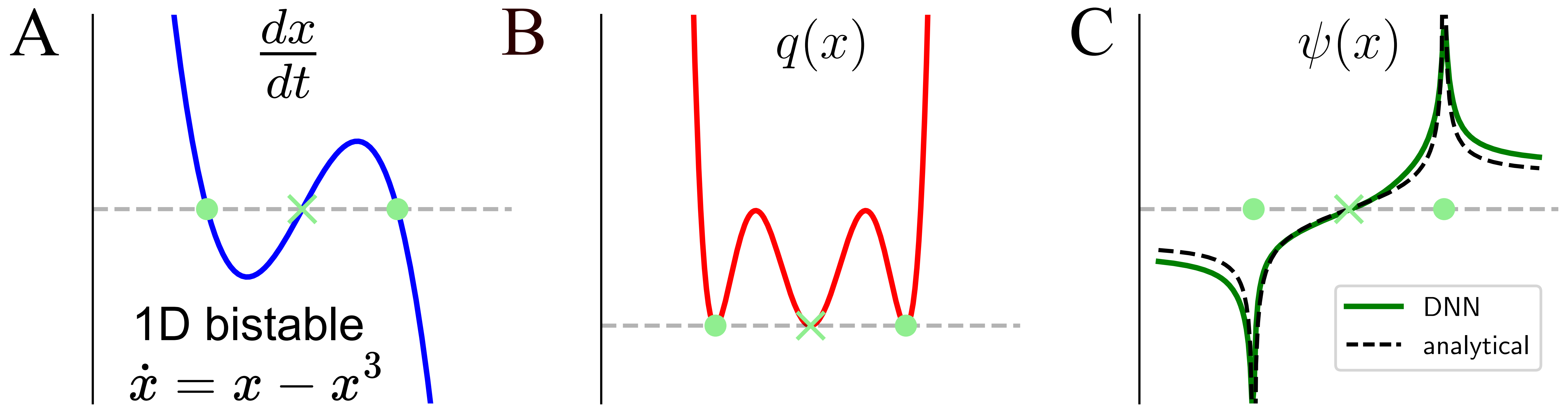
chain-rule

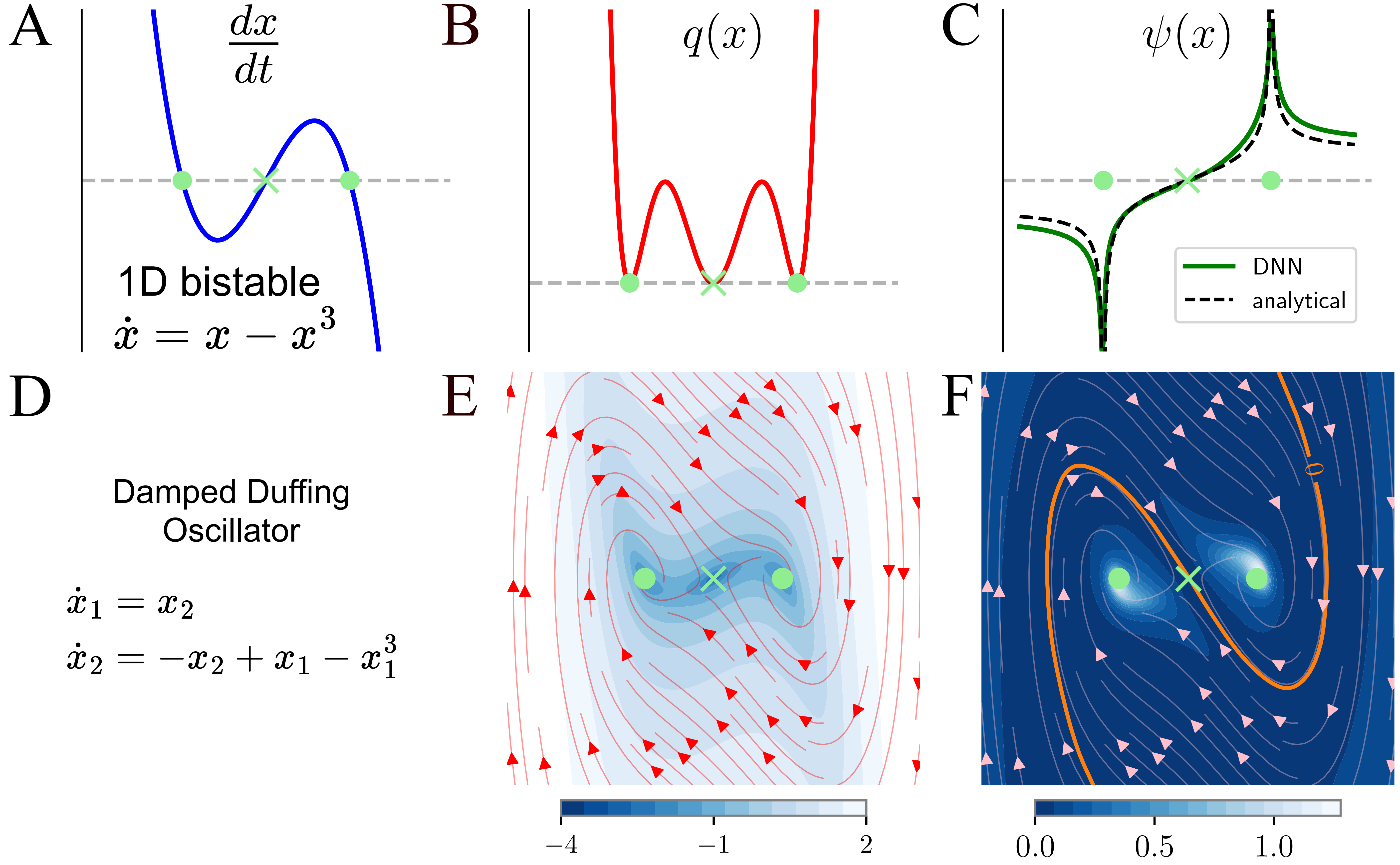
$$\mathcal{L}_{\mathrm{PDE}} = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\Big[\nabla\psi(\boldsymbol{x})^T f(\boldsymbol{x}) - \lambda\,\psi(\boldsymbol{x})\Big]^2$$

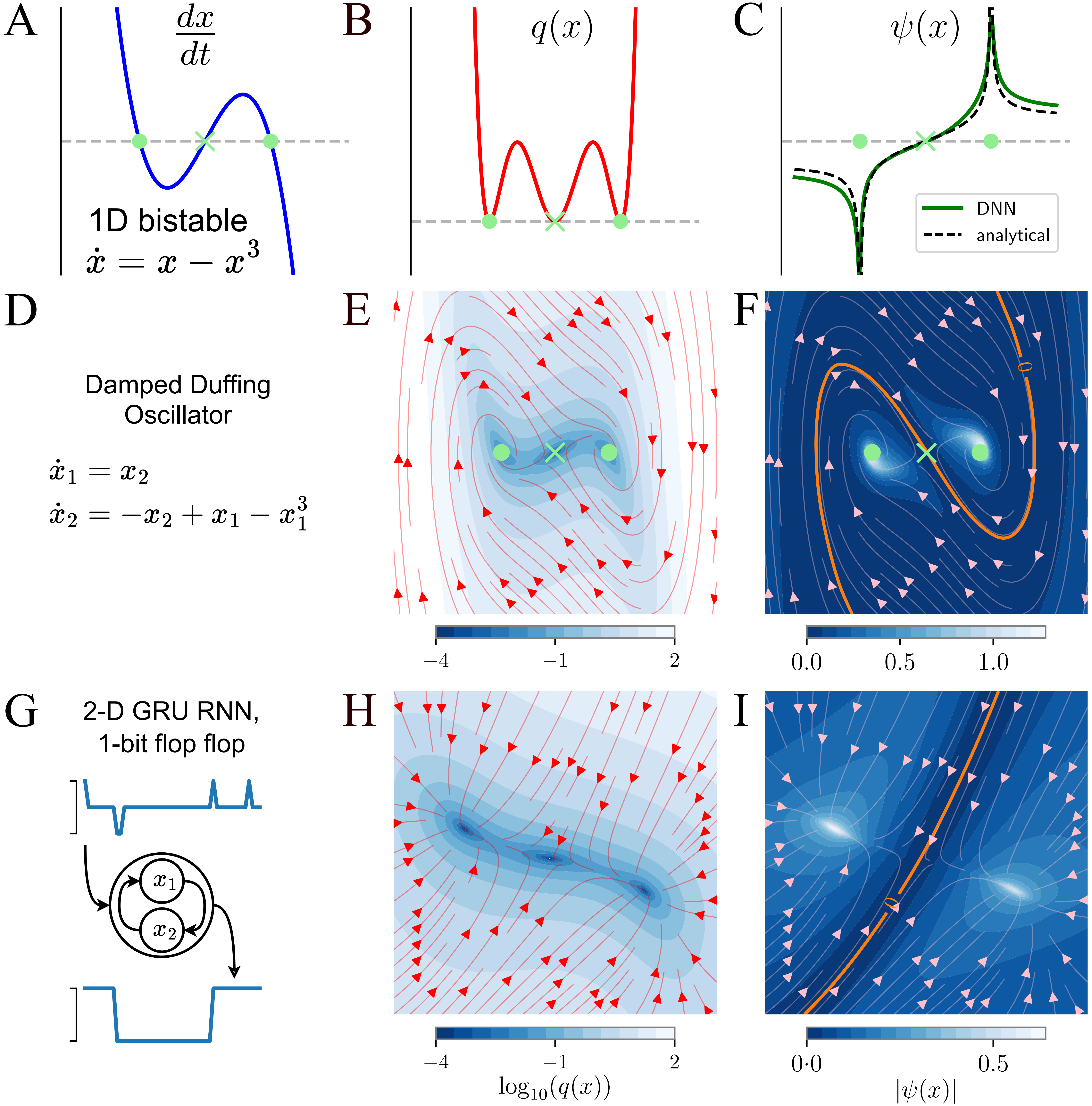
partial **differential equation (PDE)**

Loss function




A
$\frac{dx}{dt}$
1D bistable
$\dot{x} = x - x^3$
B
$q(x)$
C
$\psi(x)$
DNN
analytical


A
$\frac{dx}{dt}$
1D bistable
$\dot{x} = x - x^3$
B
$q(x)$
C
$\psi(x)$
DNN
analytical
D
Damped Duffing
Oscillator
$\dot{x}_1 = x_2$
$\dot{x}_2 = -x_2 + x_1 - x_1^3$
E
F
−4
−1
2
0.0
0.5
1.0



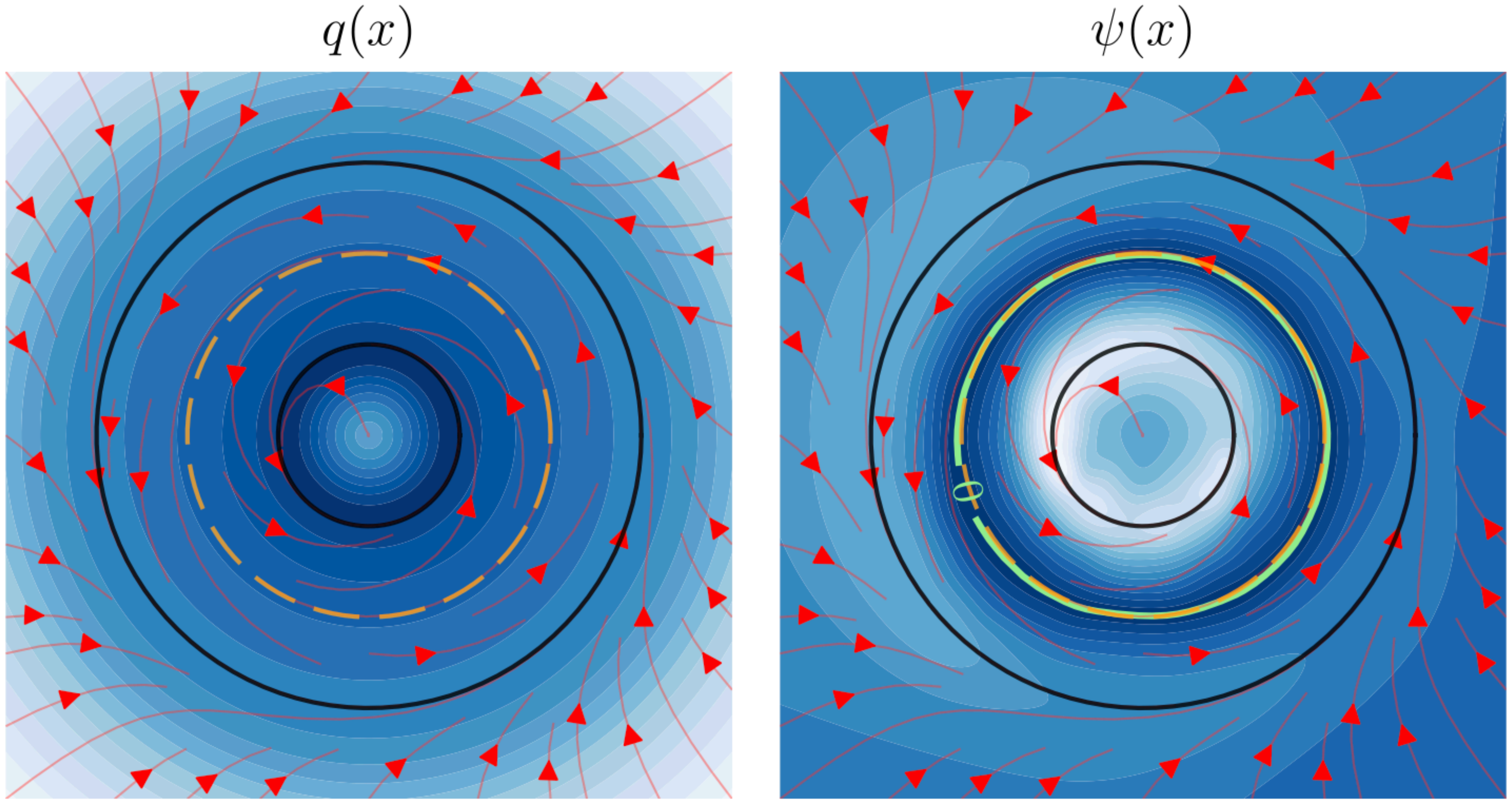
A
$\frac{dx}{dt}$
1D bistable
$\dot{x} = x - x^3$
B
$q(x)$
C
$\psi(x)$
DNN
analytical
D
Damped Duffing Oscillator
$\dot{x}_1 = x_2$
$\dot{x}_2 = -x_2 + x_1 - x_1^3$
E
−4
−1
2
F
0.0
0.5
1.0
G
2-D GRU RNN, 1-bit flop flop
x_1
x_2
H
−4
−1
2
$\log_{10}(q(x))$
I
0.0
0.5
$|\psi(x)|$

Bistable oscillations

No fixed points!

bistable oscillations

$$\dot{r} = (r-2) - (r-2)^3$$
$$\dot{\theta} = 1$$

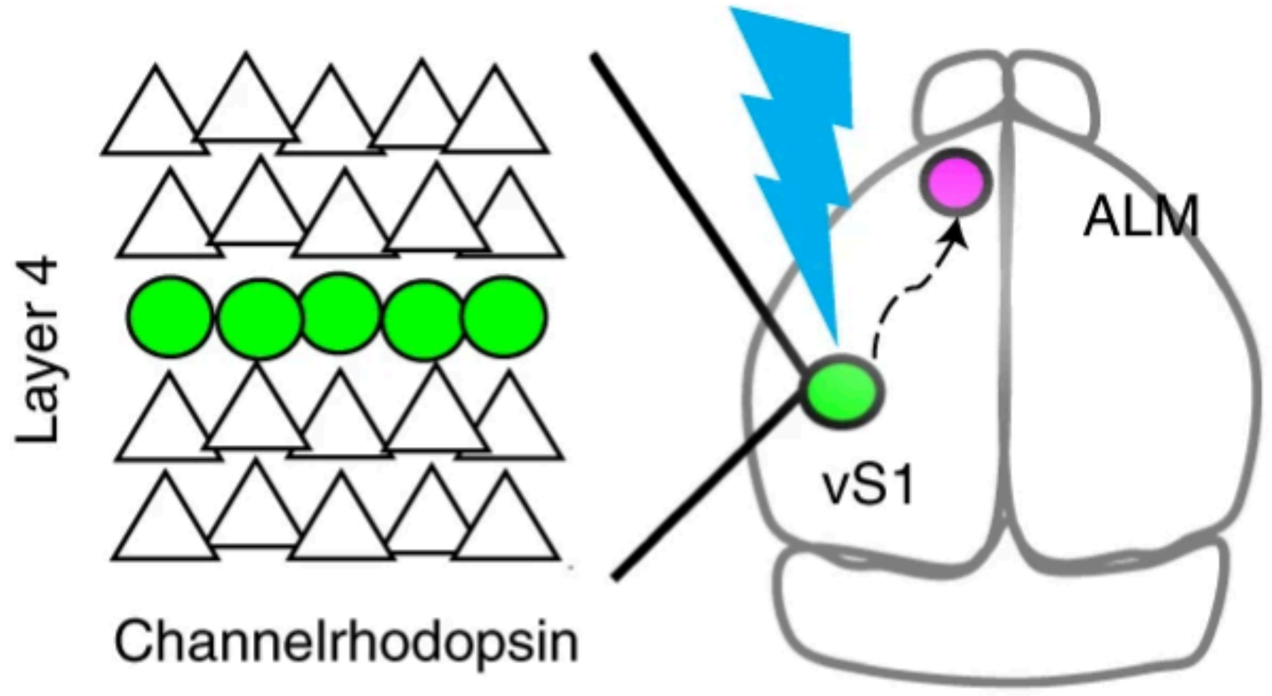


Data-trained RNN, $N = 668$ dimensions

nature neuroscience
ARTICLES
https://doi.org/10.1038/s41593-021-00840-6
Check for updates

Attractor dynamics gate cortical information flow during decision-making

Arseny Finkelstein[1,3], Lorenzo Fontolan[1,3], Michael N. Economo[1], Nuo Li[1,2], Sandro Romani[1] ✉ and Karel Svoboda[1] ✉

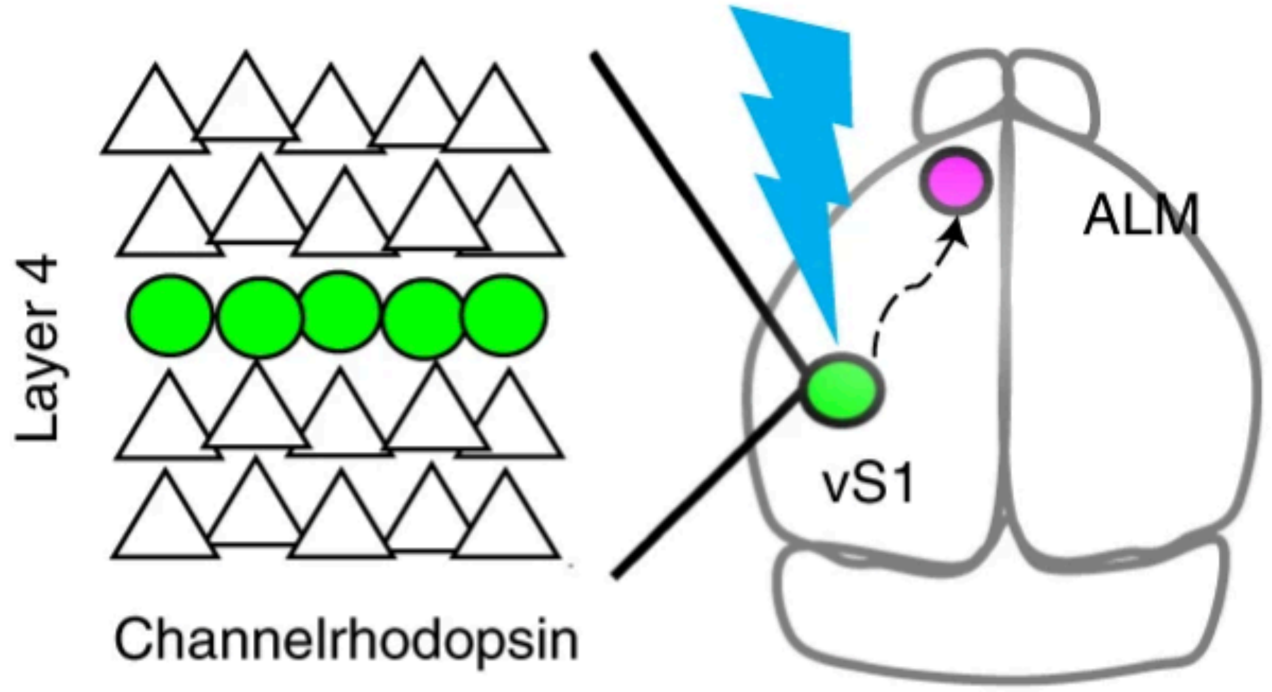


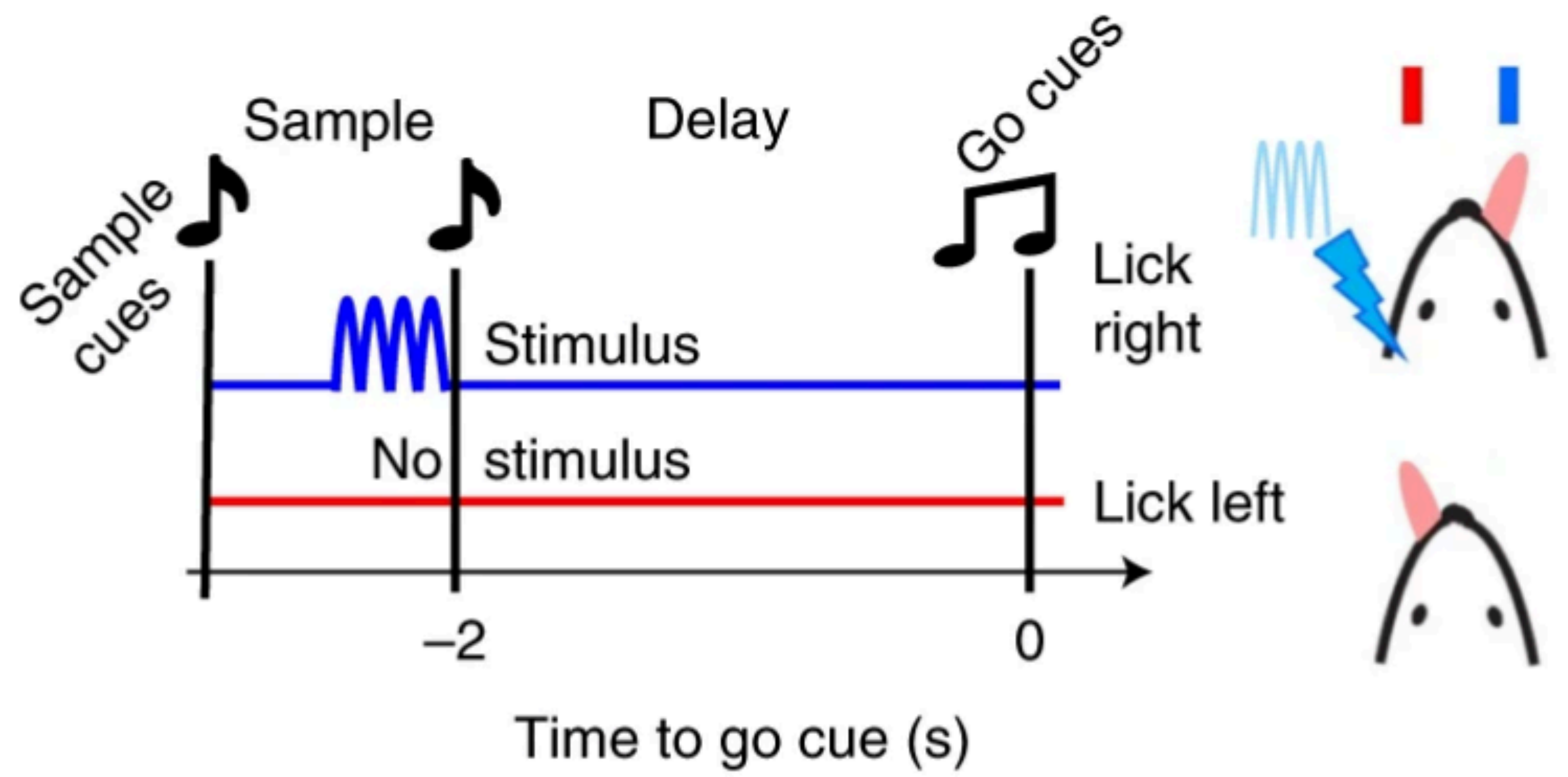
Data-trained RNN, $N = 668$ dimensions

nature neuroscience
ARTICLES
https://doi.org/10.1038/s41593-021-00840-6
Check for updates

Attractor dynamics gate cortical information flow during decision-making

Arseny Finkelstein[1,3], Lorenzo Fontolan[1,3], Michael N. Economo[1], Nuo Li[1,2], Sandro Romani[1] ✉ and Karel Svoboda[1] ✉



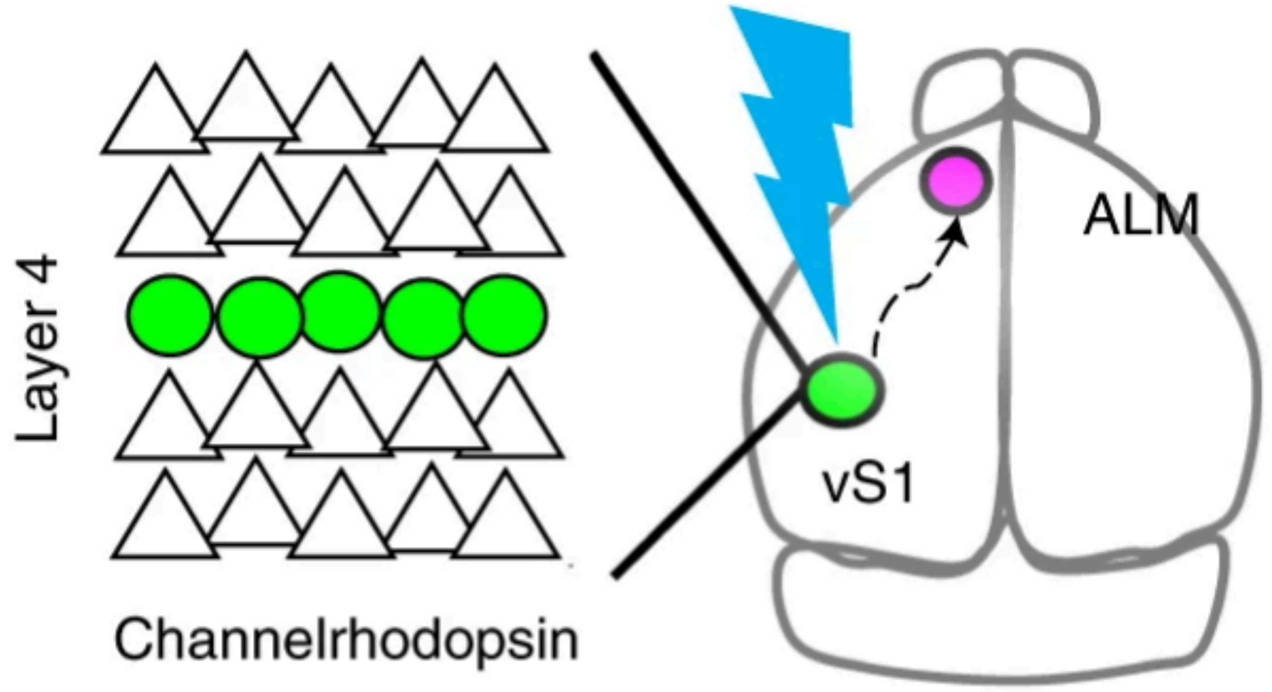


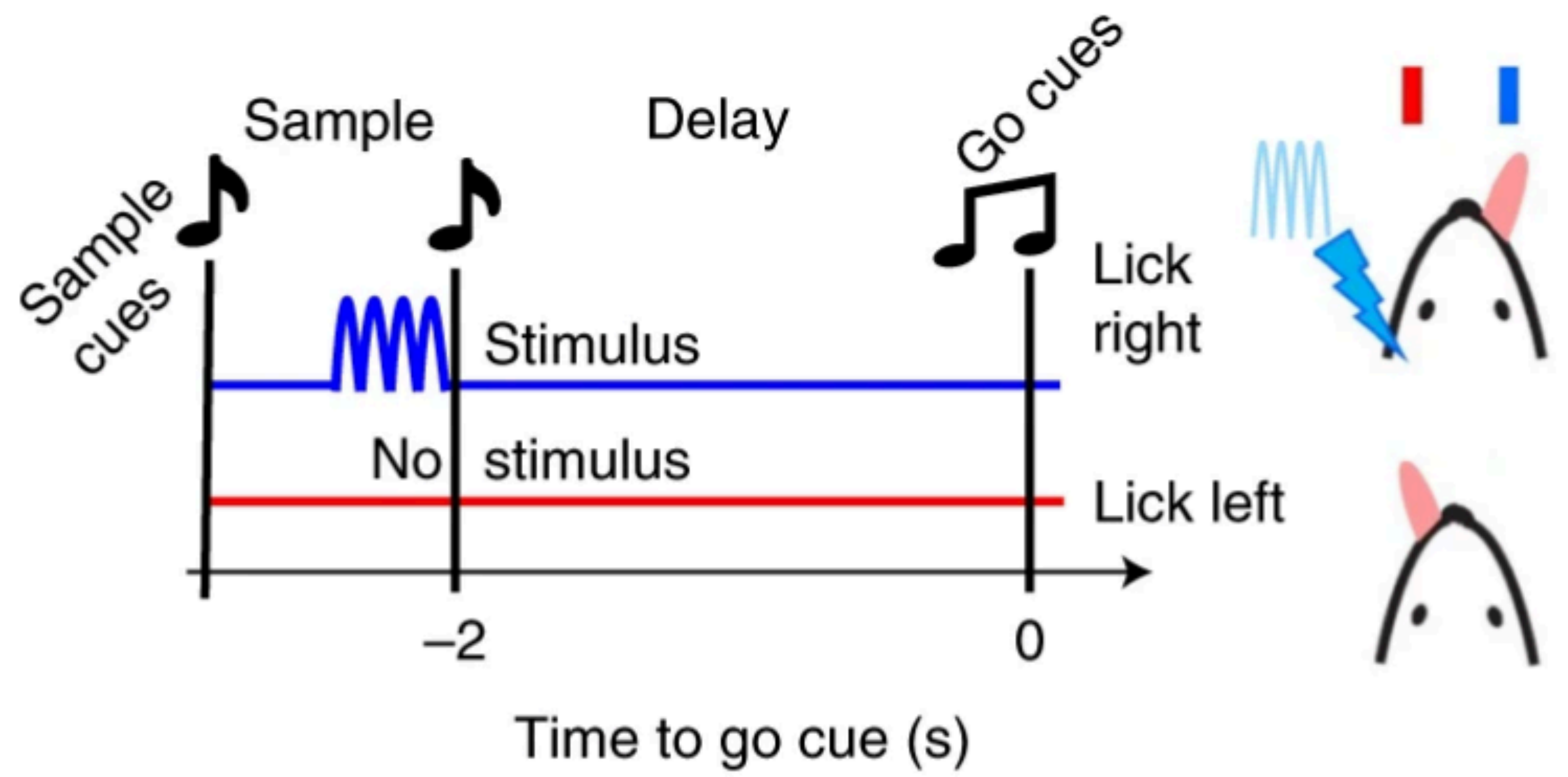
Data-trained RNN, $N = 668$ dimensions

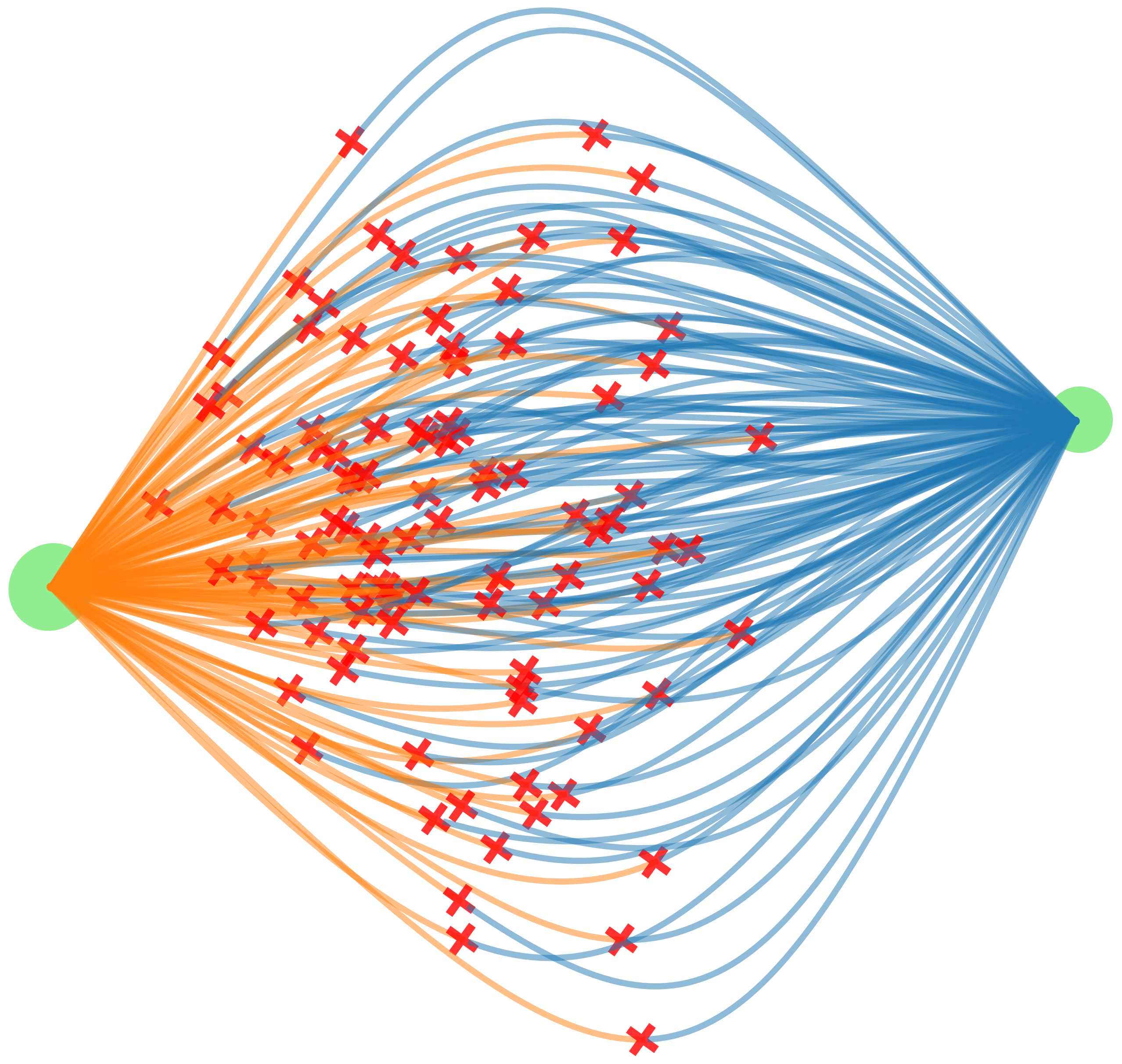
nature neuroscience
ARTICLES
https://doi.org/10.1038/s41593-021-00840-6
Check for updates

Attractor dynamics gate cortical information flow during decision-making

Arseny Finkelstein[1,3], Lorenzo Fontolan[1,3], Michael N. Economo[1], Nuo Li[1,2], Sandro Romani[1] ✉ and Karel Svoboda[1] ✉





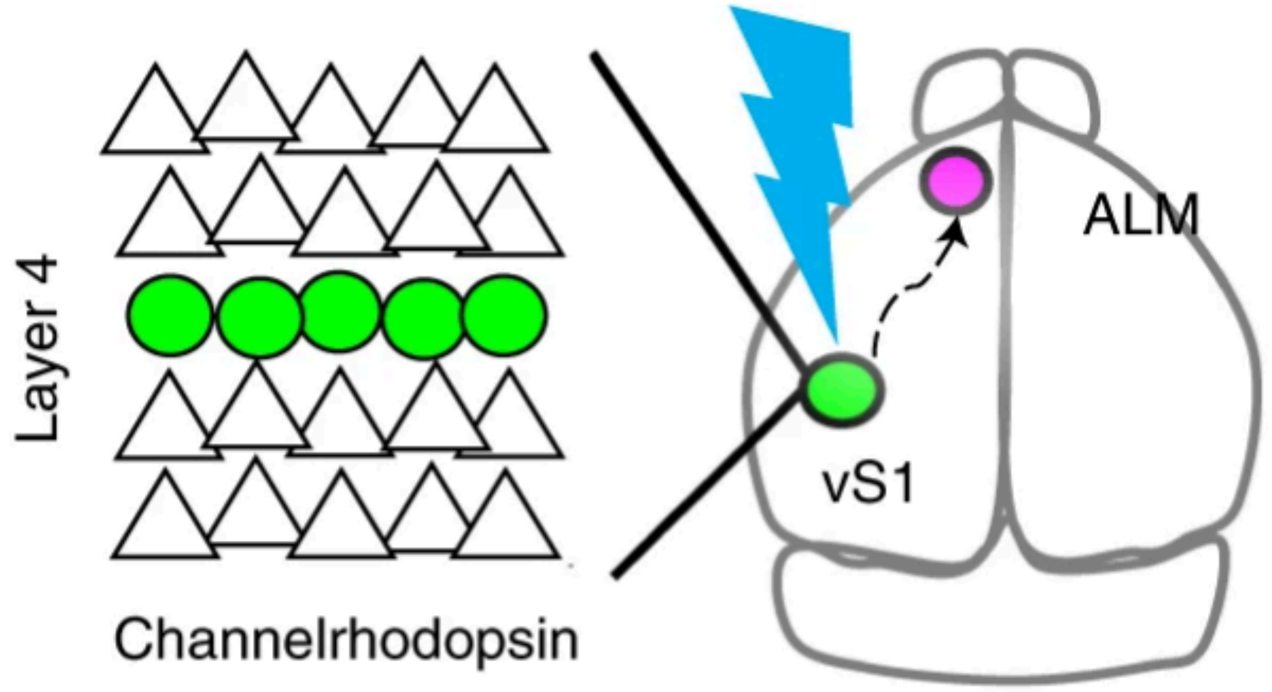


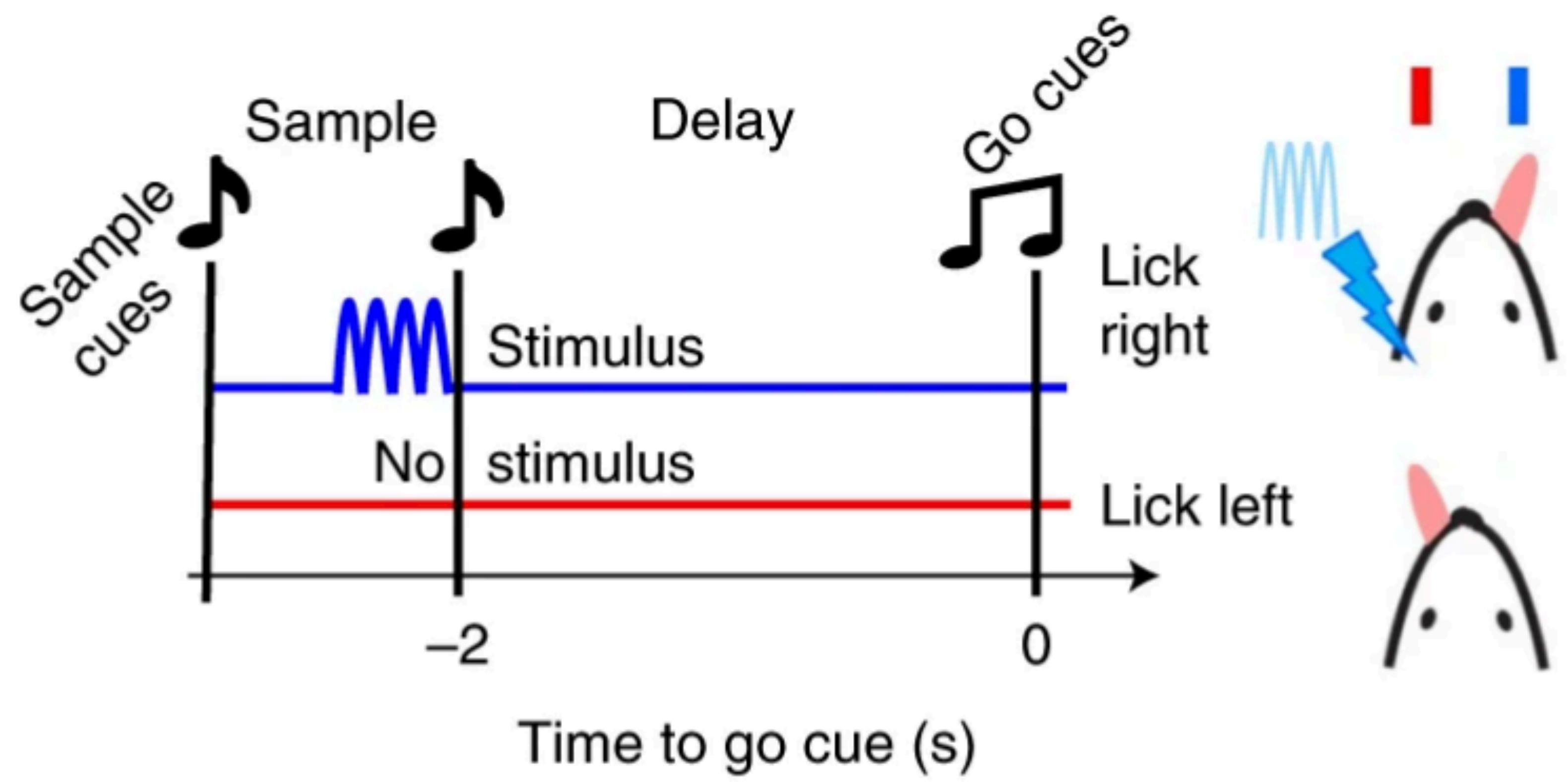
Data-trained RNN, $N = 668$ dimensions

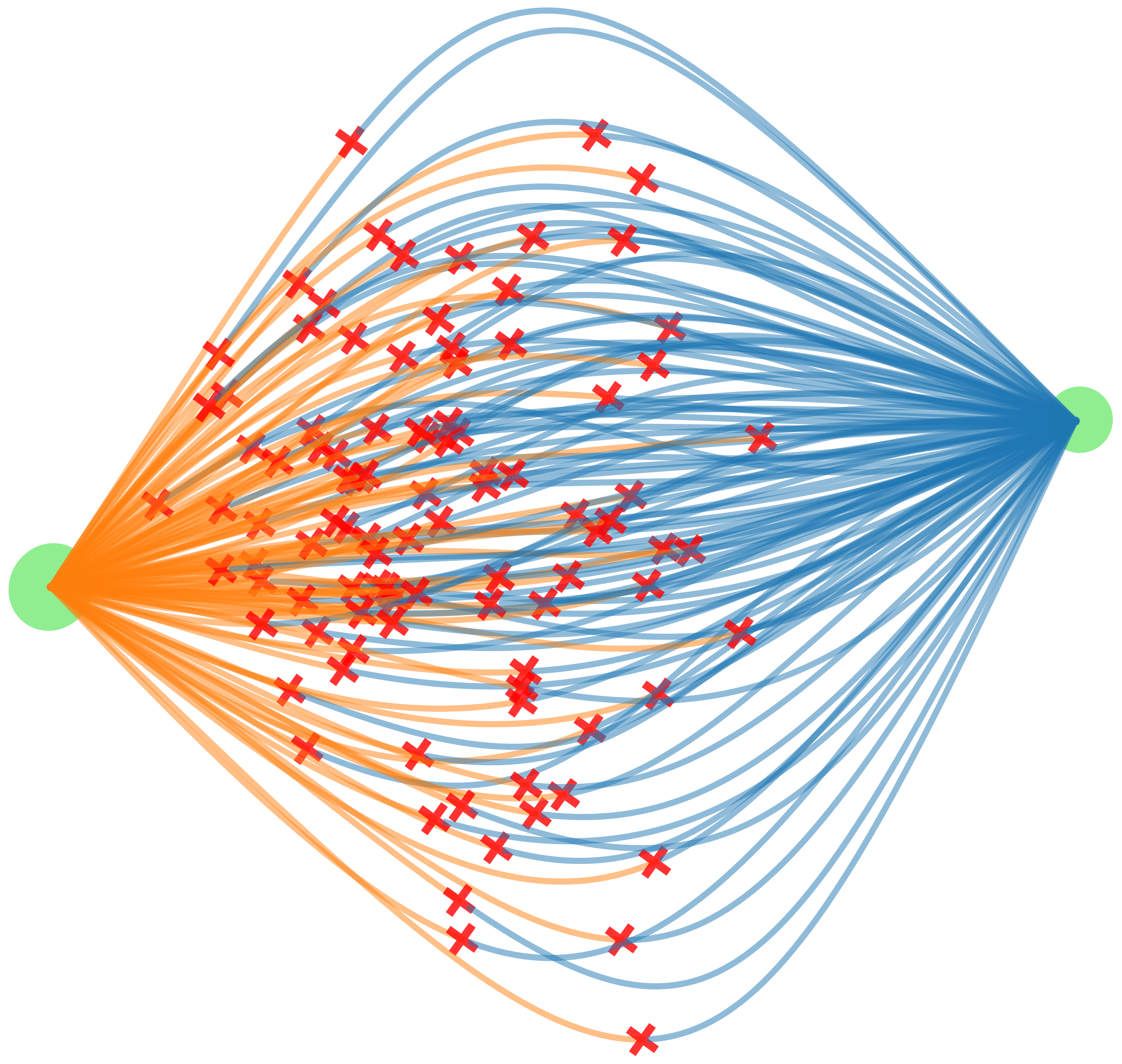

nature neuroscience
ARTICLES
https://doi.org/10.1038/s41593-021-00840-6
Check for updates

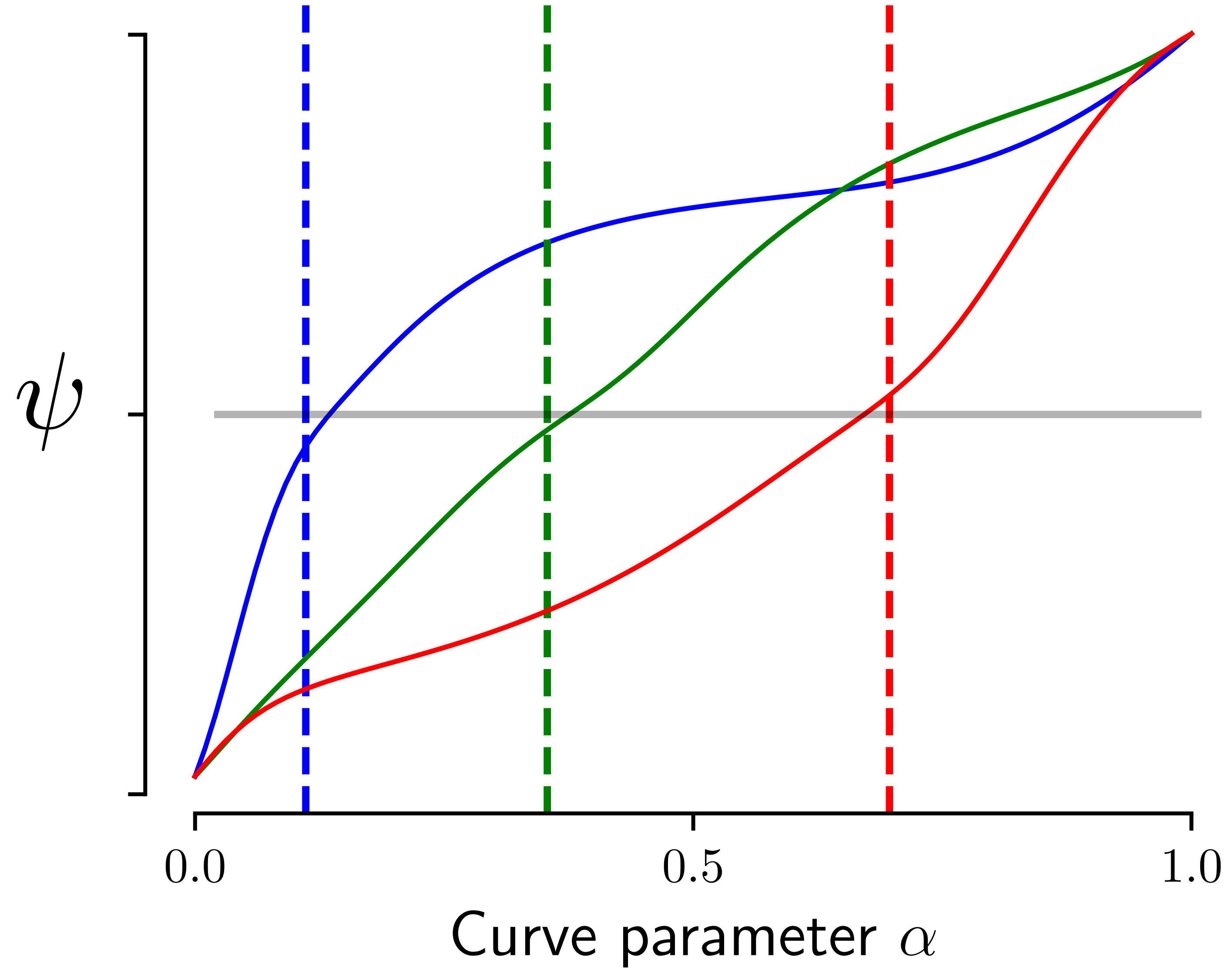
Attractor dynamics gate cortical information flow during decision-making

Arseny Finkelstein[1,3], Lorenzo Fontolan[1,3], Michael N. Economo[1], Nuo Li[1,2], Sandro Romani[1] and Karel Svoboda[1]







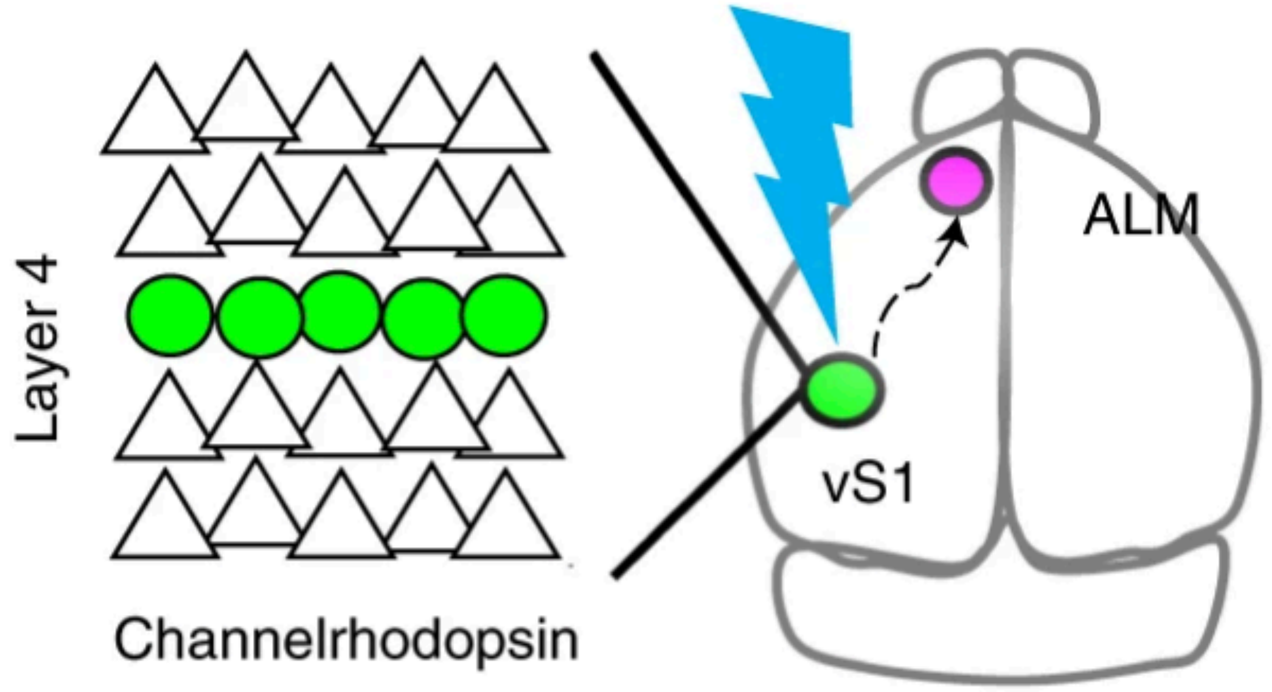


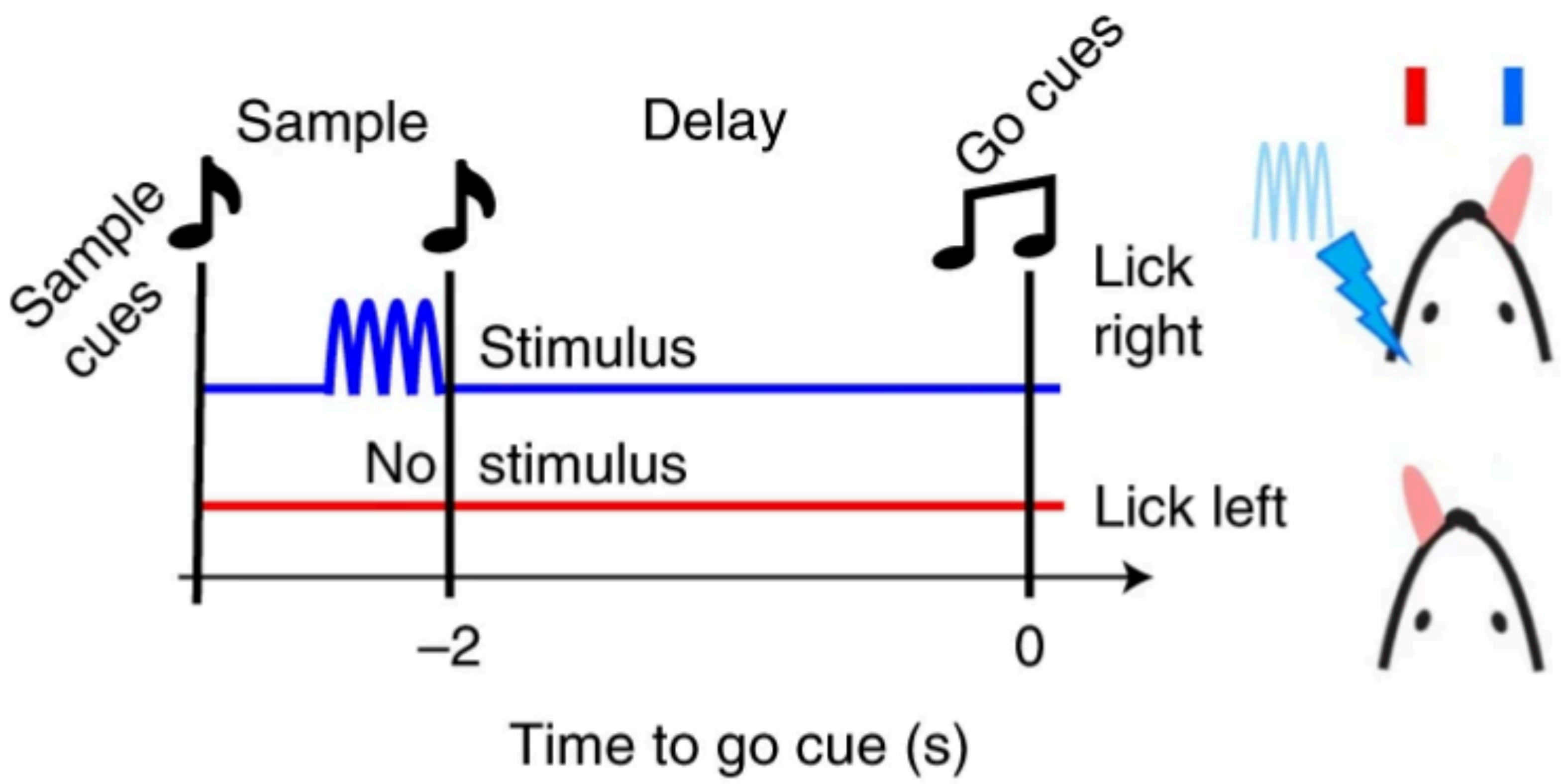
Data-trained RNN, $N = 668$ dimensions

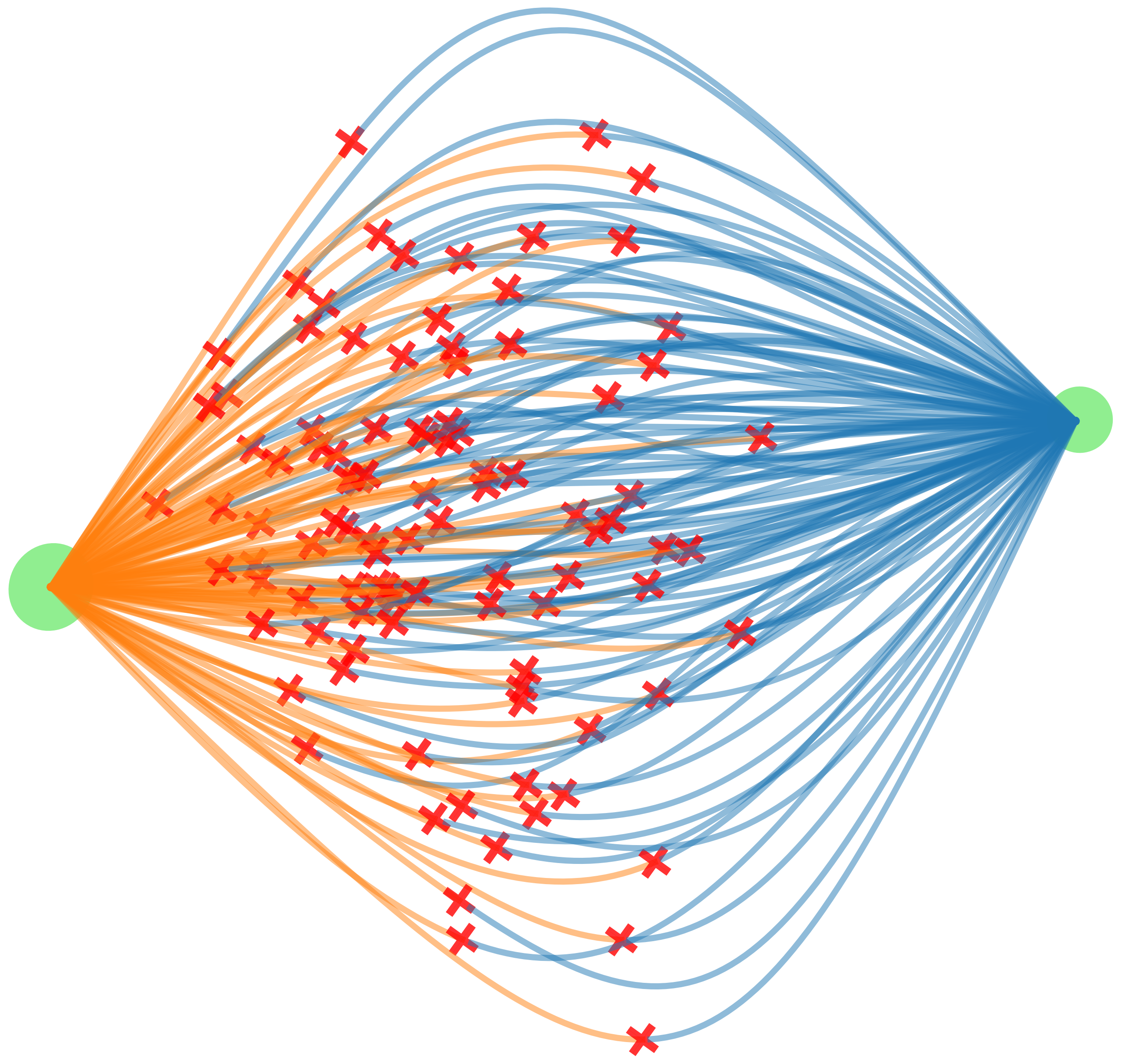

nature neuroscience | ARTICLES
https://doi.org/10.1038/s41593-021-00840-6
Check for updates

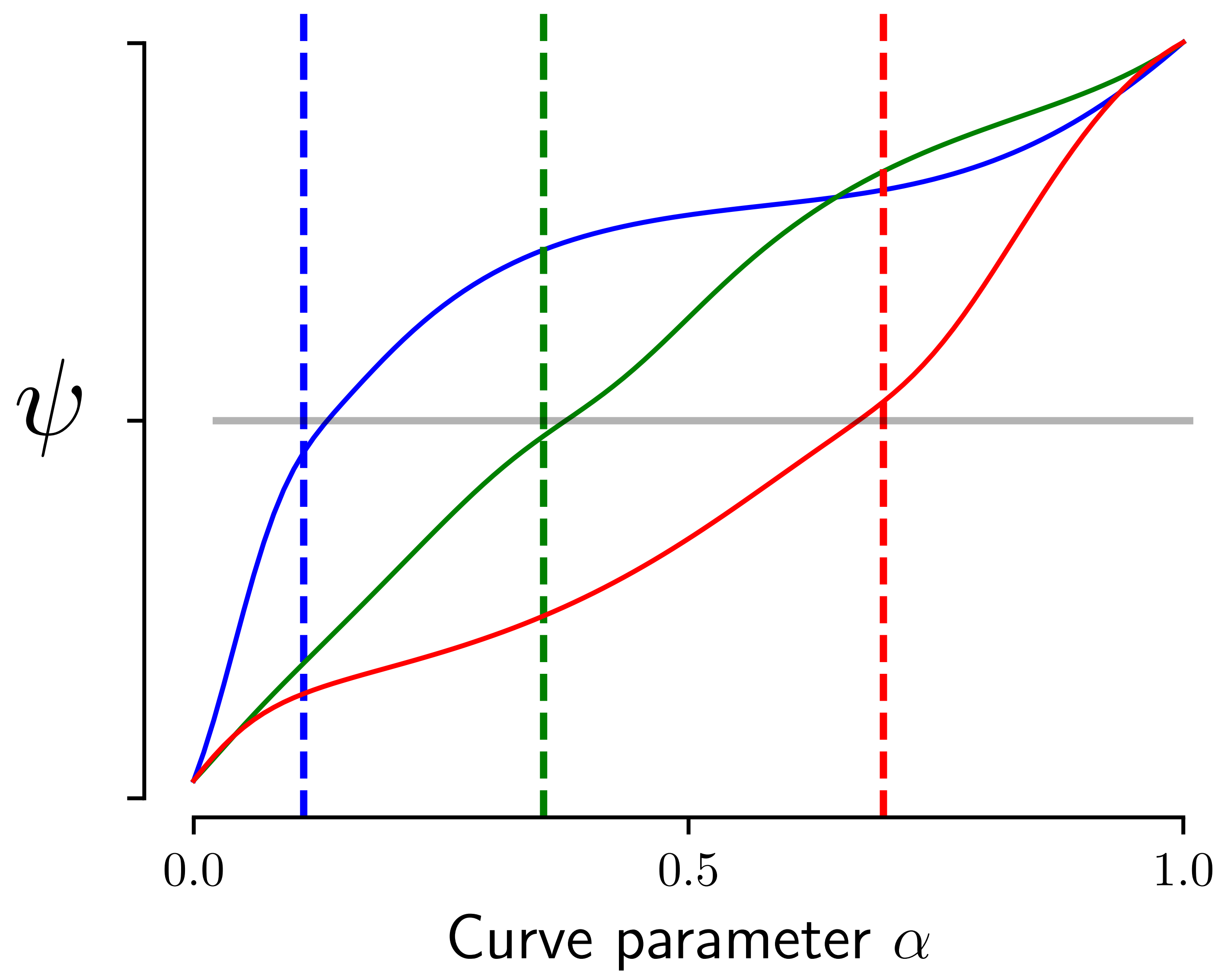
Attractor dynamics gate cortical information flow during decision-making

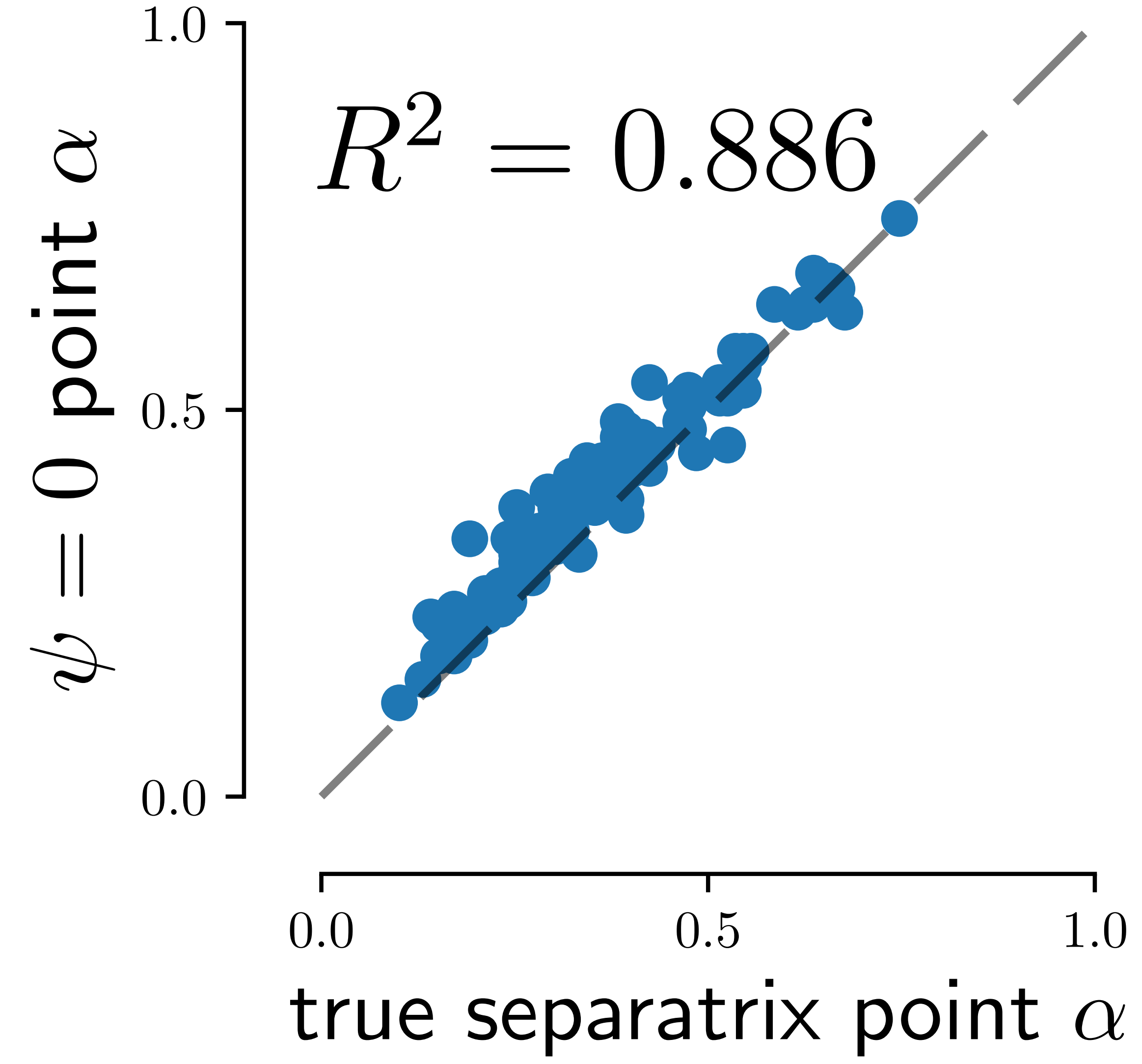
Arseny Finkelstein[1,3], Lorenzo Fontolan[1,3], Michael N. Economo[1], Nuo Li[1,2], Sandro Romani[1]✉ and Karel Svoboda[1]✉





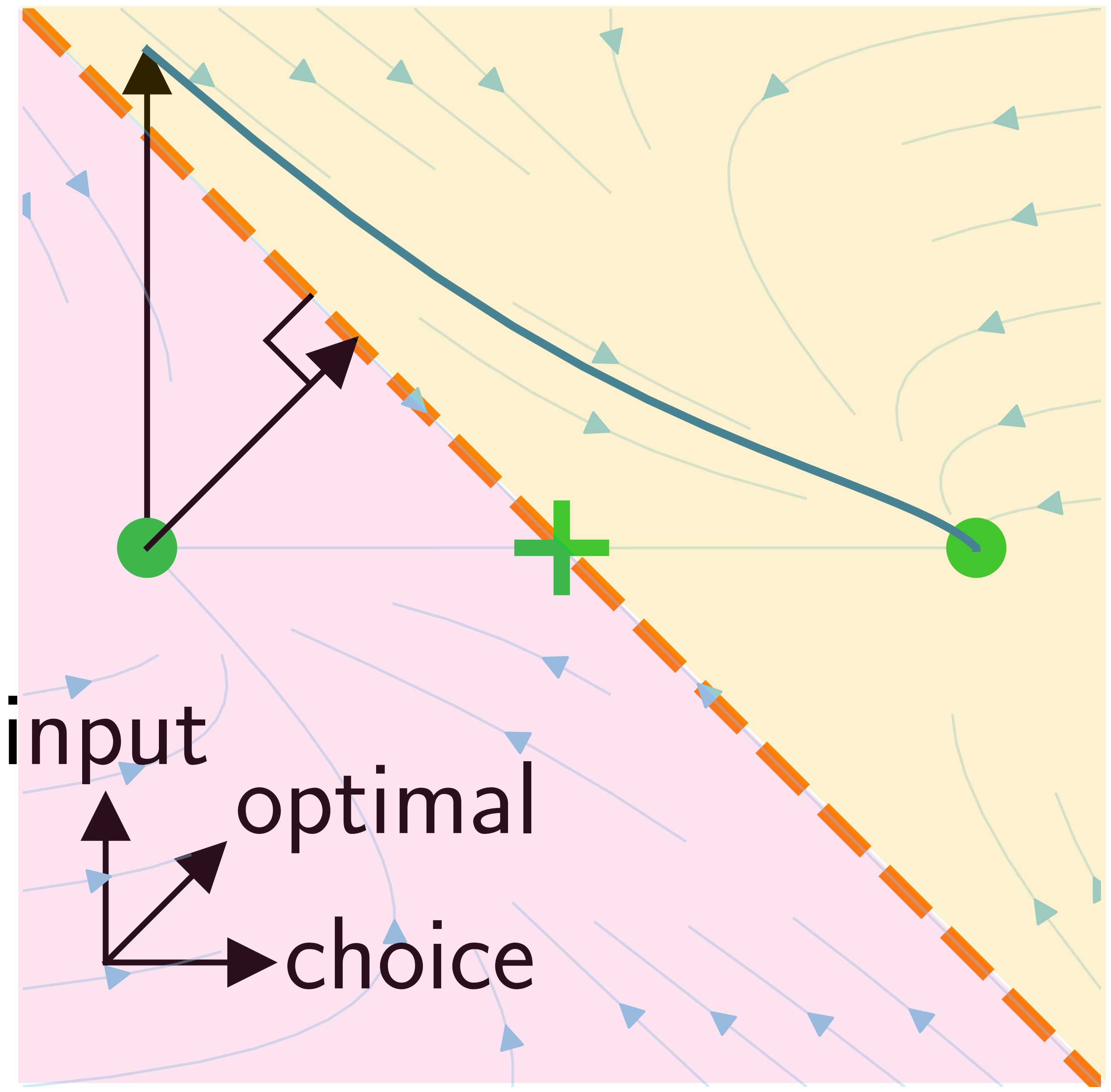




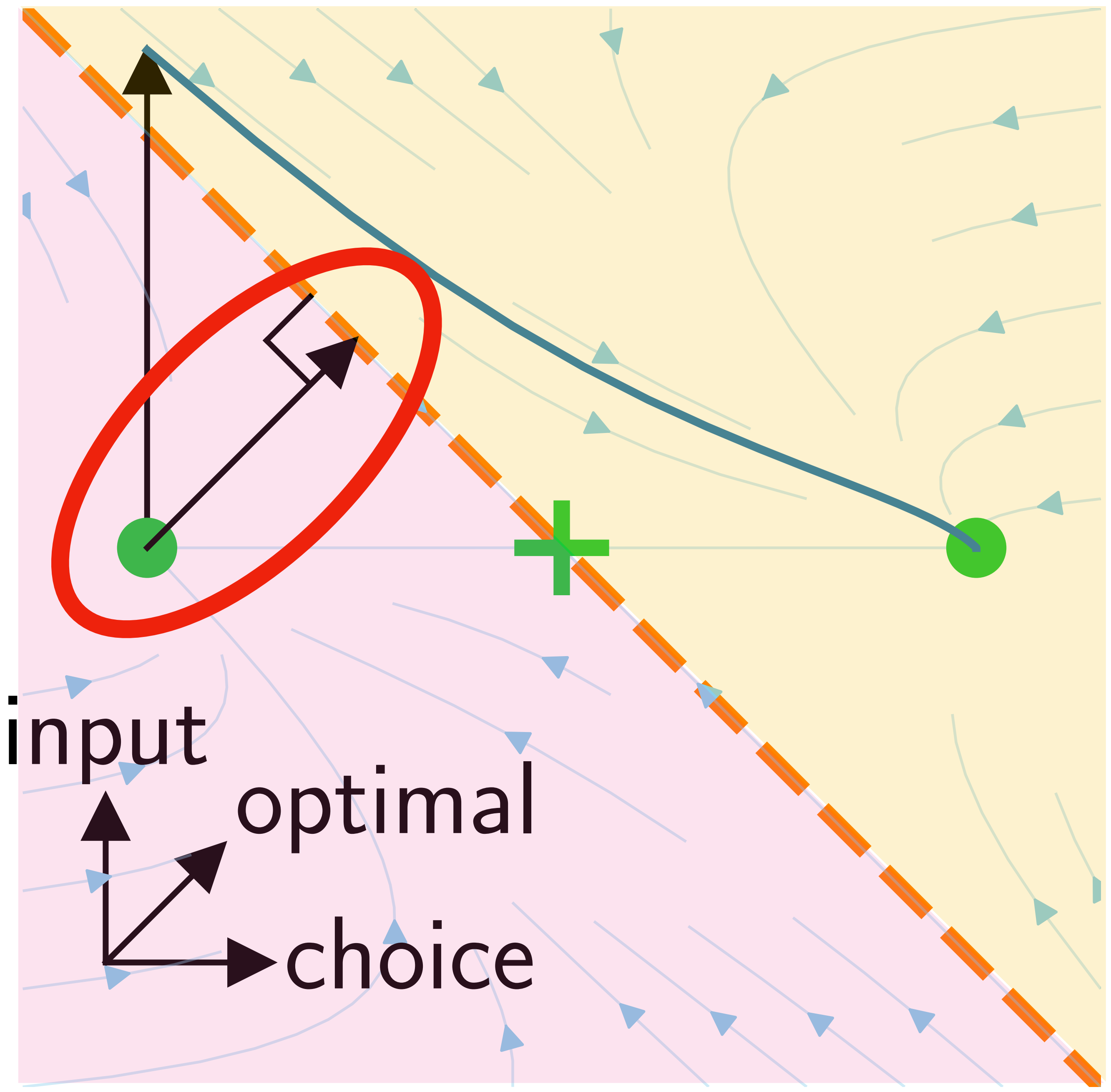


Data-trained RNN, $N = 668$

Data-trained RNN, $N = 668$

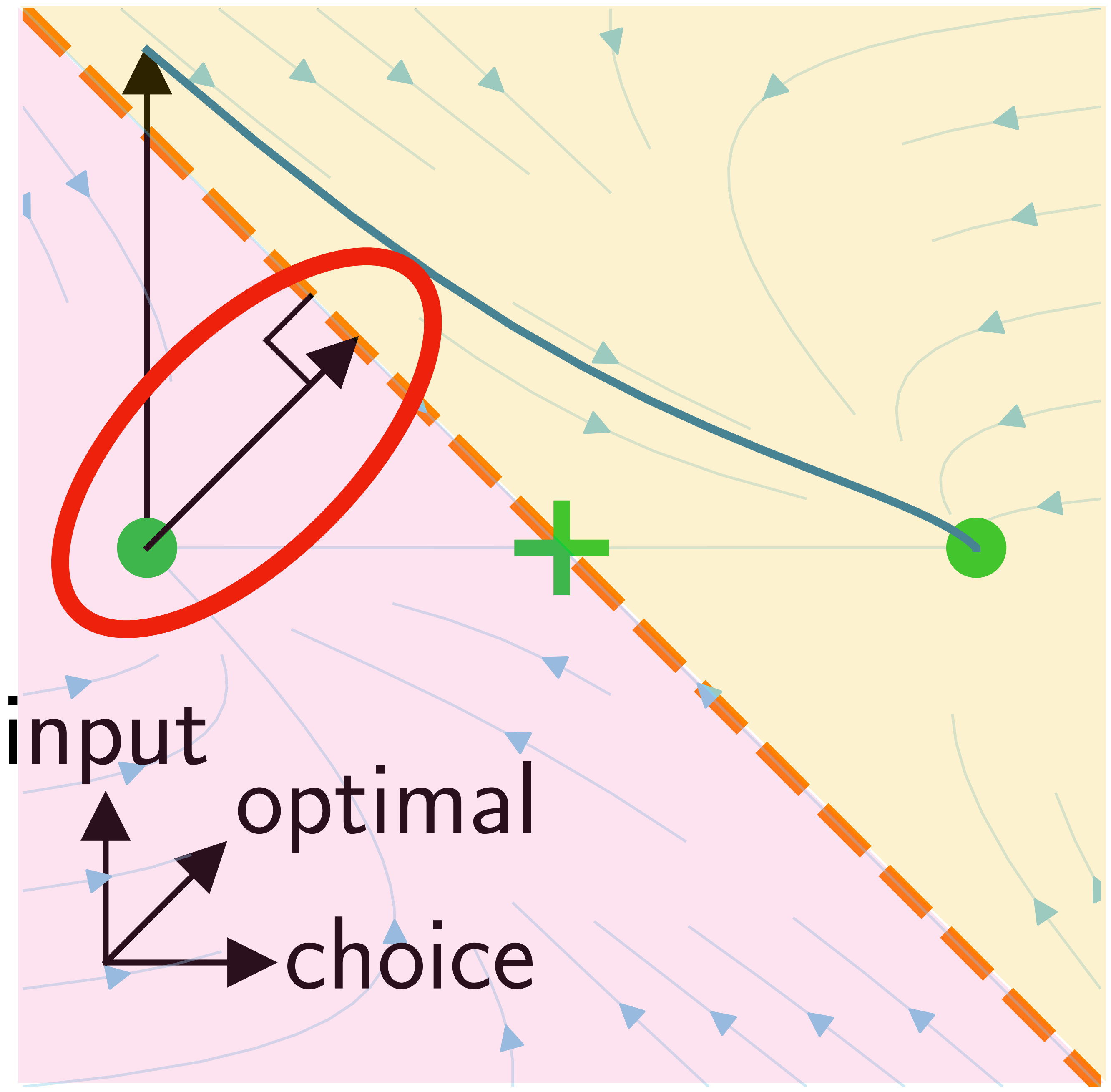


Data-trained RNN, $N = 668$

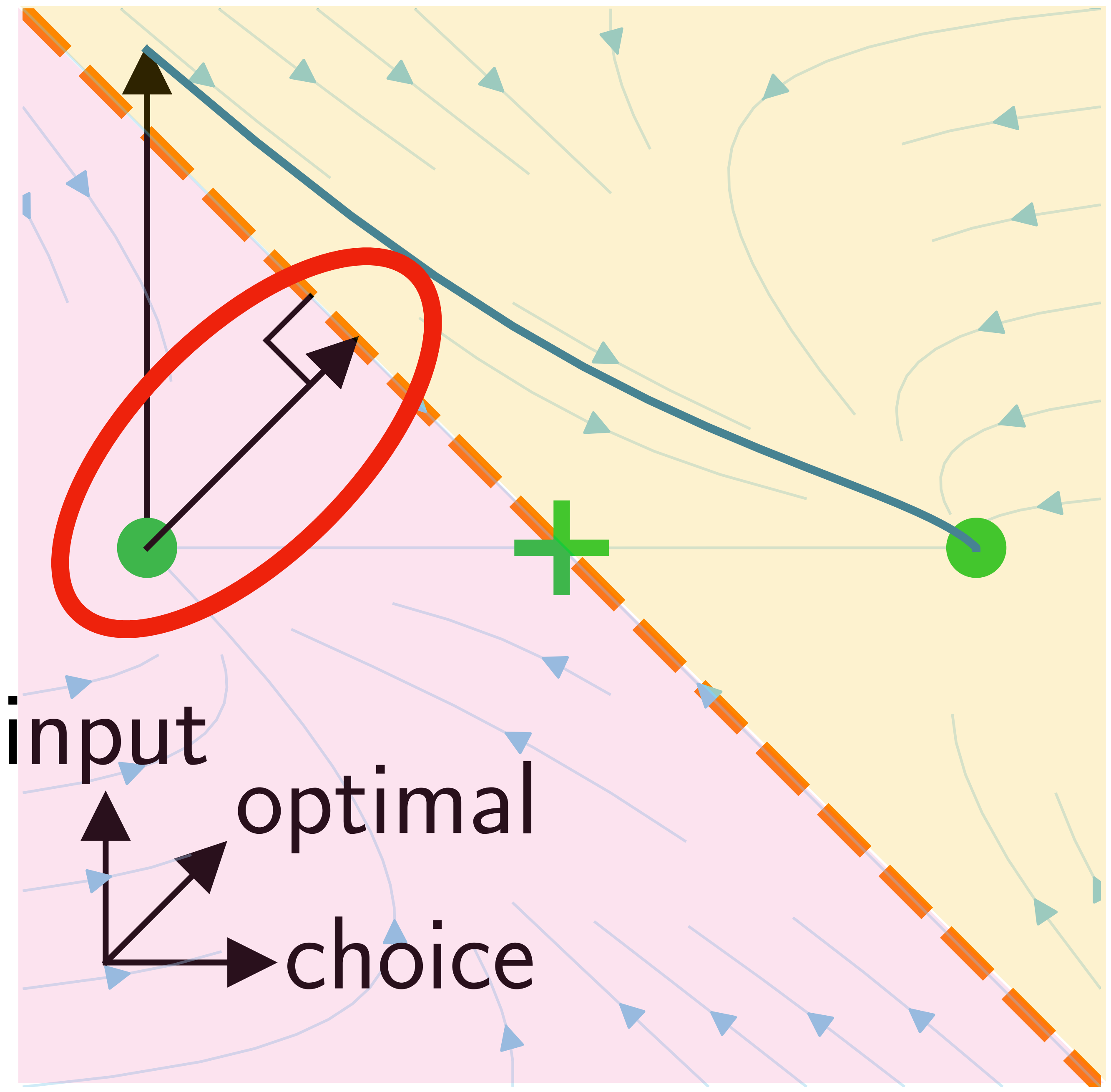


Data-trained RNN, $N = 668$

instantaneous pulse to neurons of your choice Δ



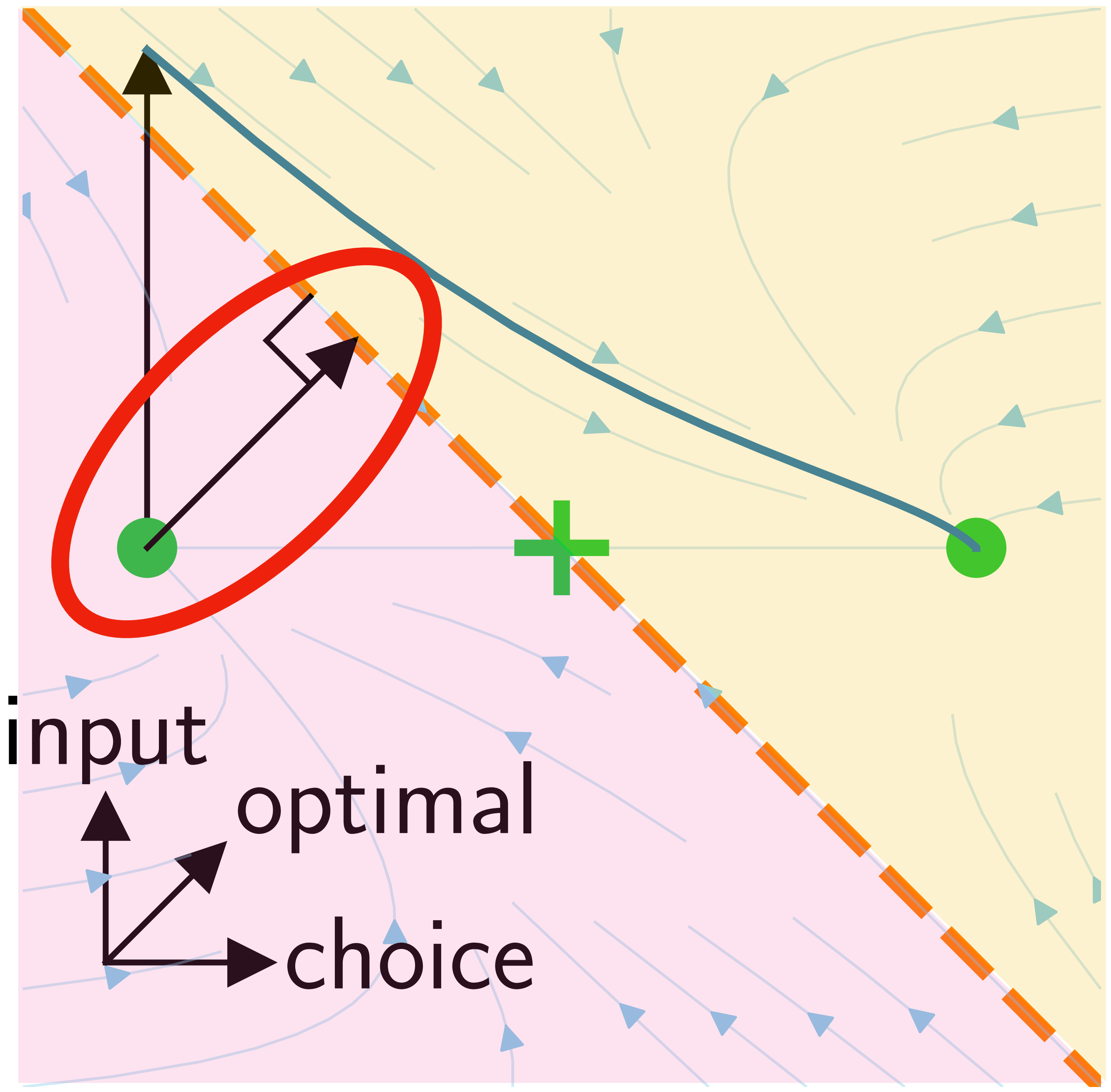
Data-trained RNN, $N = 668$



instantaneous pulse to neurons of your choice Δ

$$\Delta^* = \arg\min_{\Delta} \|\Delta\|_2^2 \quad \text{subject to} \quad |\psi(x_{\text{base}} + \Delta)| = 0.$$

Data-trained RNN, $N = 668$

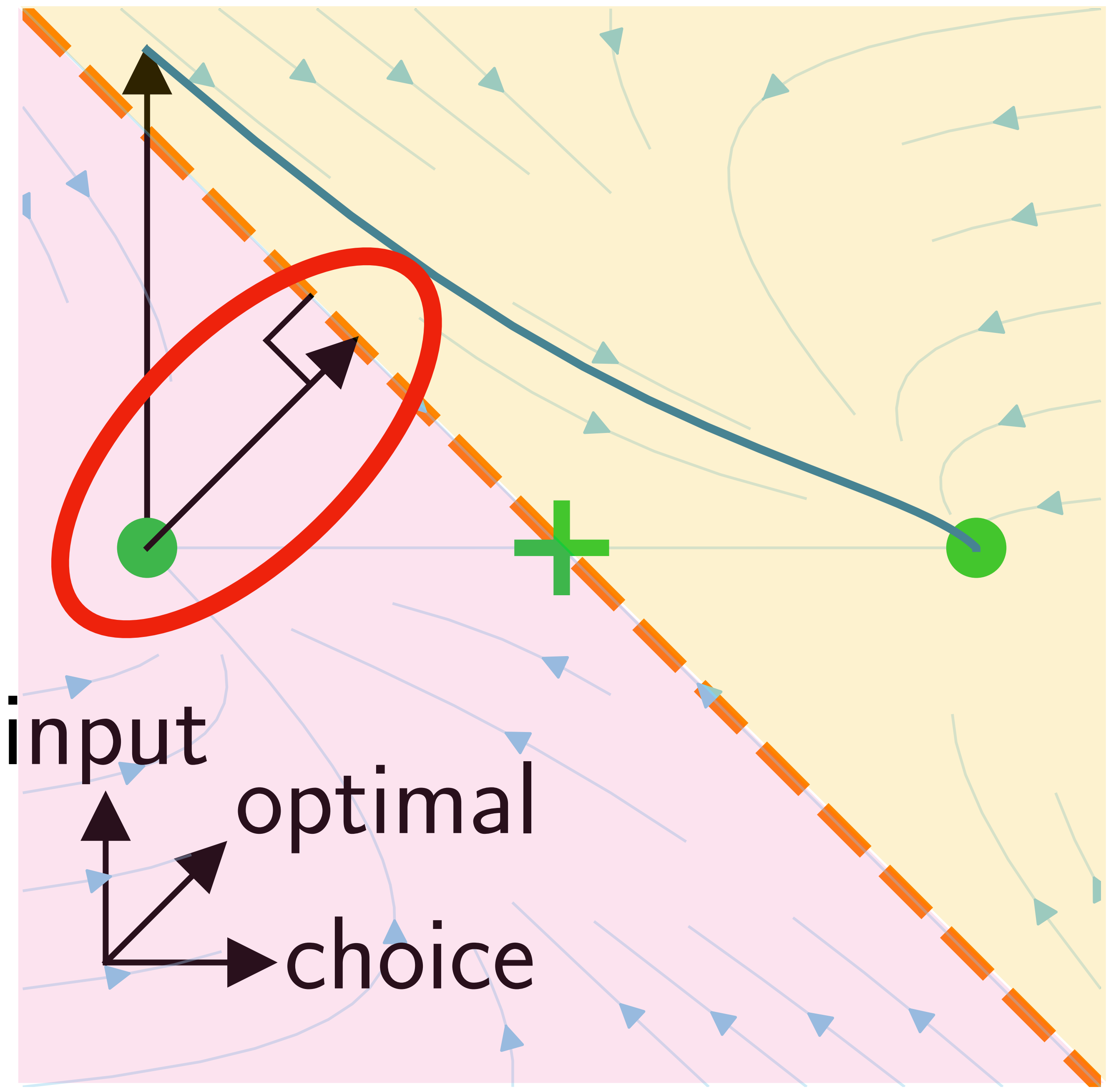


instantaneous pulse to neurons of your choice Δ

$$\Delta^* = \arg\min_{\Delta} \|\Delta\|_2^2 \quad \text{subject to} \quad |\psi(x_{\text{base}} + \Delta)| = 0.$$

minimise
norm of perturbation

Data-trained RNN, $N = 668$



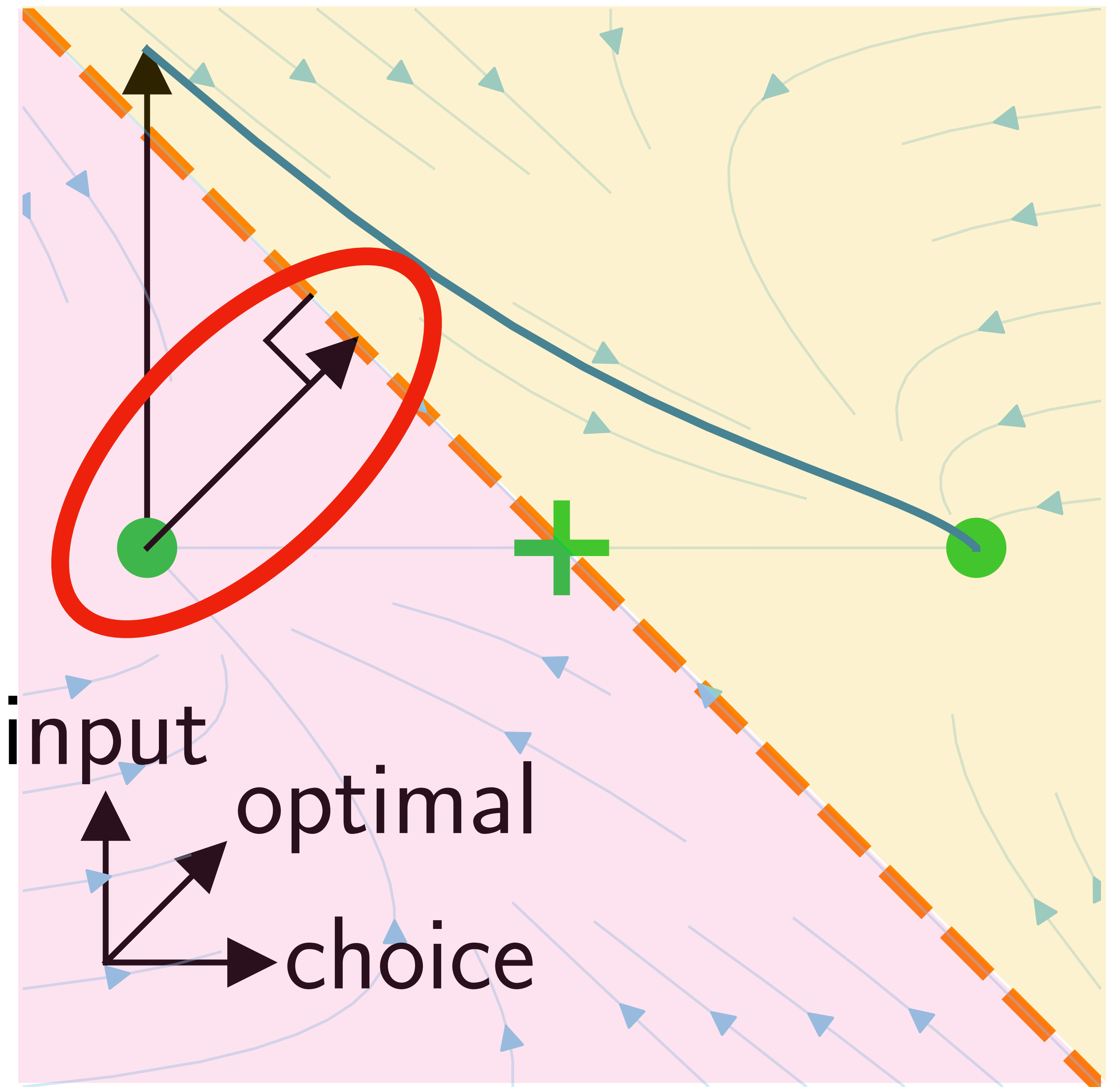
instantaneous pulse to neurons of your choice Δ

$$\Delta^* = \arg\min_{\Delta} \|\Delta\|_2^2 \quad \text{subject to} \quad |\psi(x_{\text{base}} + \Delta)| = 0.$$

minimise
norm of perturbation

final point on separatrix

Data-trained RNN, $N = 668$

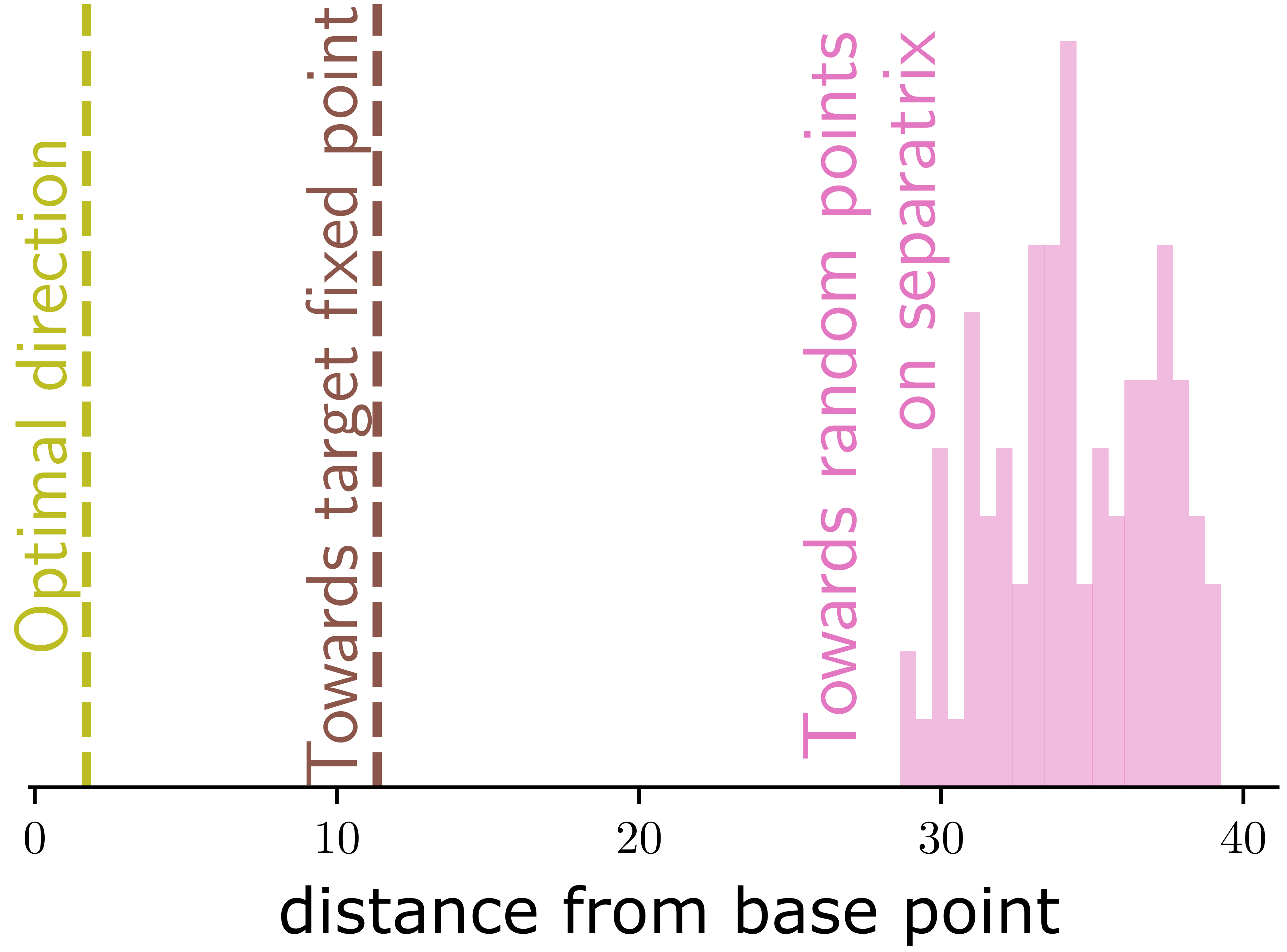


instantaneous pulse to neurons of your choice Δ

$$\Delta^* = \arg\min_{\Delta} \|\Delta\|_2^2 \quad \text{subject to} \quad |\psi(x_{\text{base}} + \Delta)| = 0.$$

minimise norm of perturbation

final point on separatrix



Summary

Summary

- Propose a framework to find separatrices in high dimensional nonlinear dynamical systems

Summary

- Propose a framework to find separatrices in high dimensional nonlinear dynamical systems
- How? Map the system to a canonical 1D system

Summary

- Propose a framework to find separatrices in high dimensional nonlinear dynamical systems
- How? Map the system to a canonical 1D system
- Demonstrate the method on variety of systems including data-trained RNNs

Summary

- Propose a framework to find separatrices in high dimensional nonlinear dynamical systems
- How? Map the system to a canonical 1D system
- Demonstrate the method on variety of systems including data-trained RNNs
- Use it to design optimal perturbations

Thanks!

Discussions: Matthijs Pals, Yoav Ger, Aviv Ratzon

PhD Advisor and co-author: Omri Barak

https://github.com/KabirDabholkar/separatrix_locator

Try it yourself! Reach out to us!

Thanks!

Discussions: Matthijs Pals, Yoav Ger, Aviv Ratzon

PhD Advisor and co-author: Omri Barak

https://github.com/KabirDabholkar/separatrix_locator

Try it yourself! Reach out to us!