2- Questions and Discussion

Data Science and AI for Neuroscience Summer School 2022, Chen Institute at Caltech

Chethan Pandarinath, Ph.D.
(he/him)
Assistant Professor
Coulter Department of Biomedical Engineering
Department of Neurosurgery
Emory University & Georgia Tech

@chethan

Christopher Versteeg, Ph.D.
Postdoctoral Fellow
Coulter Department of Biomedical Engineering
Emory University & Georgia Tech

SYSTEMS NEURAL ENGINEERING LABORATORY
snel.gatech.edu

EMORY UNIVERSITY

Georgia Tech

What you'll hear about in this lecture

What you'll hear about in this lecture

Neural network basics

What you'll hear about in this lecture

Neural network basics

Deep autoencoders

What you'll hear about in this lecture

Neural network basics

Deep autoencoders

Intro to neural population dynamics

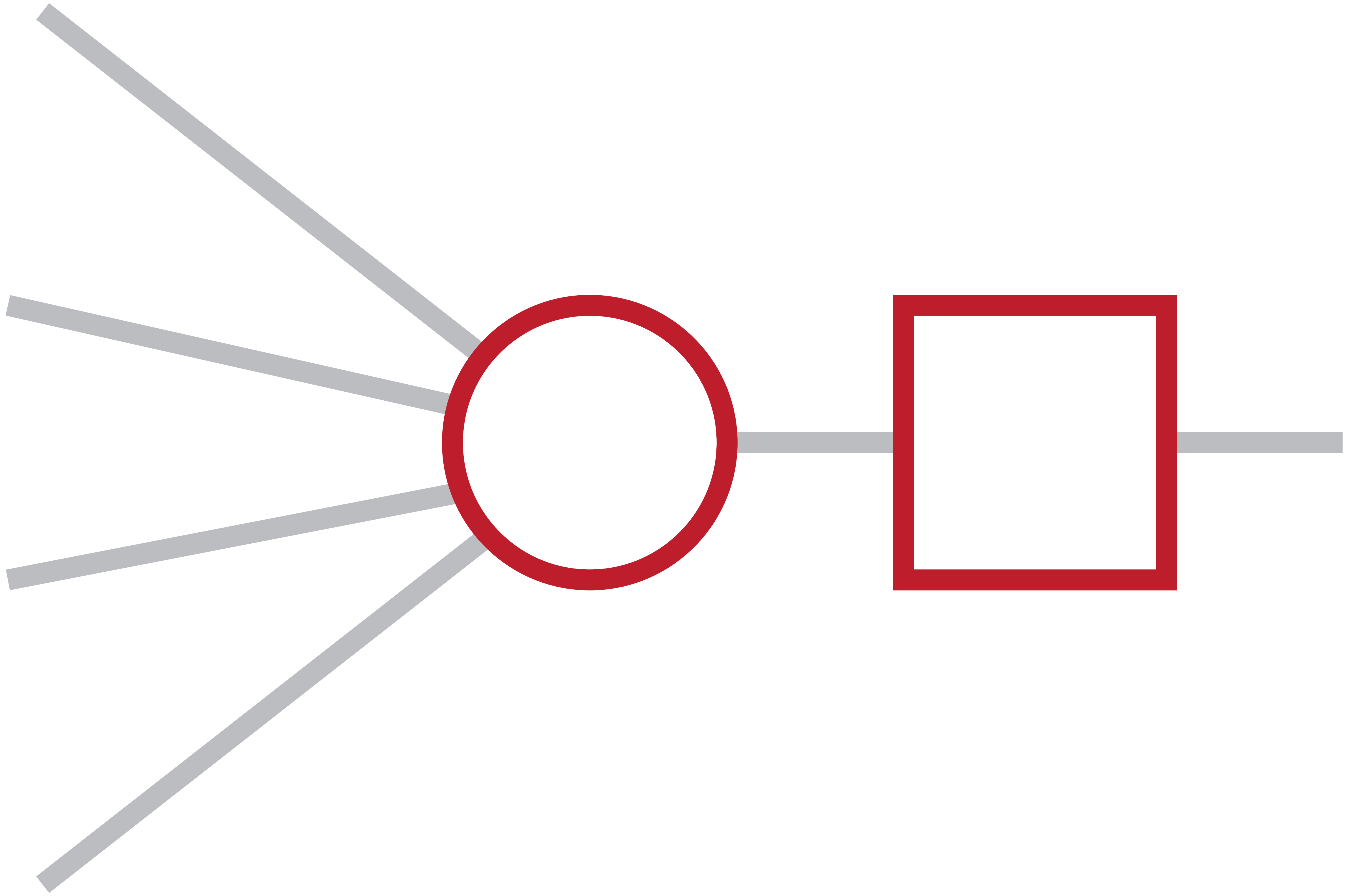
What you'll hear about in this lecture

Neural network basics

Deep autoencoders

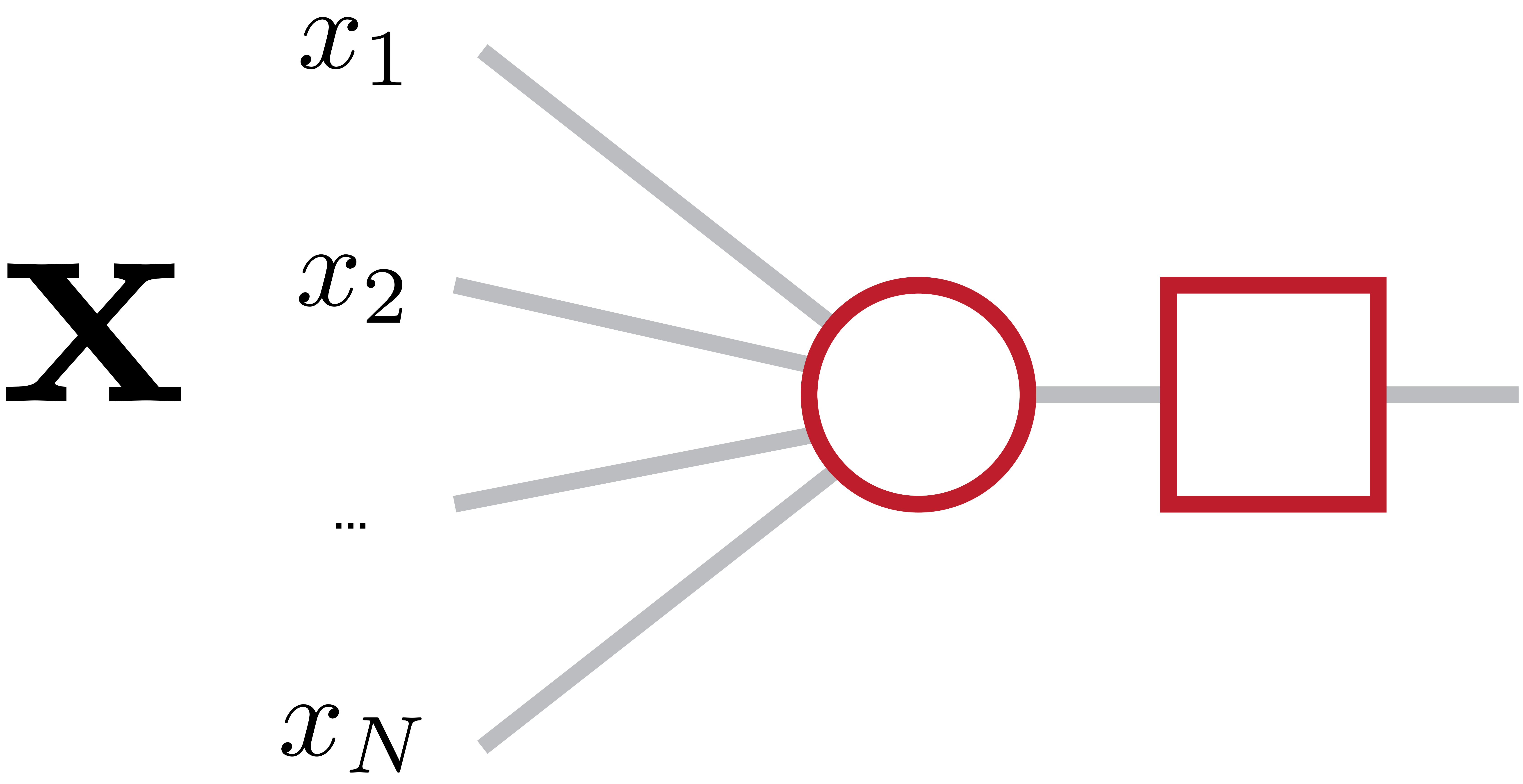
Intro to neural population dynamics

Artificial neurons

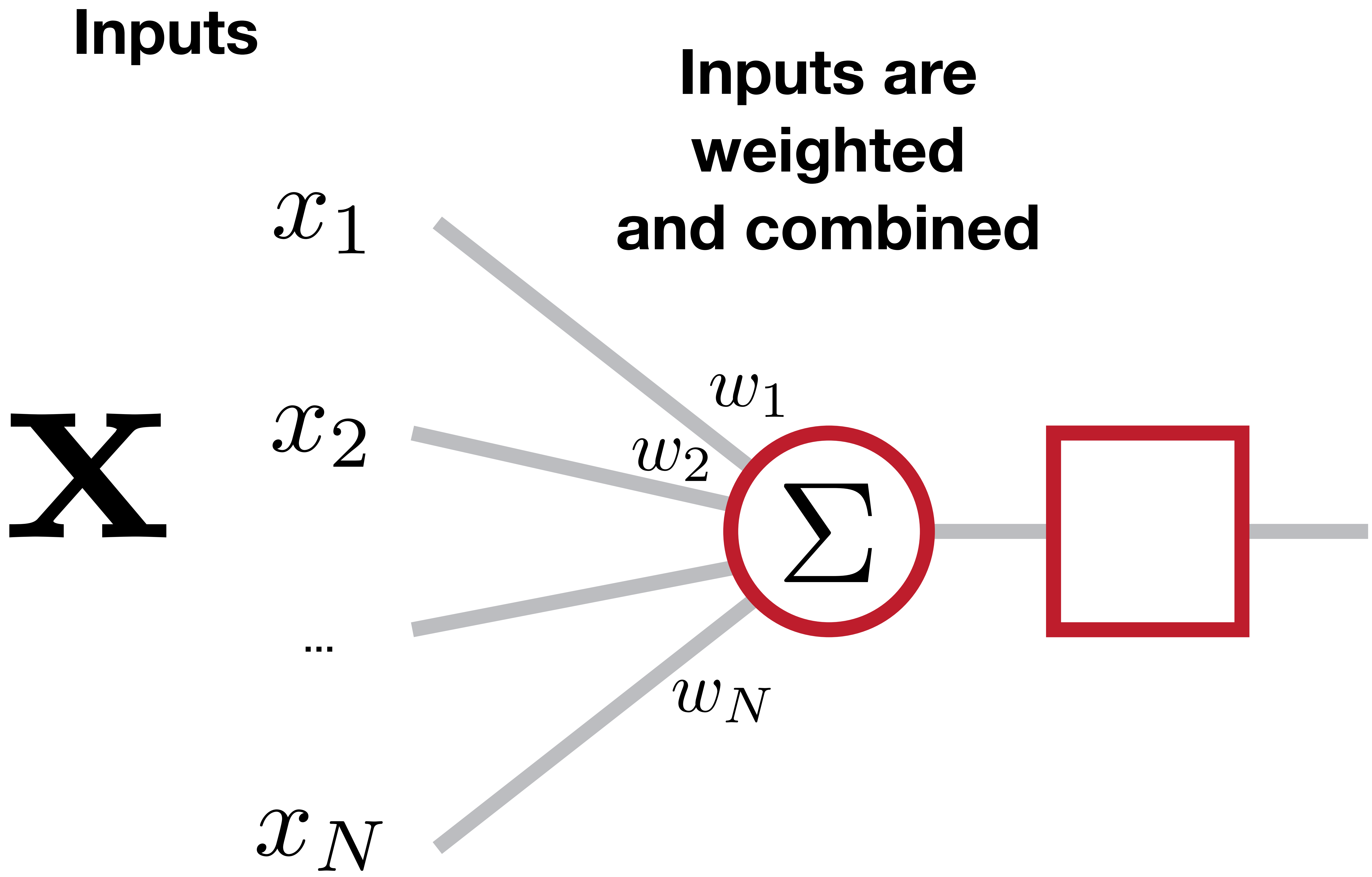


Artificial neurons

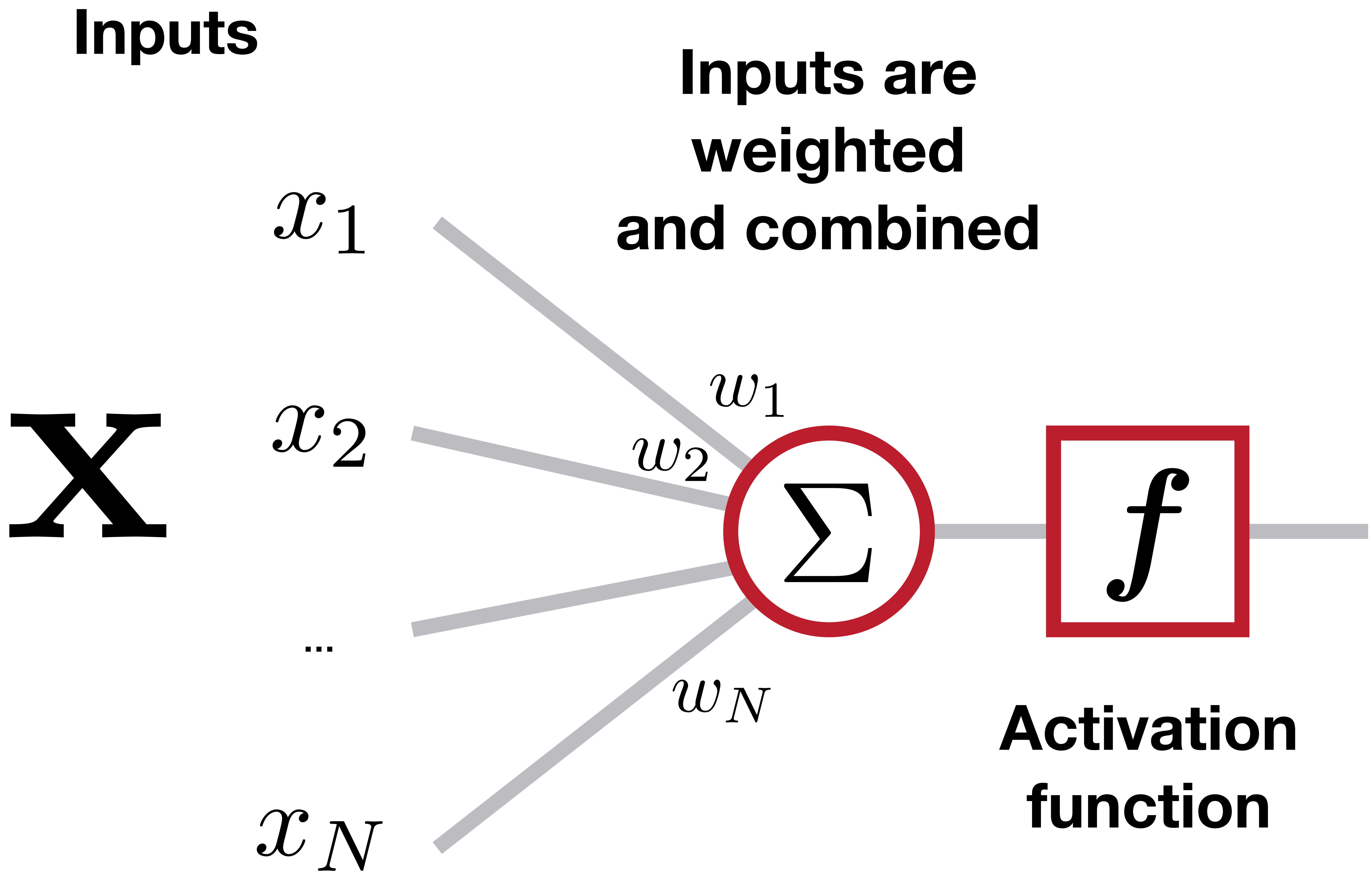
Inputs



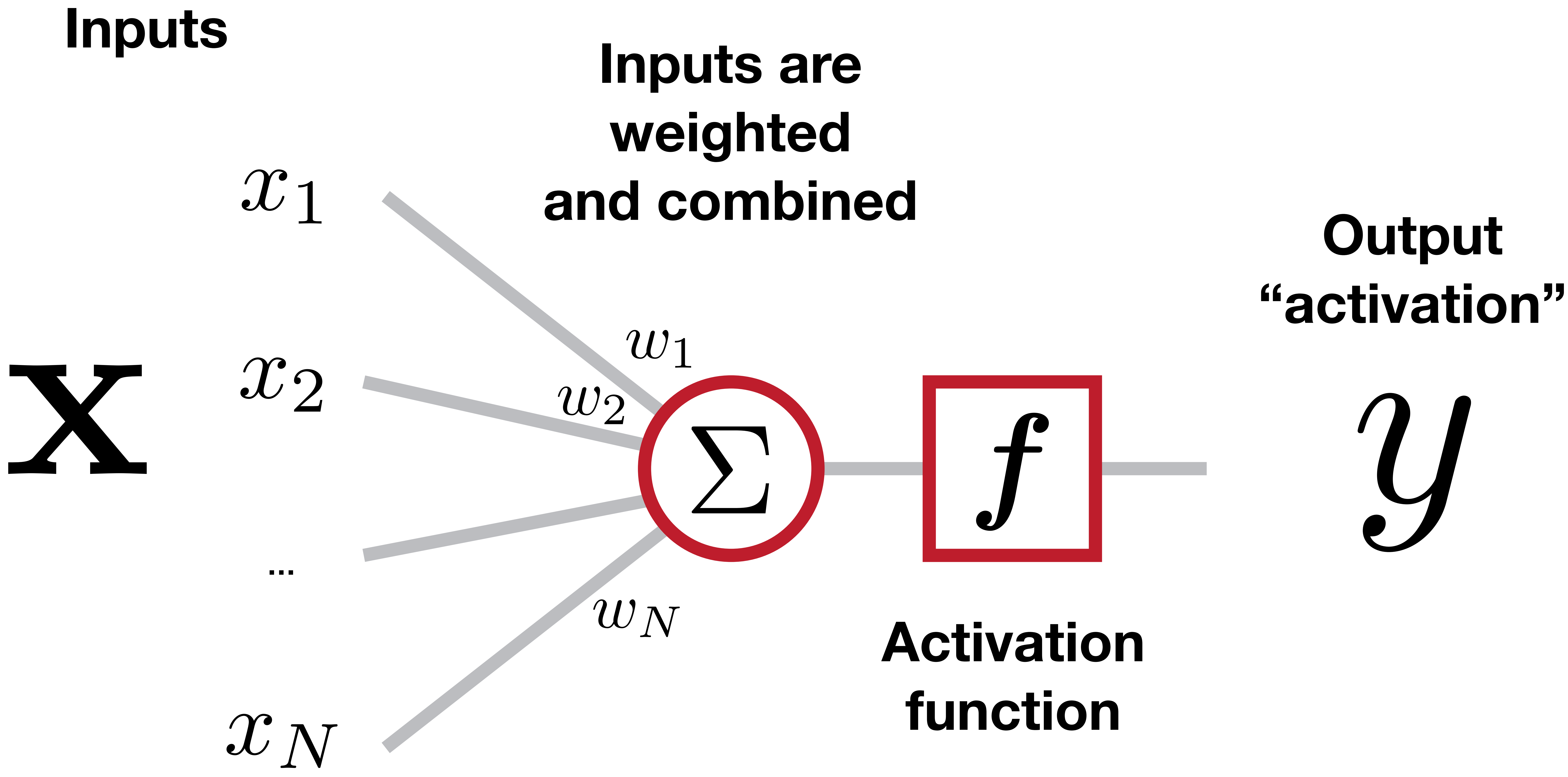
Artificial neurons



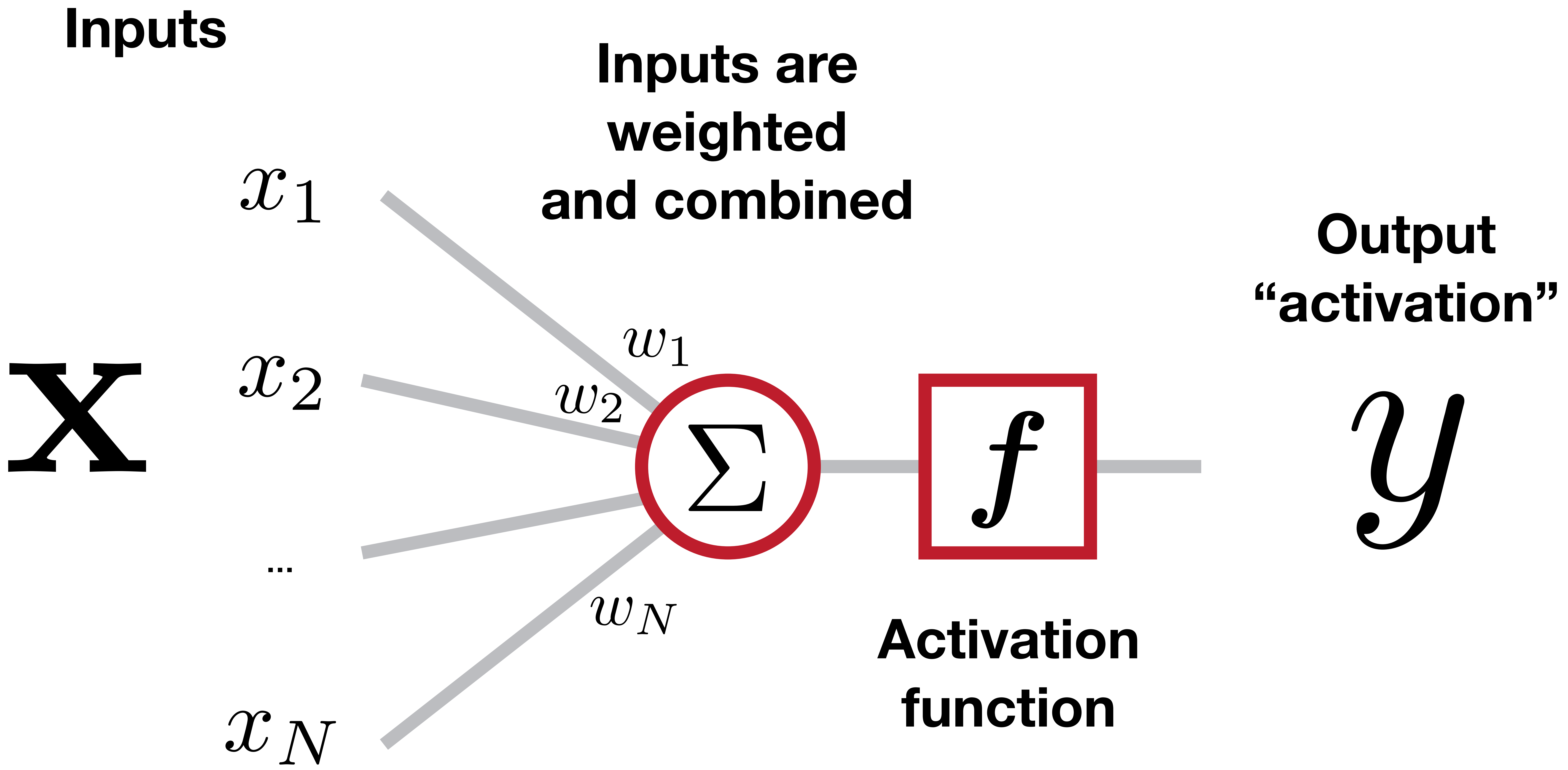
Artificial neurons



Artificial neurons

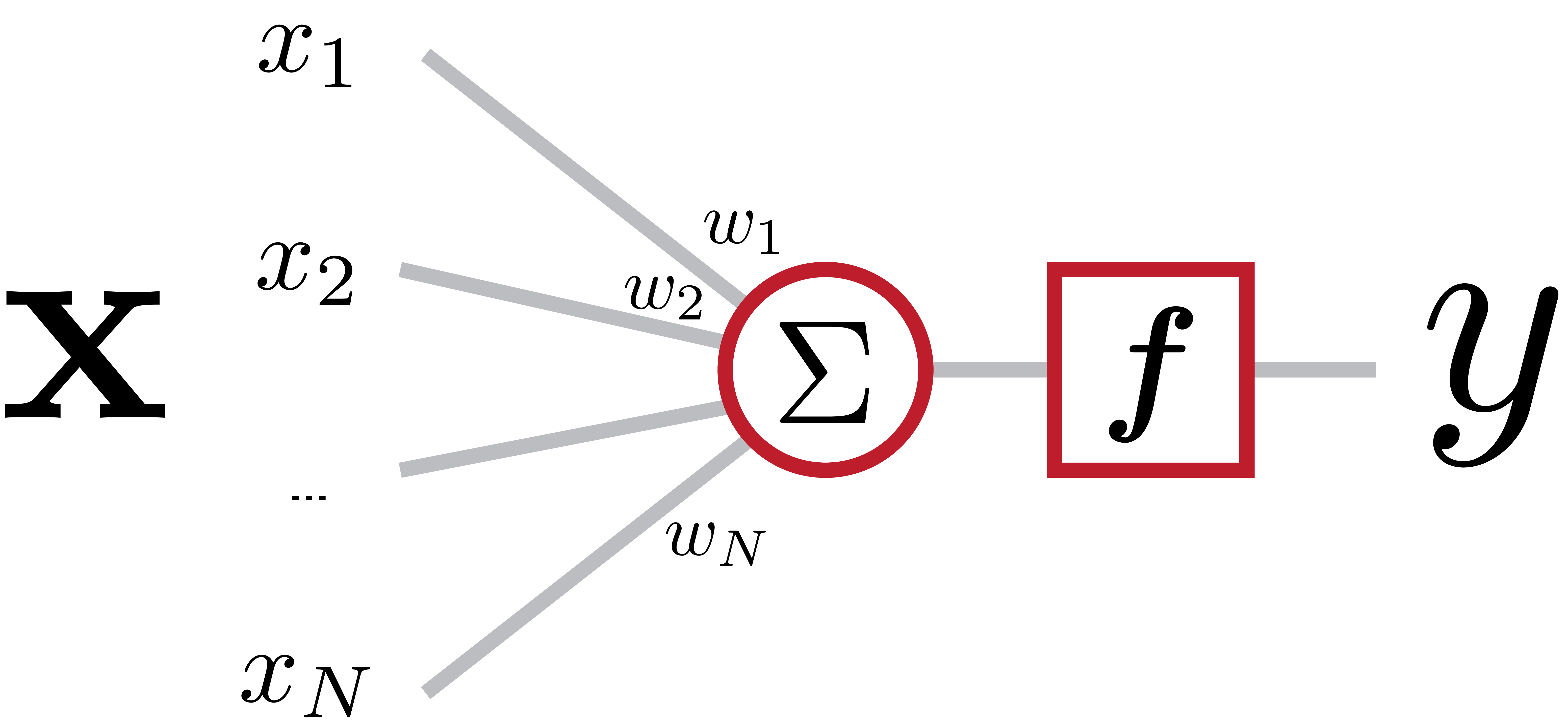


Artificial neurons



Warren S. McCulloch and Walter Pitts (1943)
"A logical calculus of the ideas immanent in nervous activity.”
Bulletin of Mathematical Biophysics 5:115-133

Artificial neurons

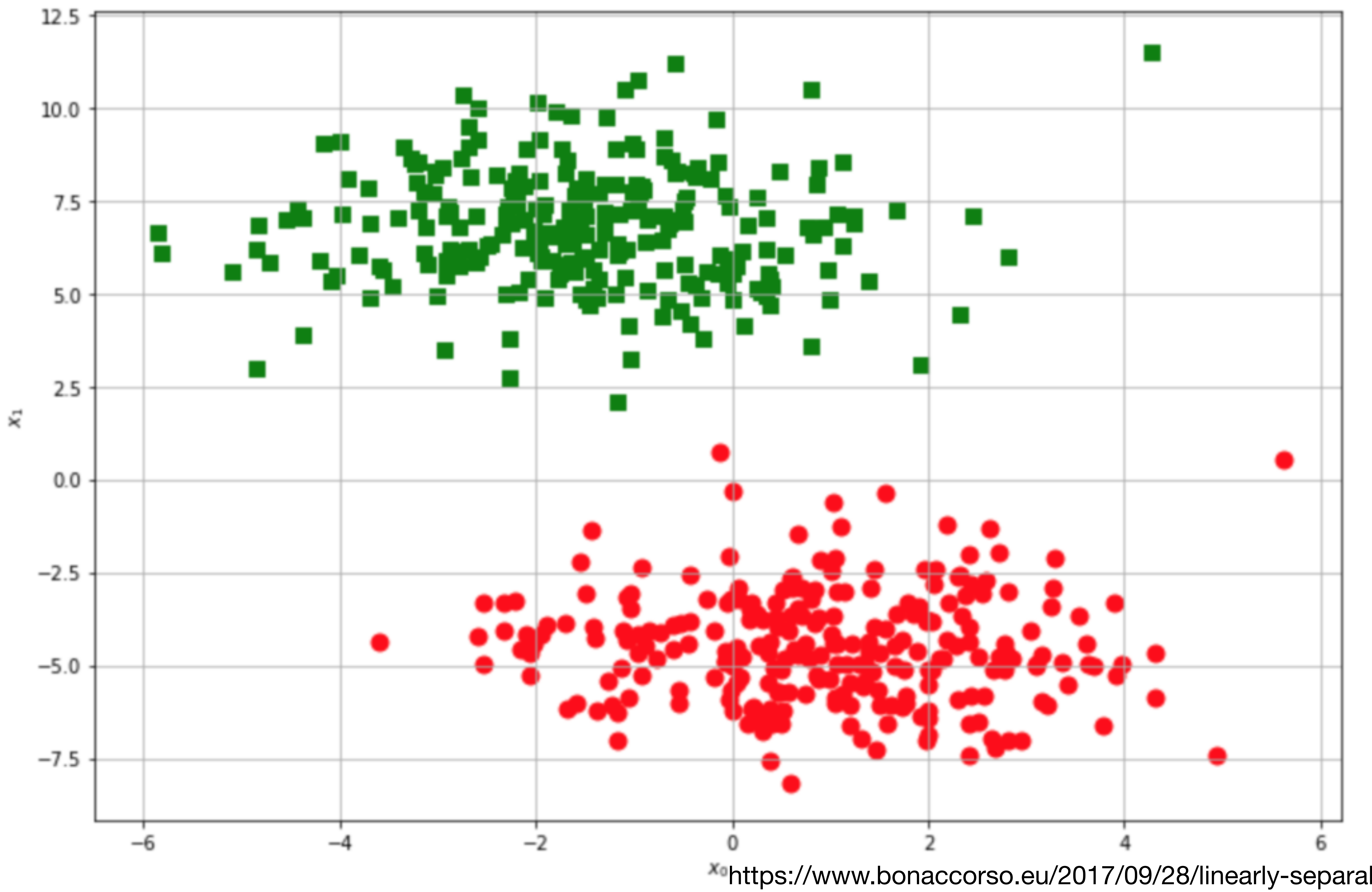


$$y = f(w_1 * x_1 + w_2 * x_2 + \ldots + b)$$

Simple problem - 2 dimensional classification

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

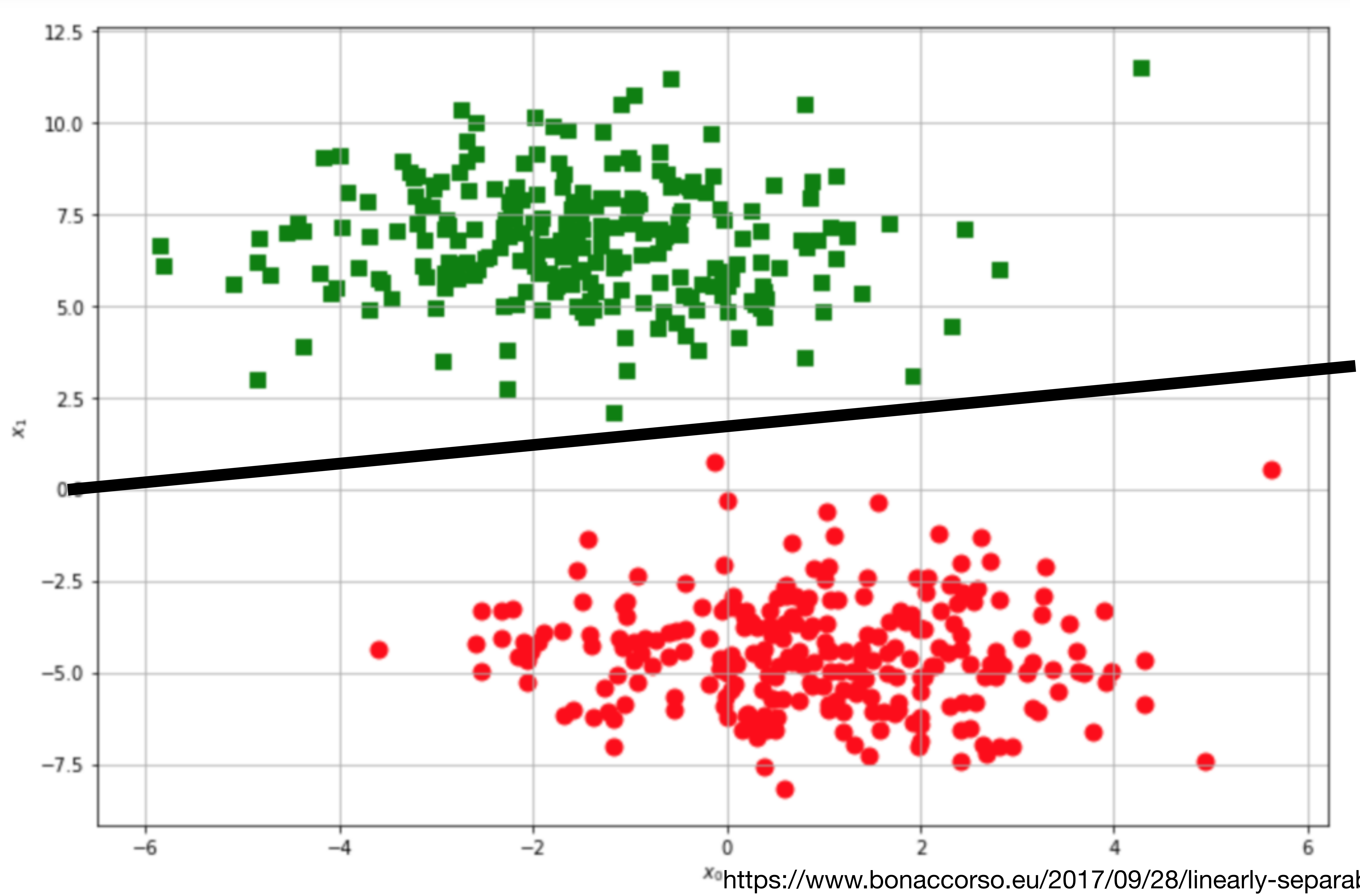
Green: class 0
Red: class 1



https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Simple problem - 2 dimensional classification

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

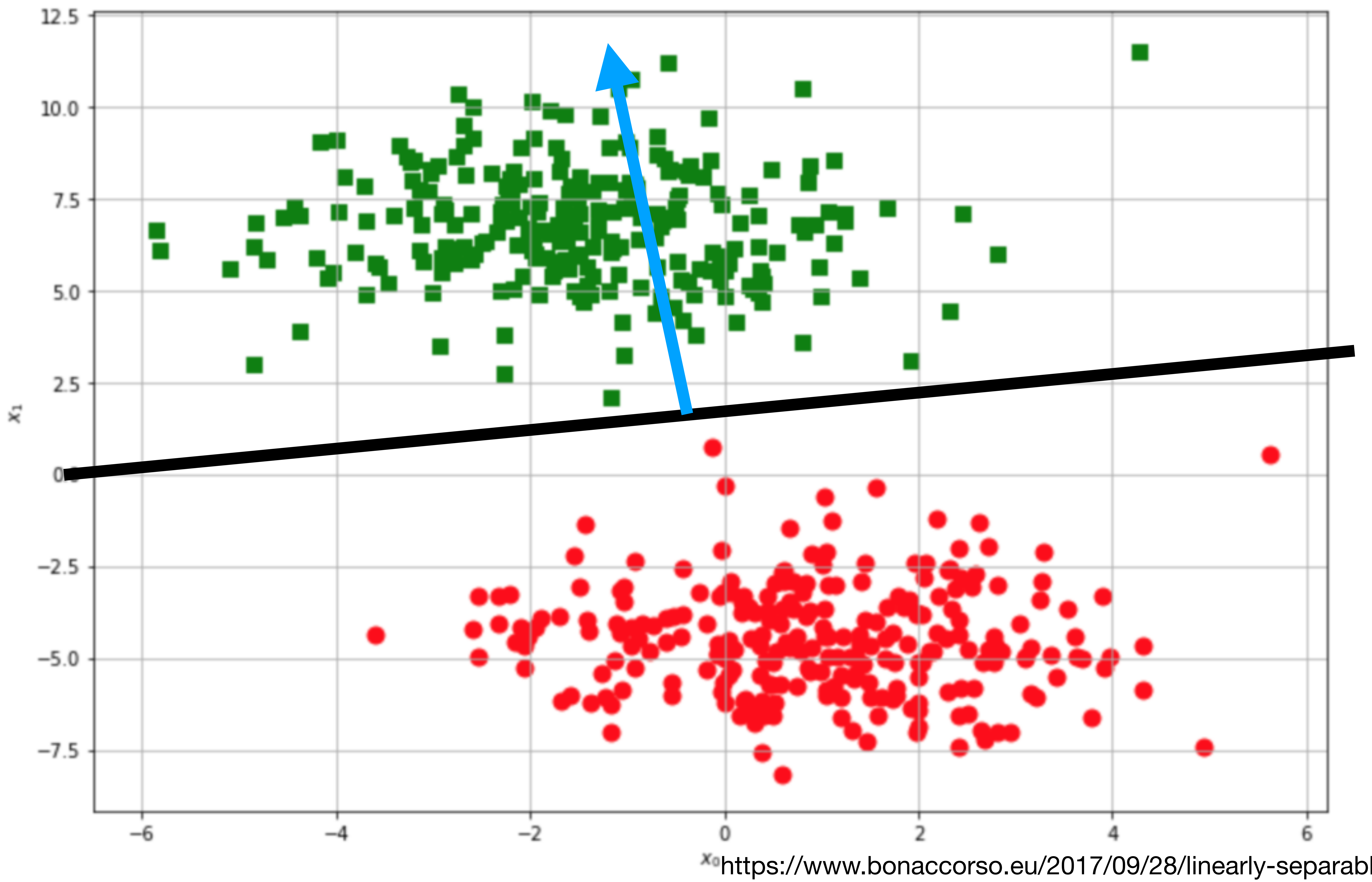


Green: class 0
Red: class 1

https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Simple problem - 2 dimensional classification

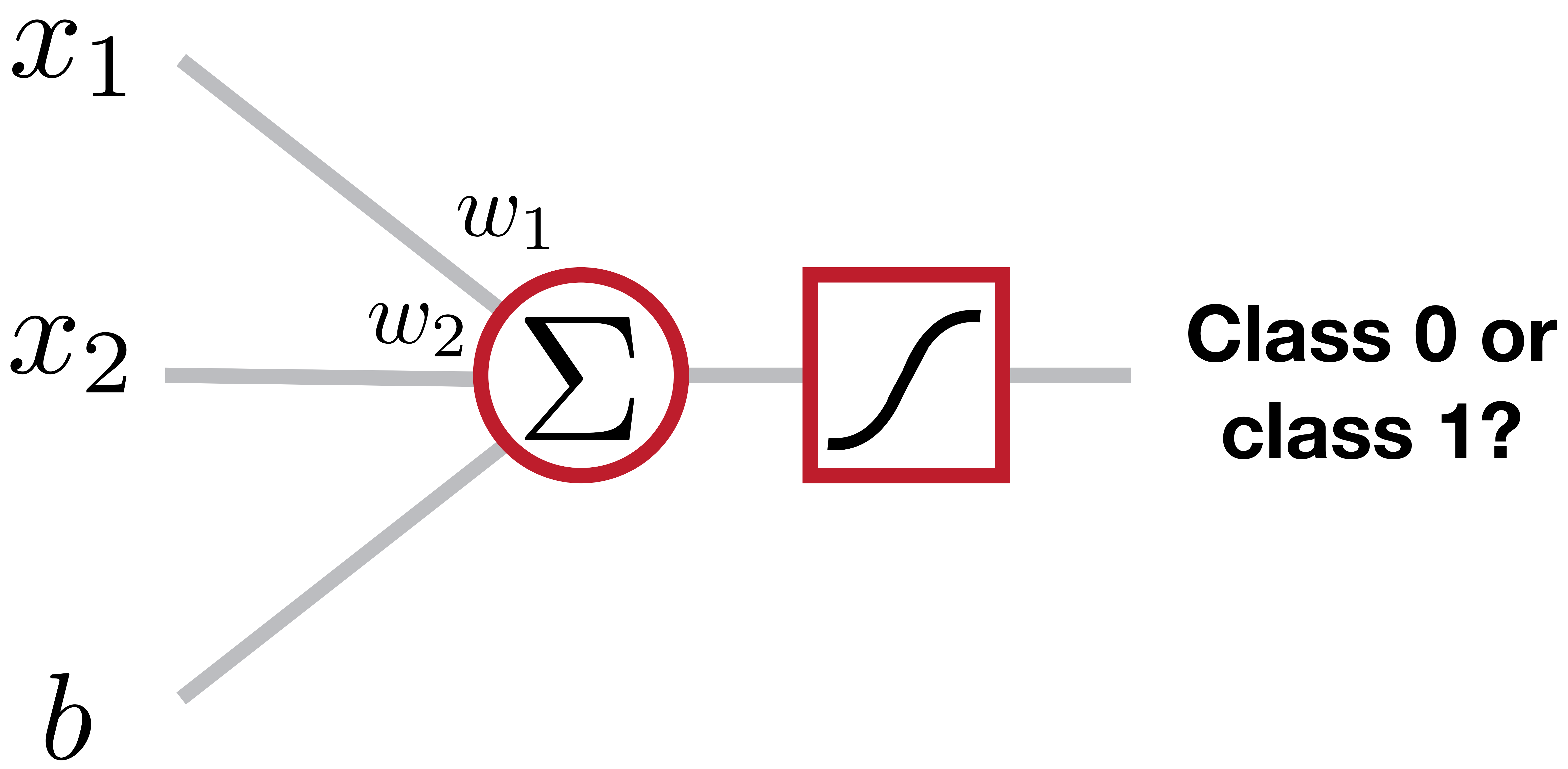
$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



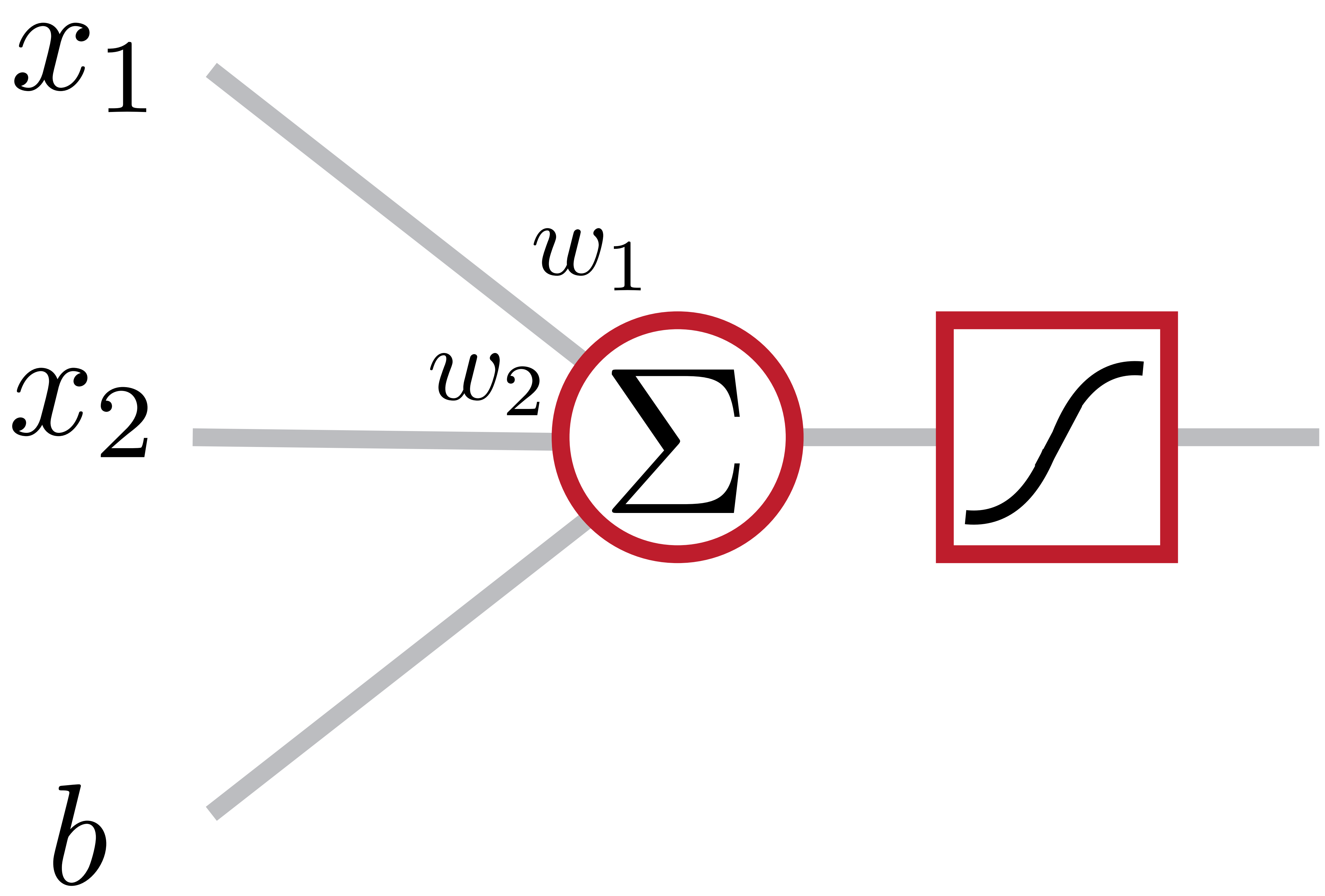
Green: class 0
Red: class 1

https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Simple classifier - Perceptron



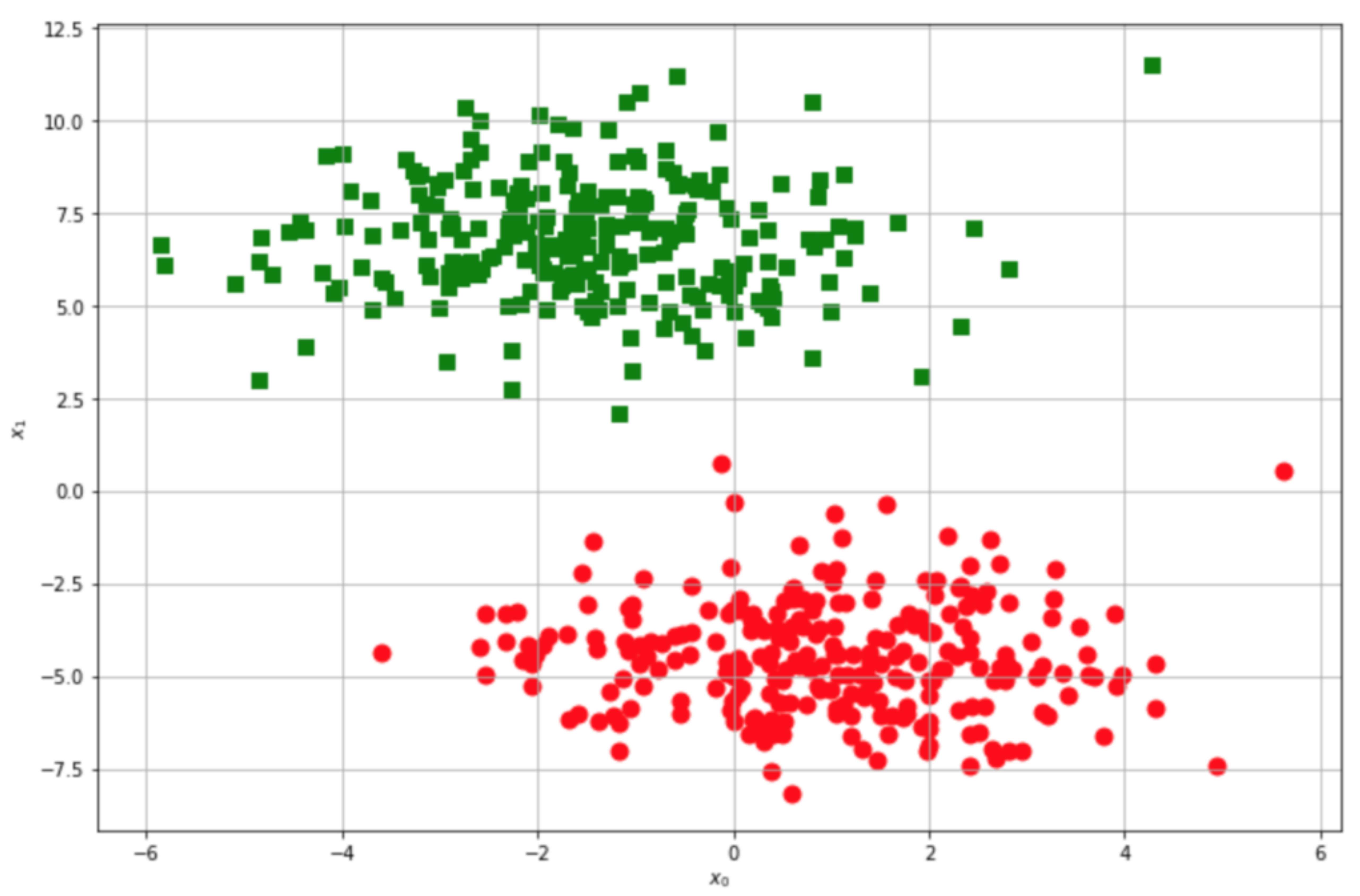
Simple classifier - Perceptron

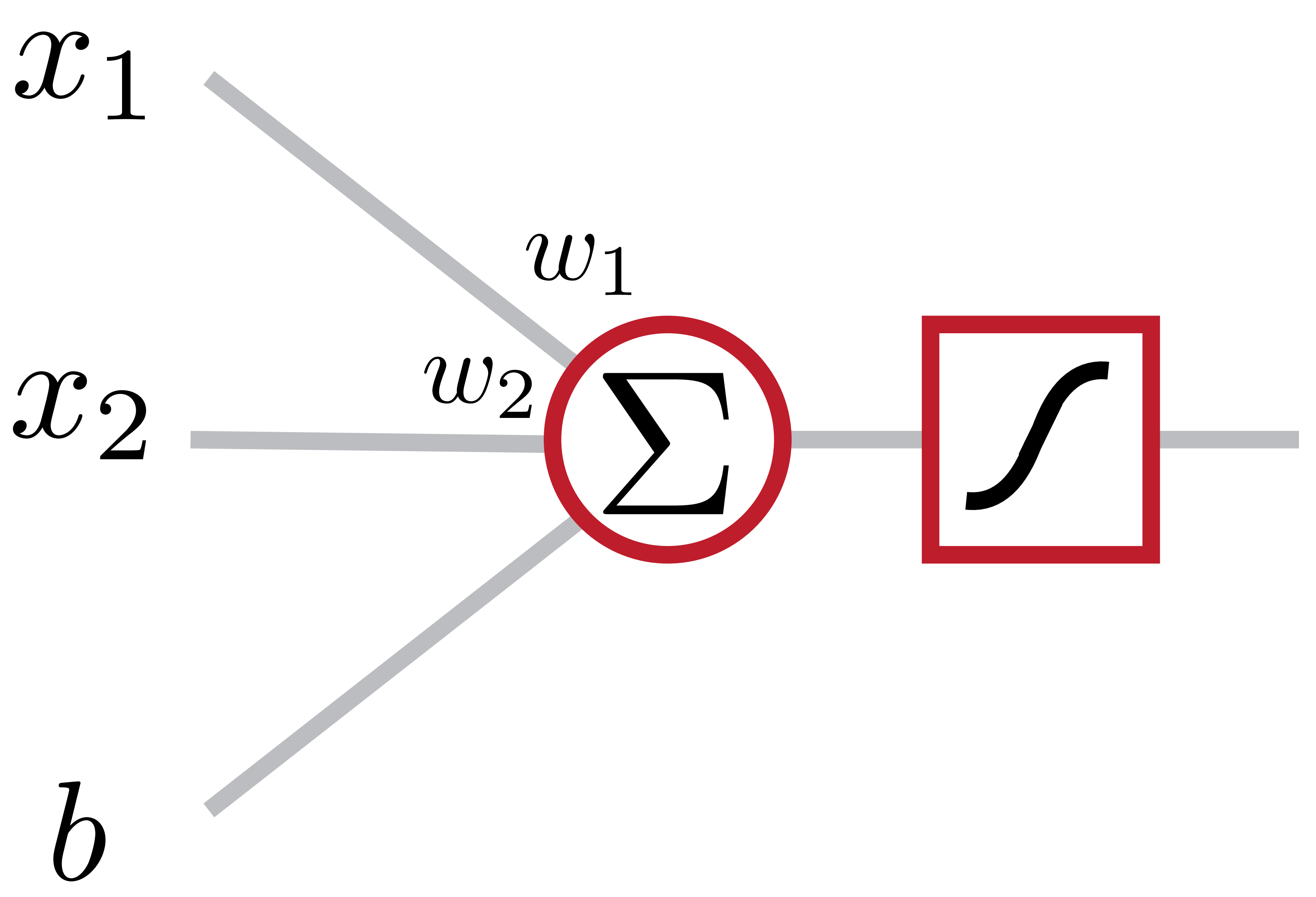


Class 0 or class 1?

Frank Rosenblatt (1957)
"The Perceptron — a perceiving and recognizing automaton."
Report 85–460–1, Cornell Aeronautical Laboratory.

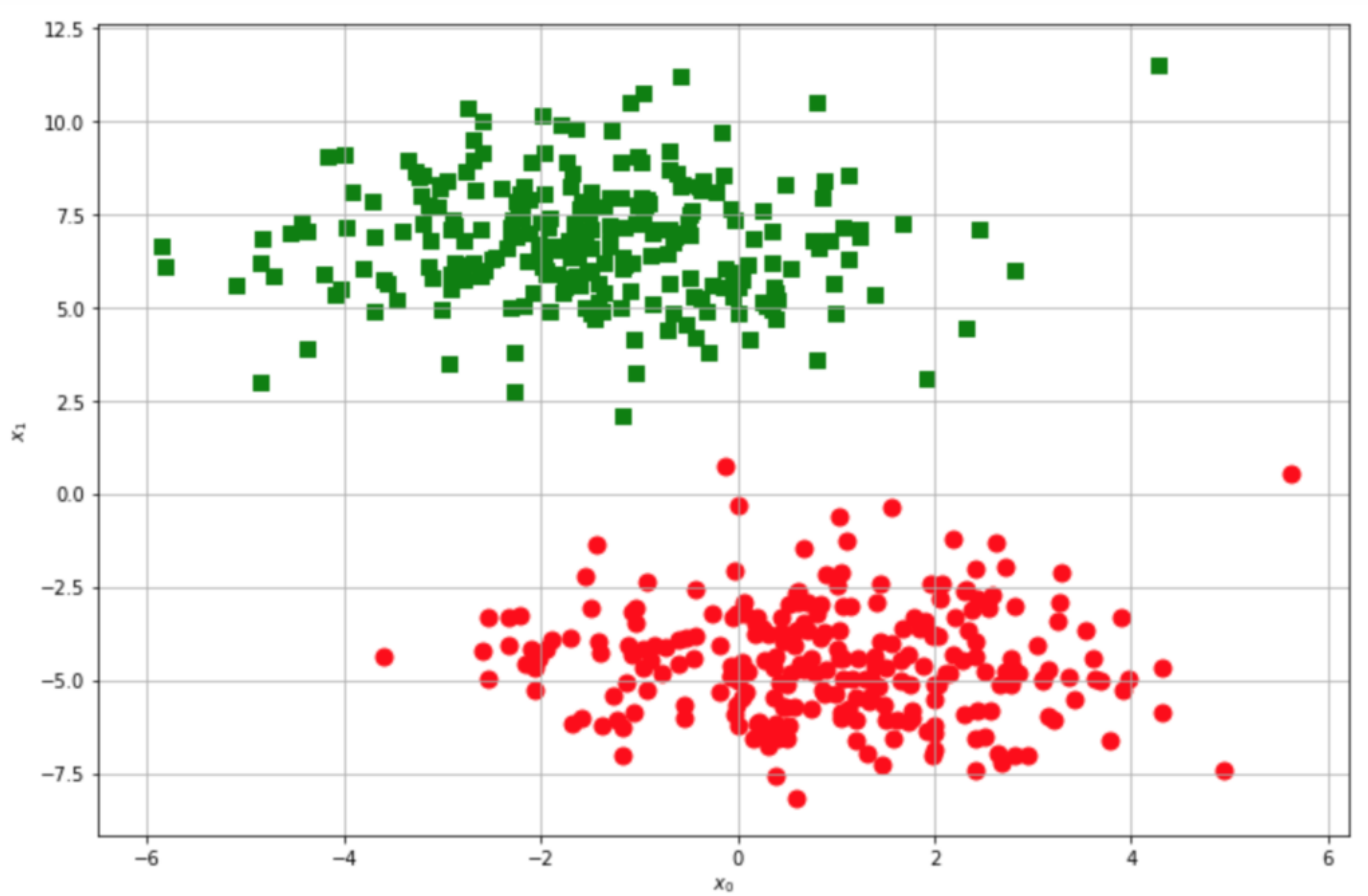
Classification with the perceptron

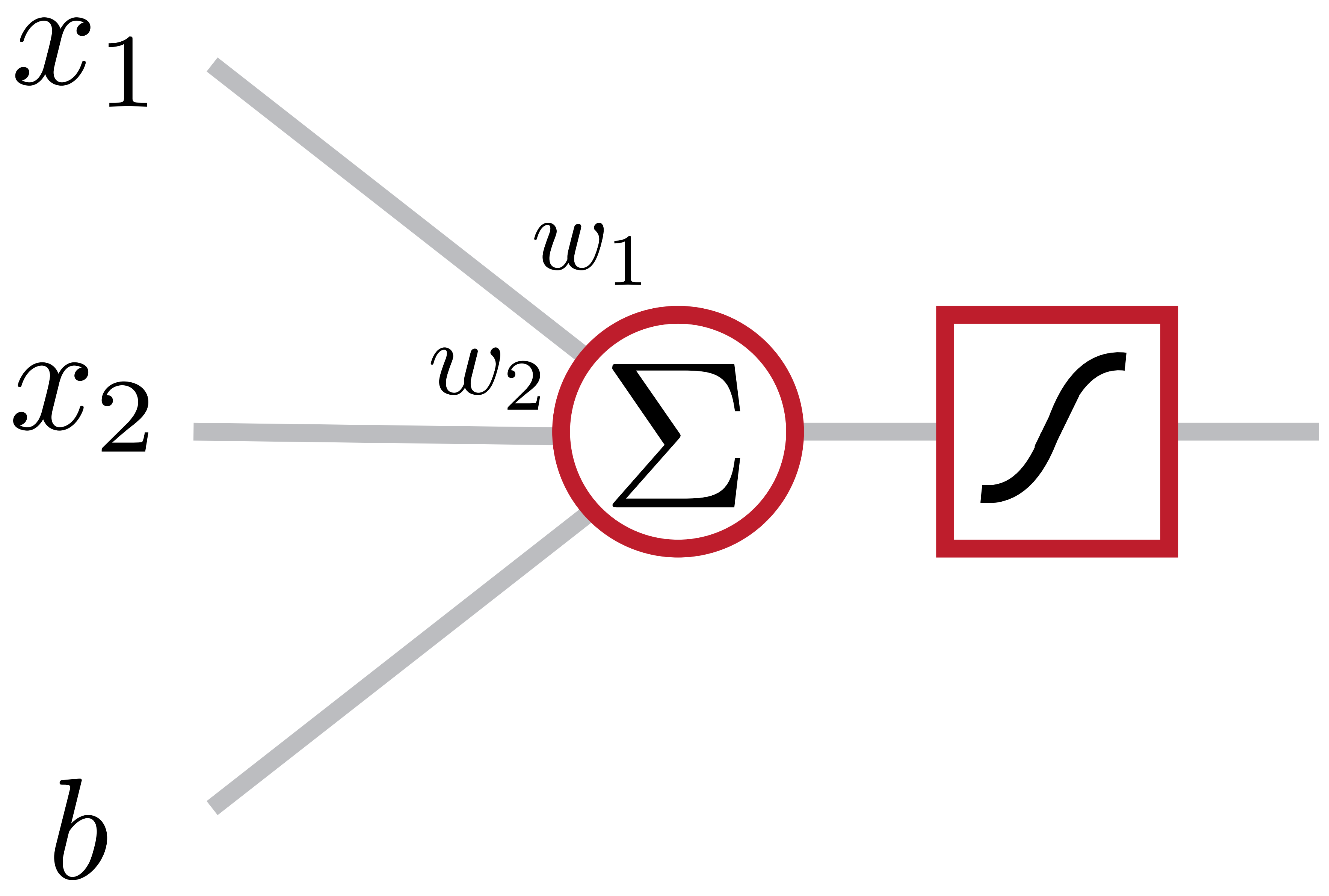




https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Classification with the perceptron



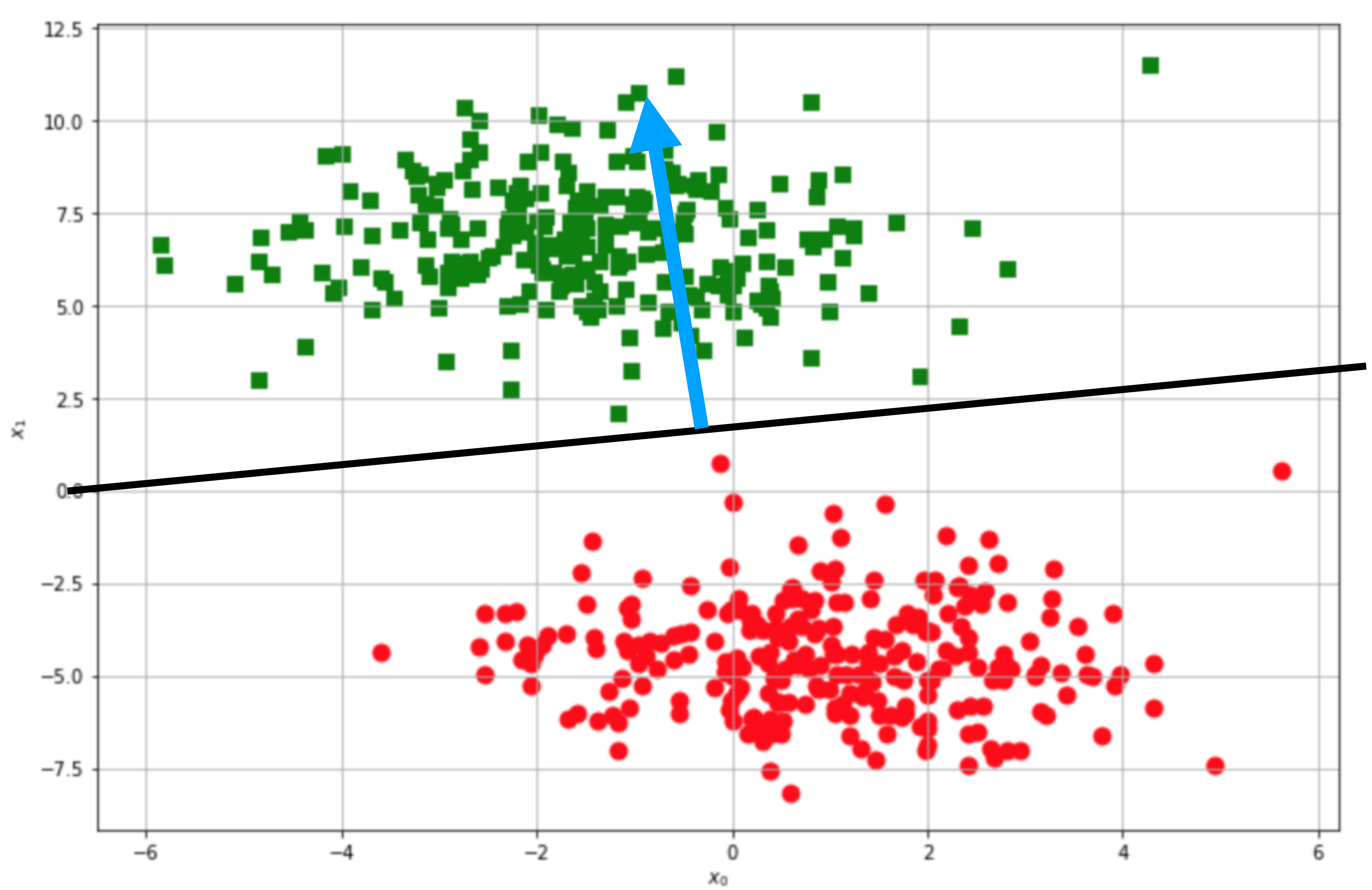


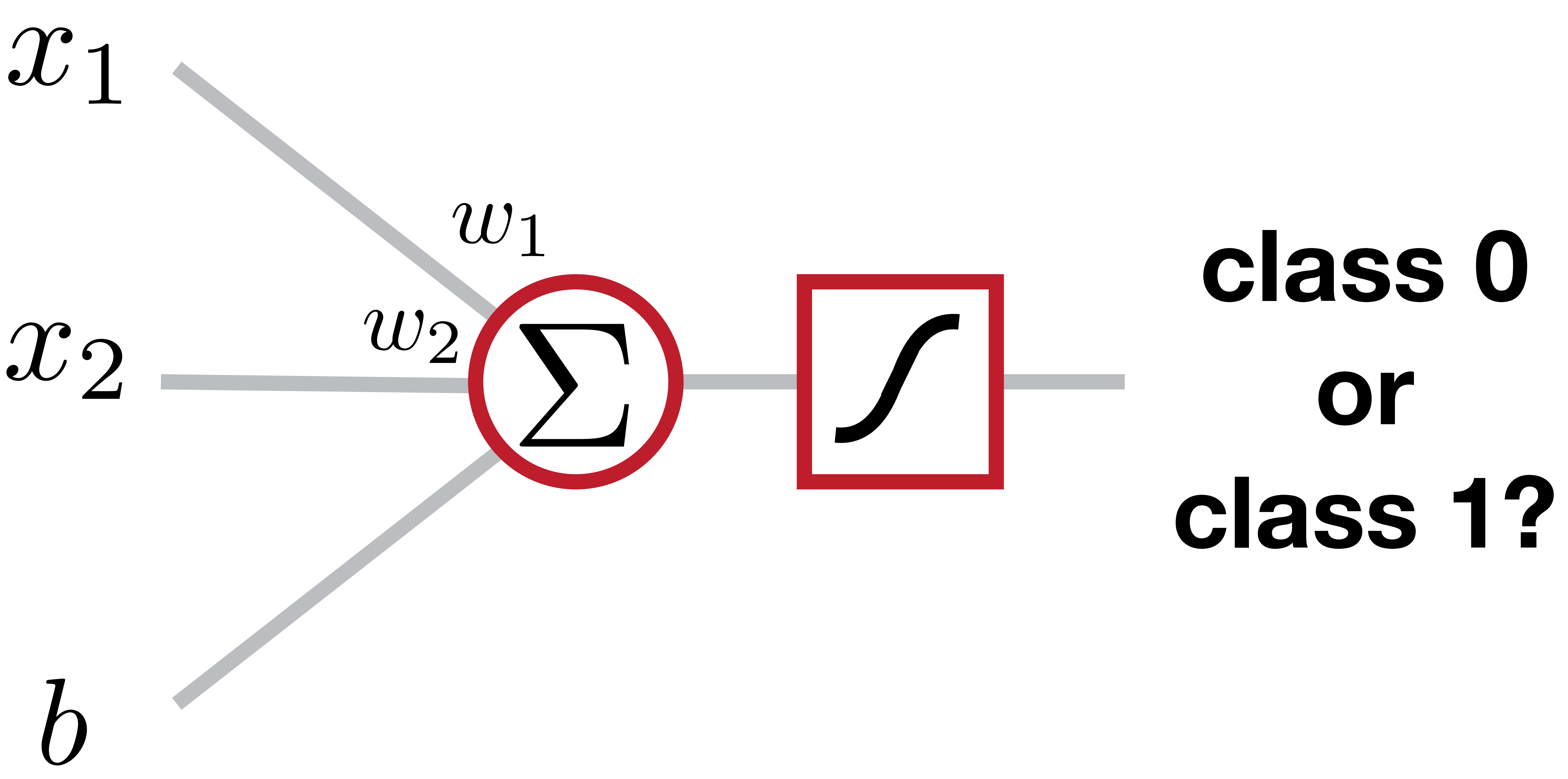
class 0
or
class 1?

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \bullet \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + b$$

https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Classification with the perceptron

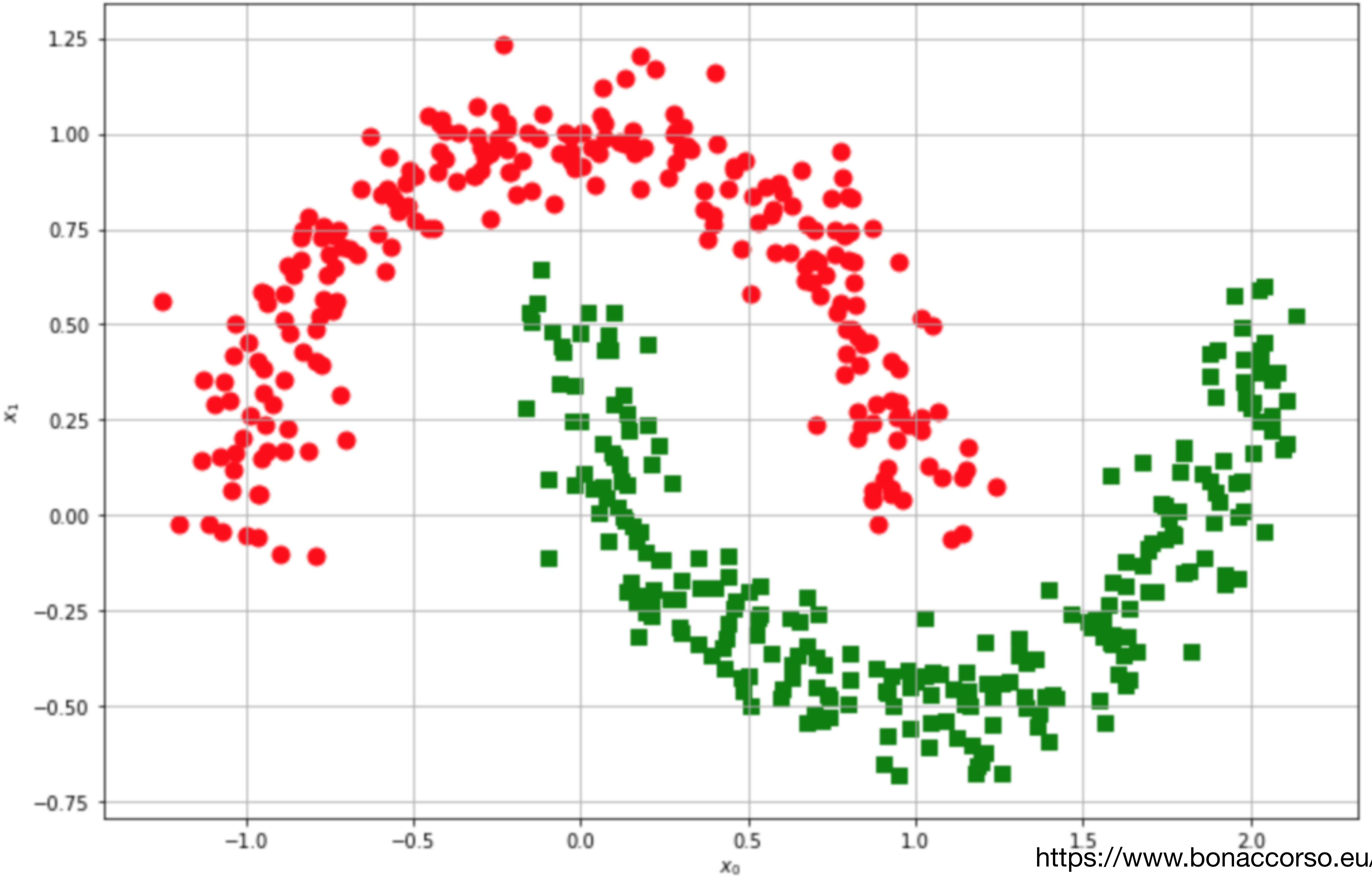




$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \bullet \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + b$$

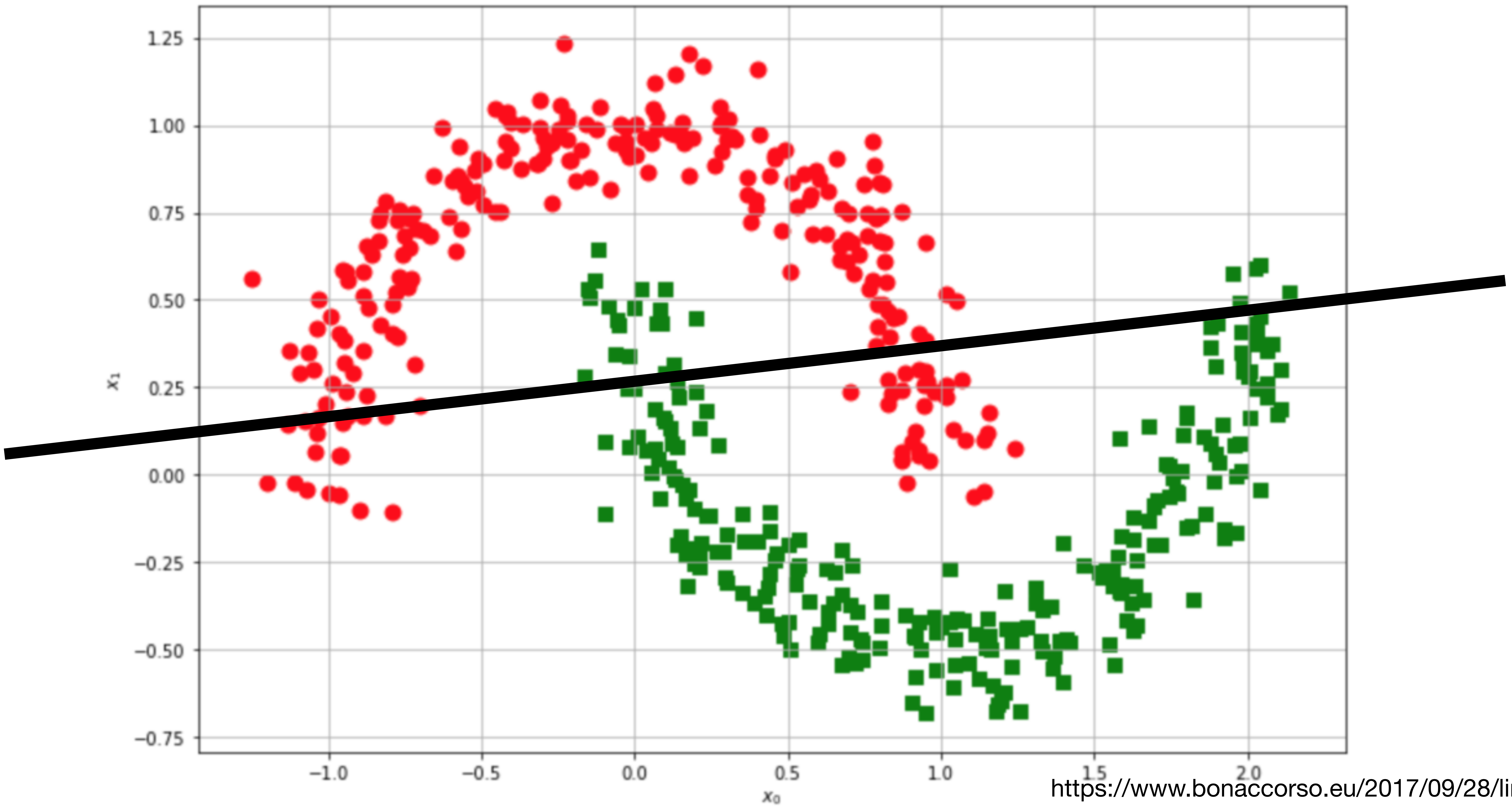
https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Harder problem



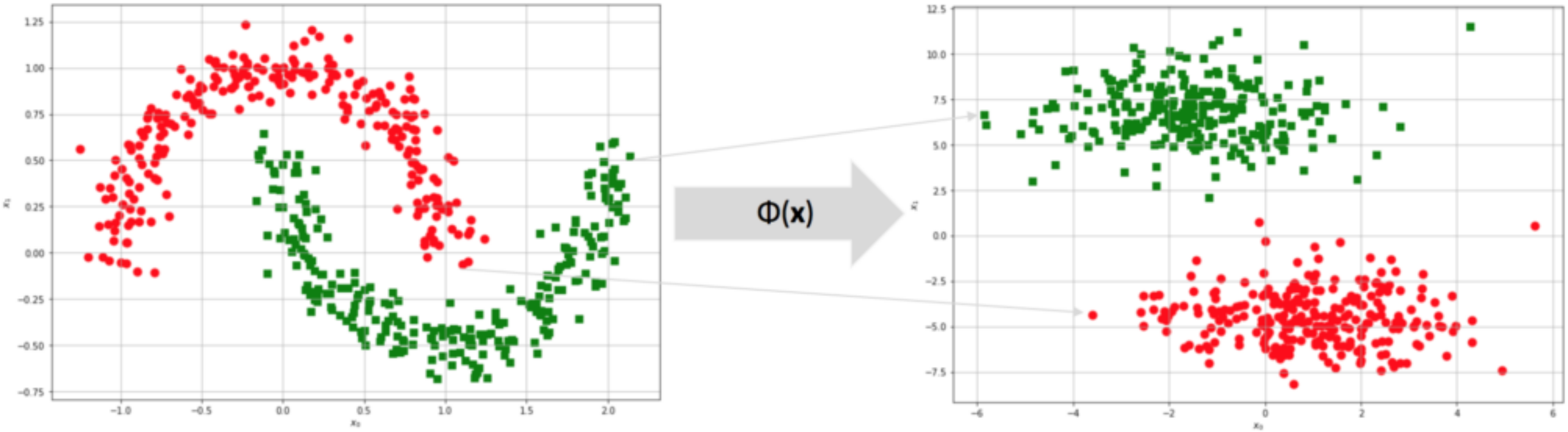
https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Harder problem



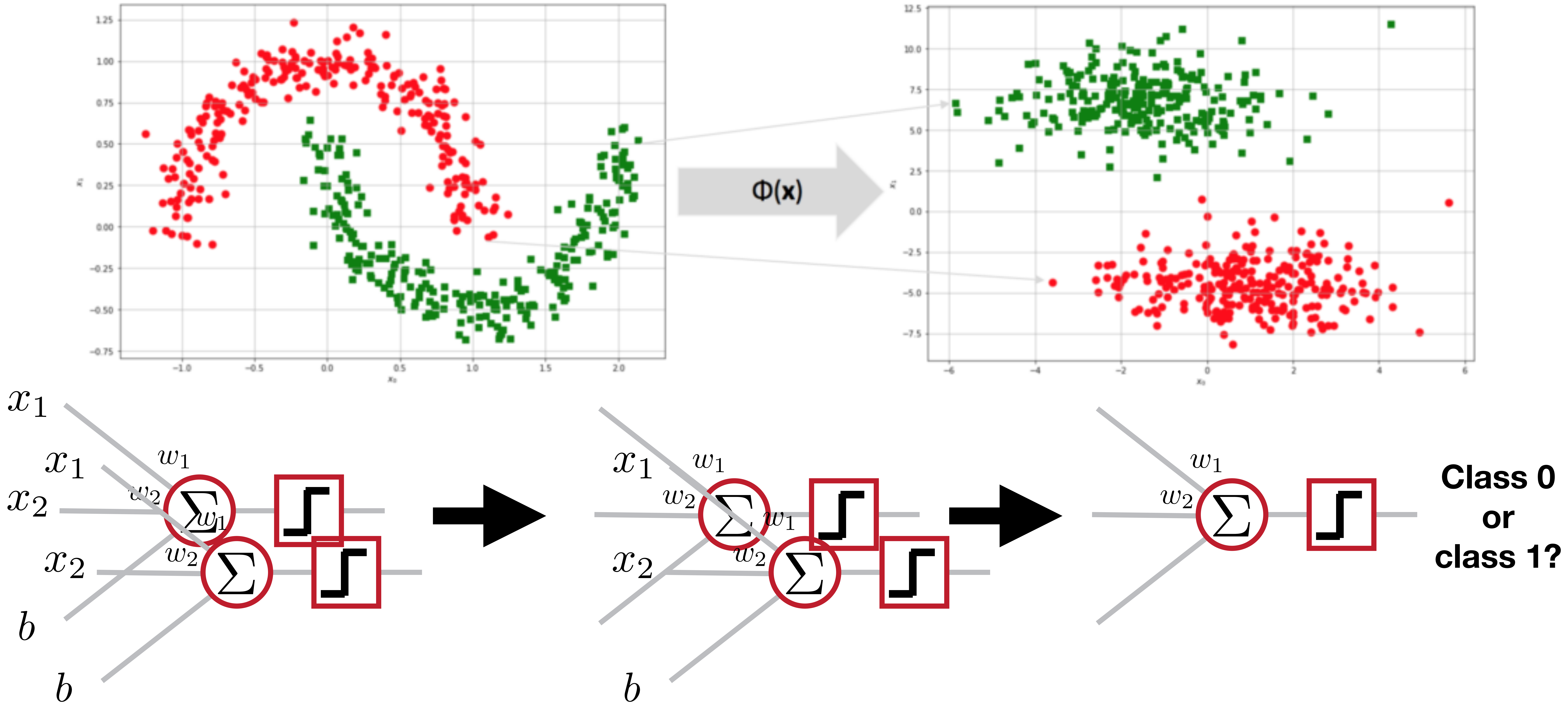
https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Strategy - multiple transformations

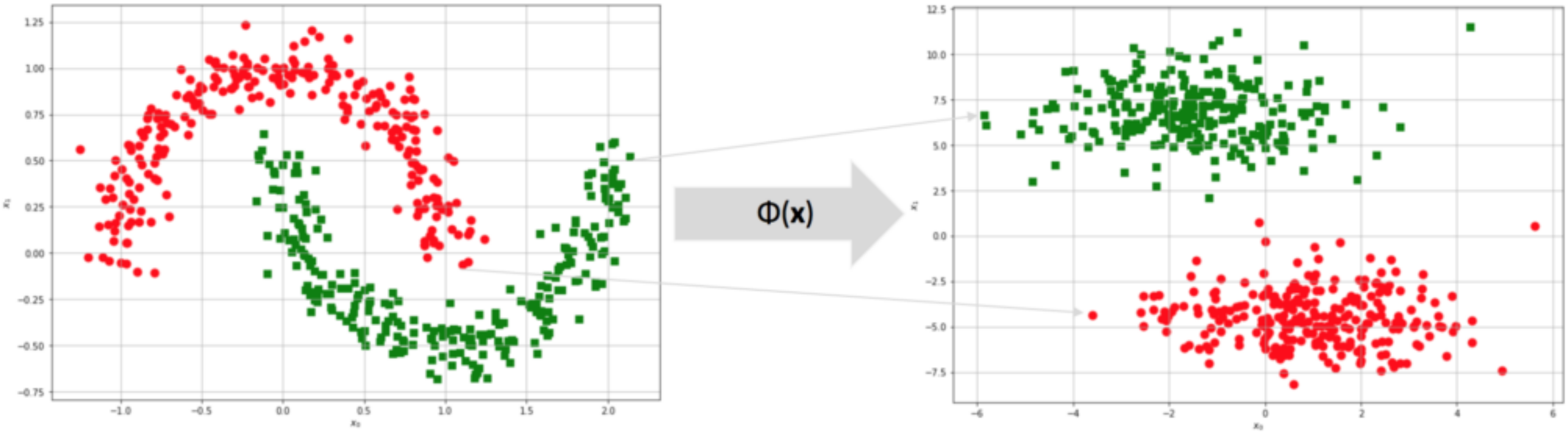


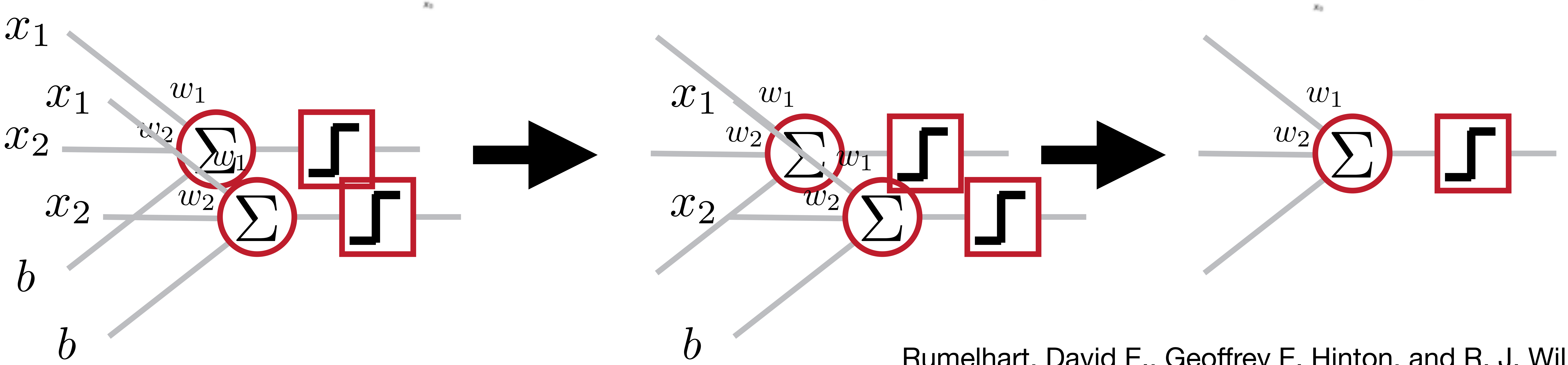
https://www.bonaccorso.eu/2017/09/28/linearly-separable-no-for-me-it-is-a-brief-introduction-to-kernel-methods/

Multi-layer Perceptron (MLP)

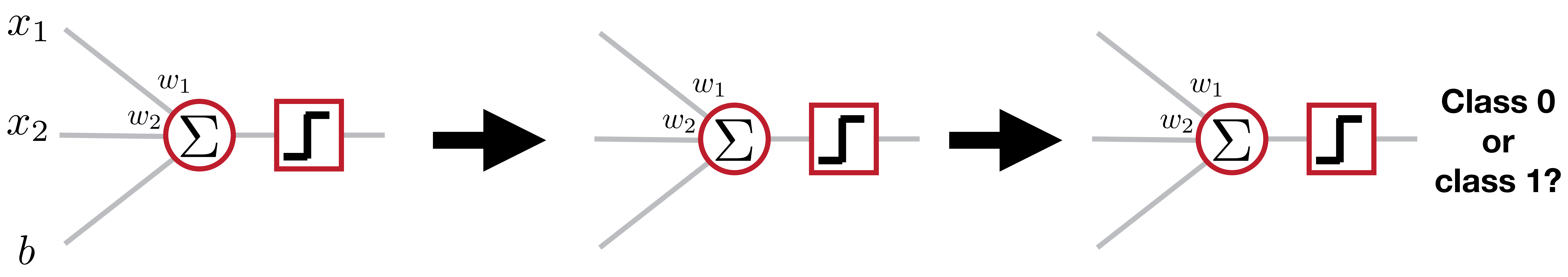


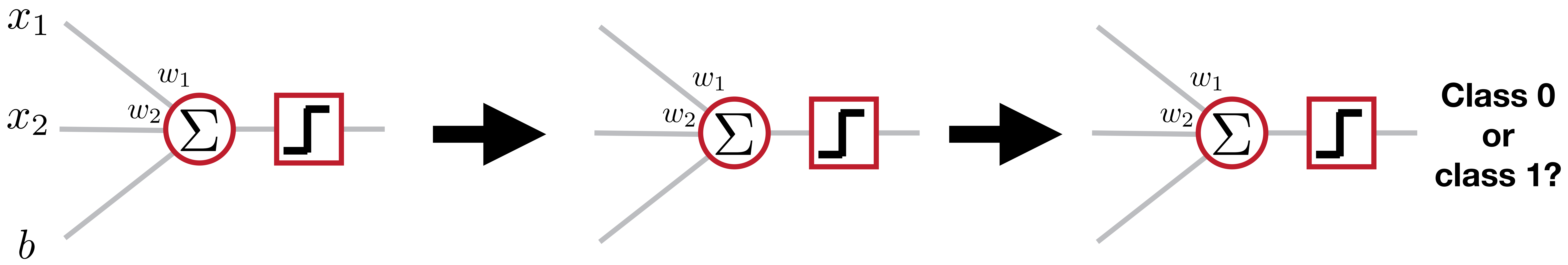
Multi-layer Perceptron (MLP)



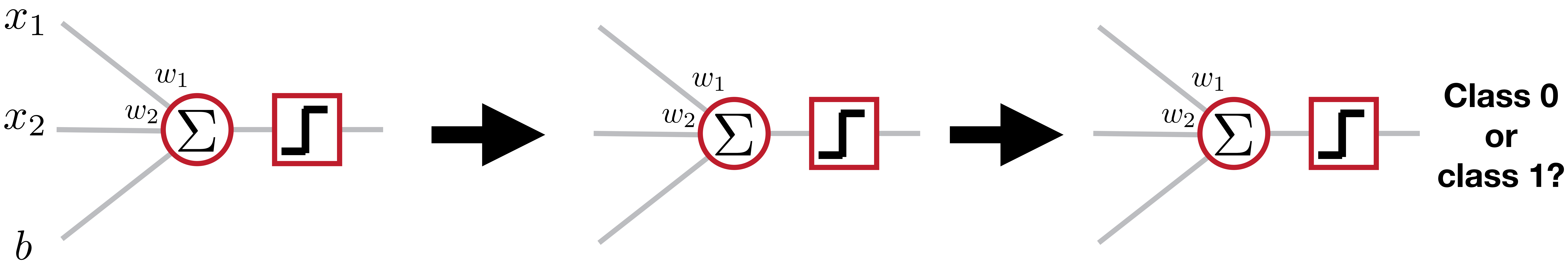


Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. (1986) "Learning Internal Representations by Error Propagation". *Parallel distributed processing*


x_1
x_2
b
w_1
w_2
Σ
Class 0
or
class 1?

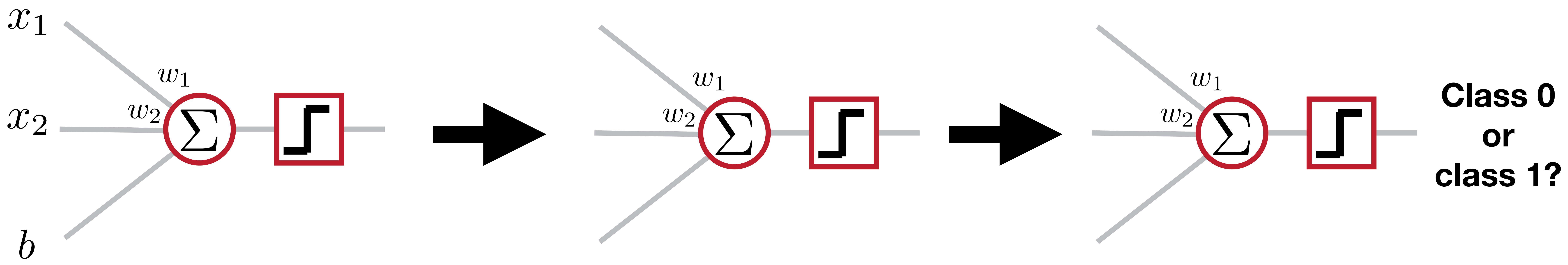


What are the parameters?



What are the parameters?

What is the objective (loss)?



What are the parameters?

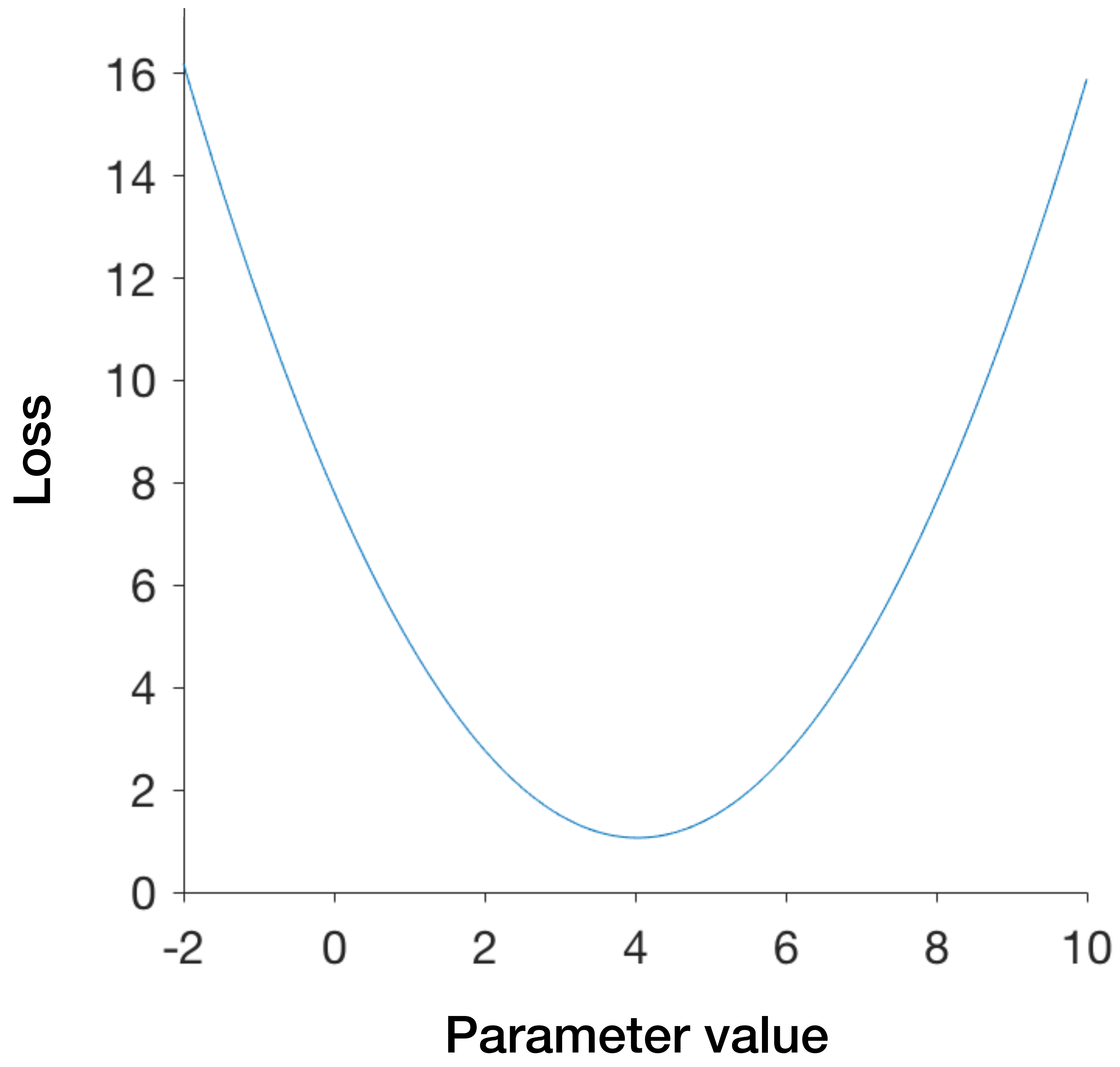
What is the objective (loss)?

How do we find the parameters?

Gradient descent for one parameter

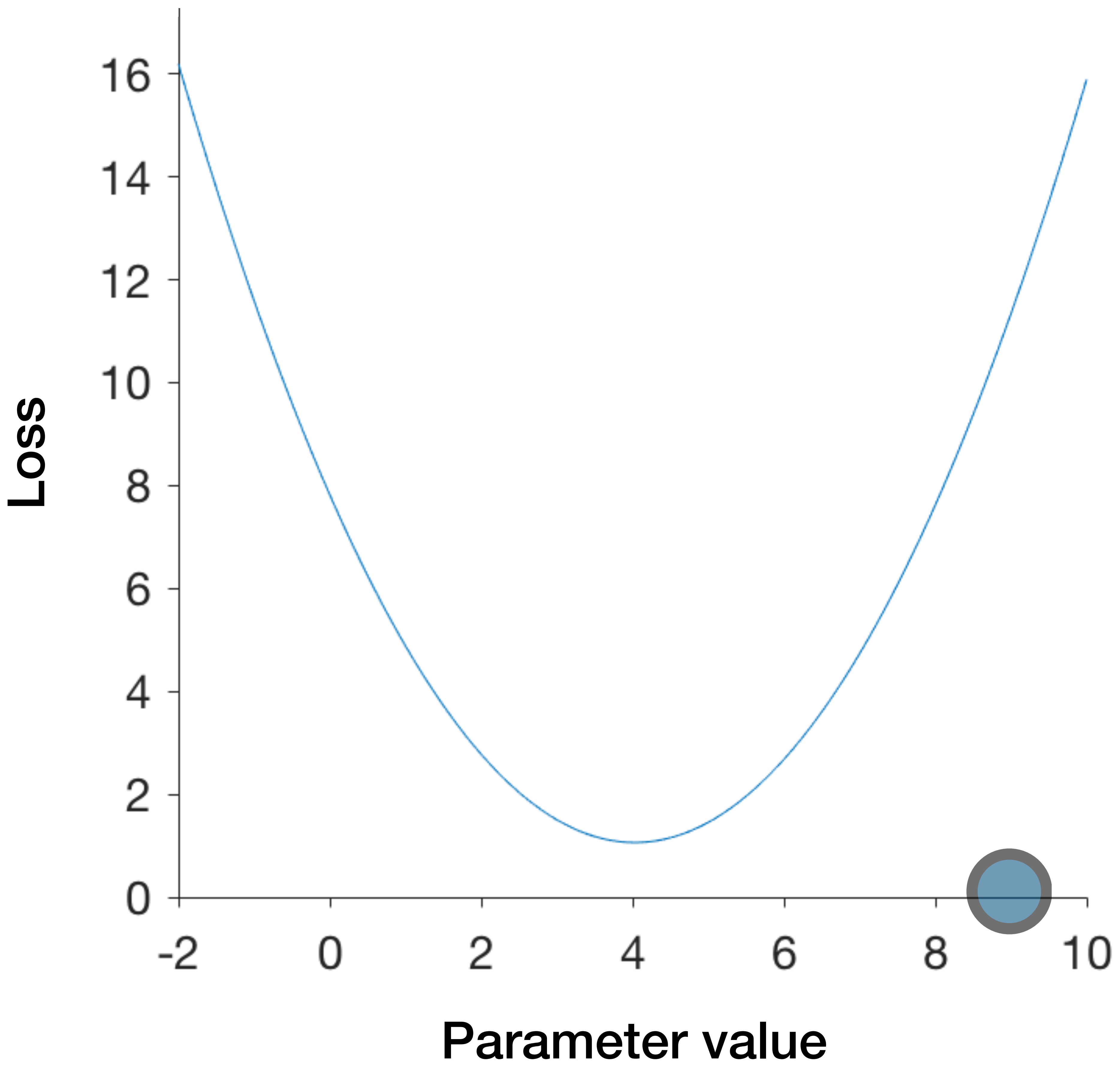
(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values



Gradient descent for one parameter
(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values

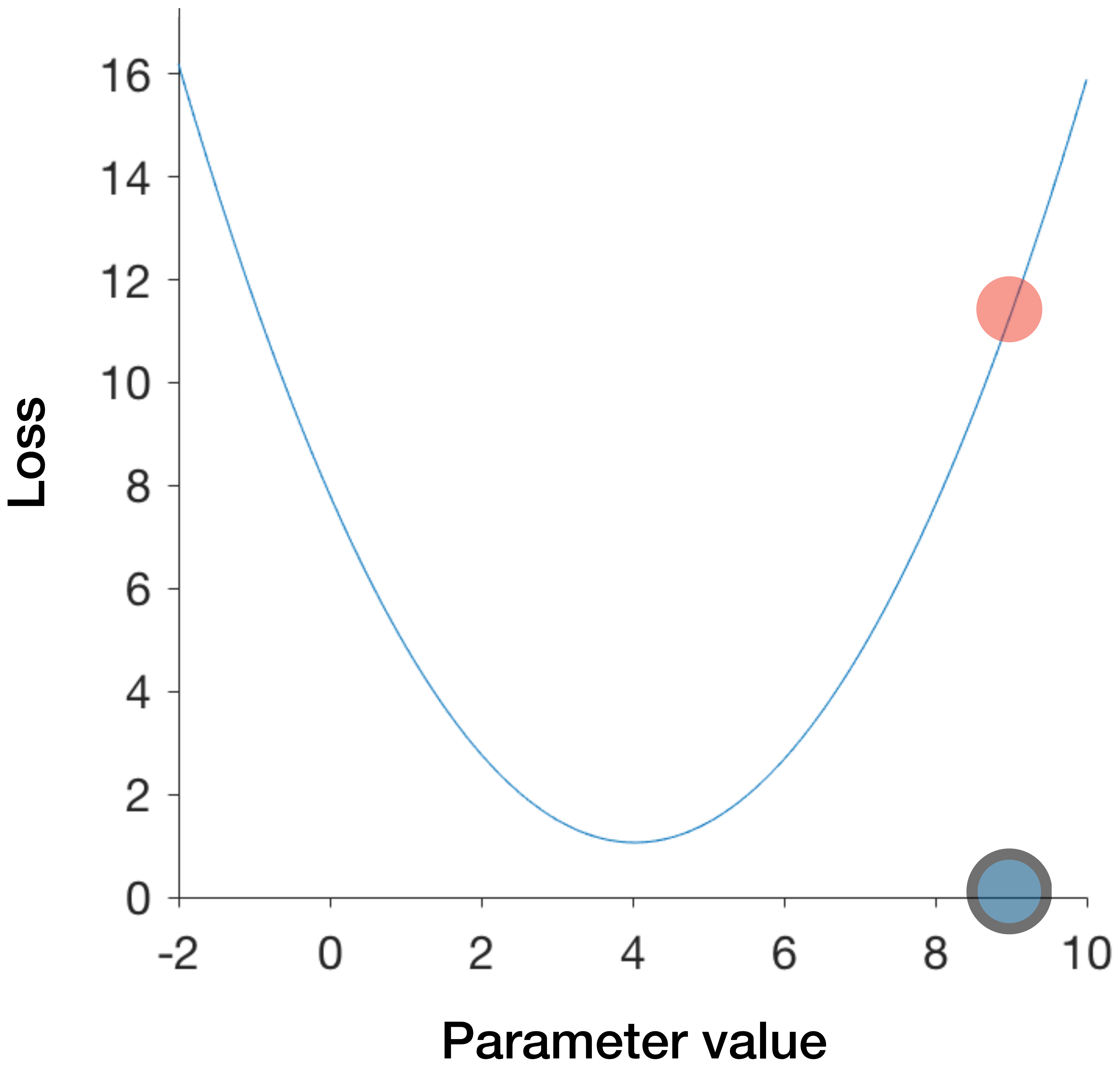


1. Guess a parameter value

Gradient descent for one parameter

(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values



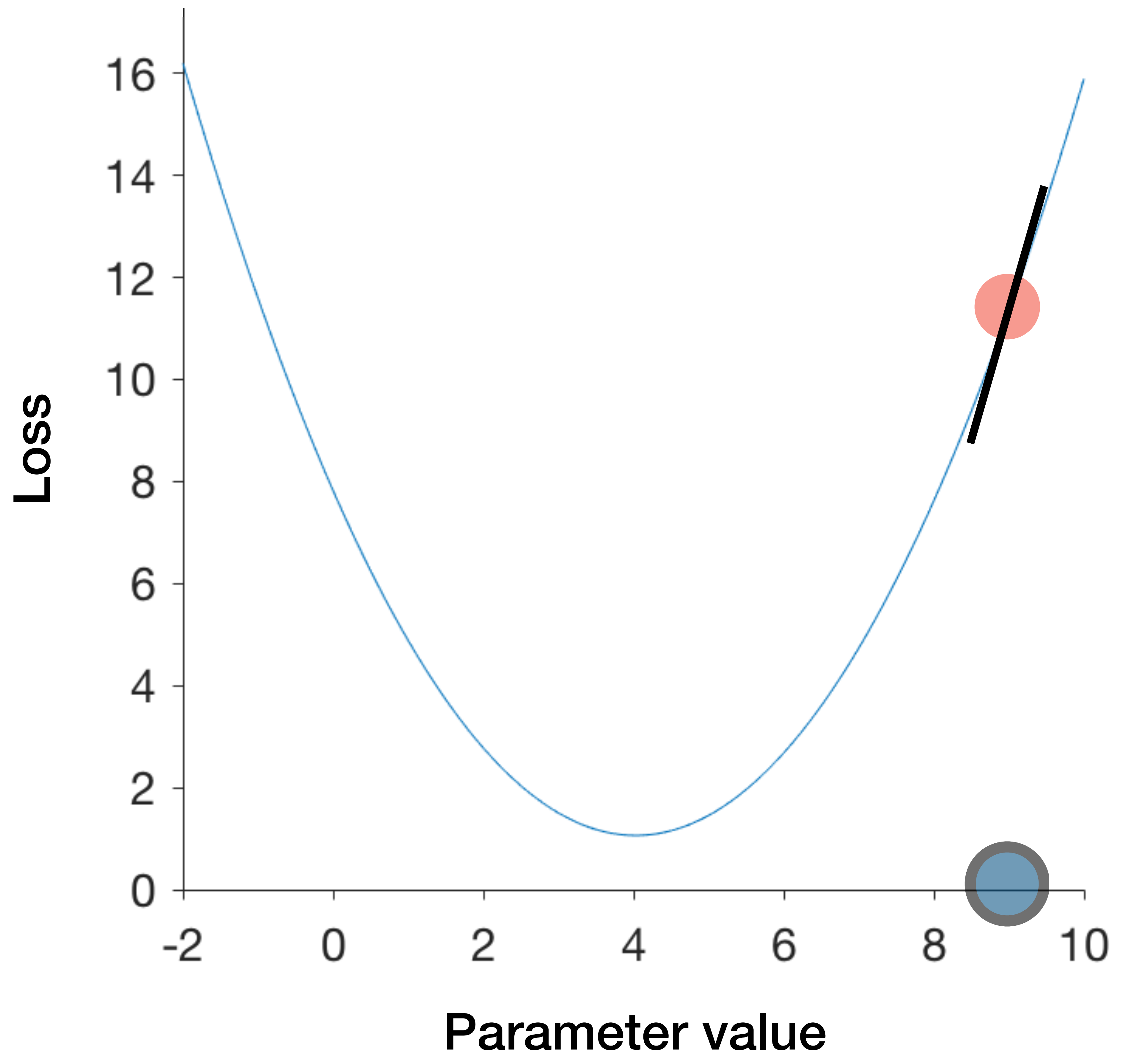
1. Guess a parameter value

2. Evaluate the loss

Gradient descent for one parameter

(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values



1. Guess a parameter value

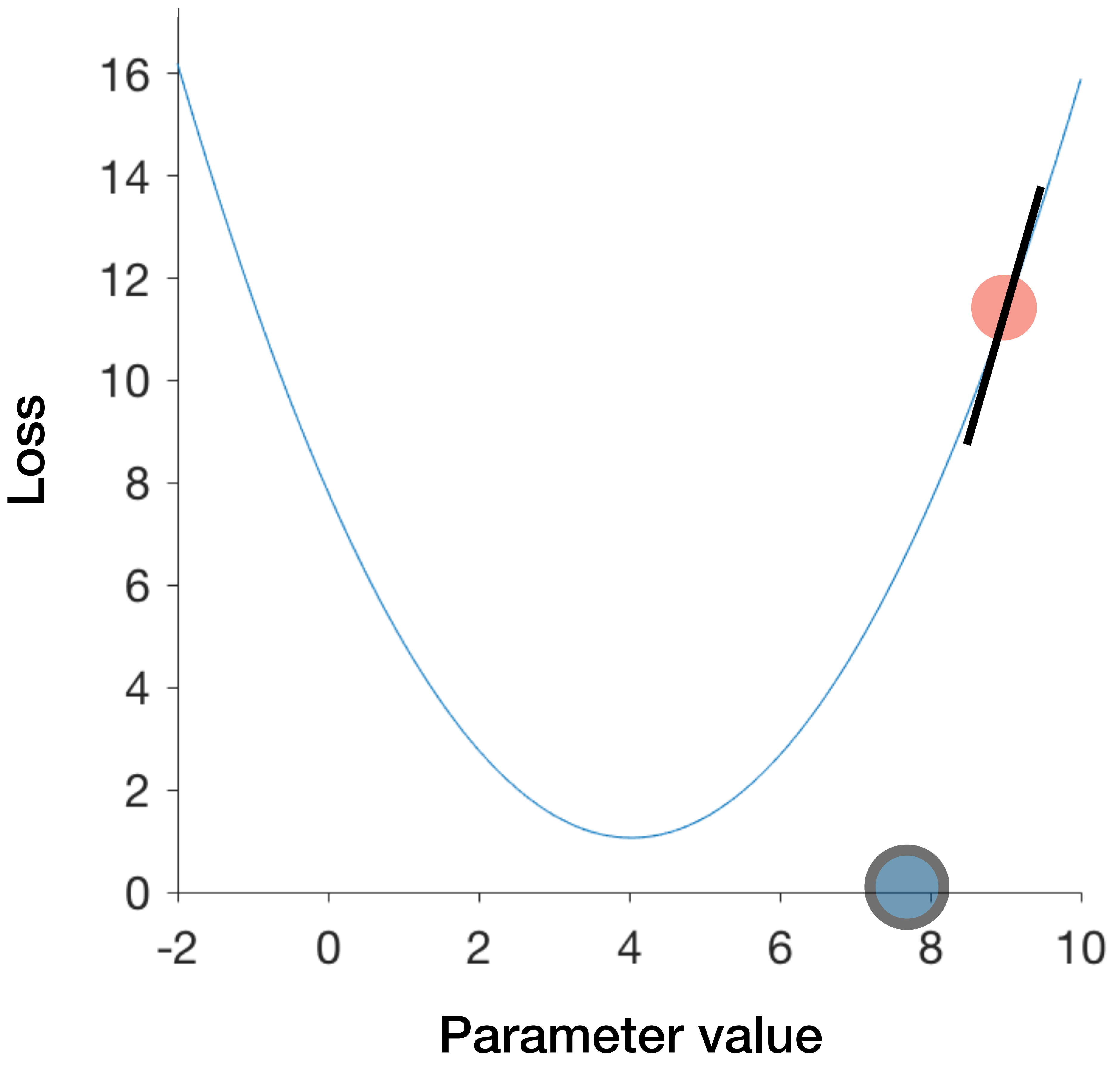
2. Evaluate the loss

3. Calculate loss derivative w.r.t. parameter (gradient)

Gradient descent for one parameter

(a.k.a., "try something, then keep trying to make it better")

Loss landscape = loss for different parameter values

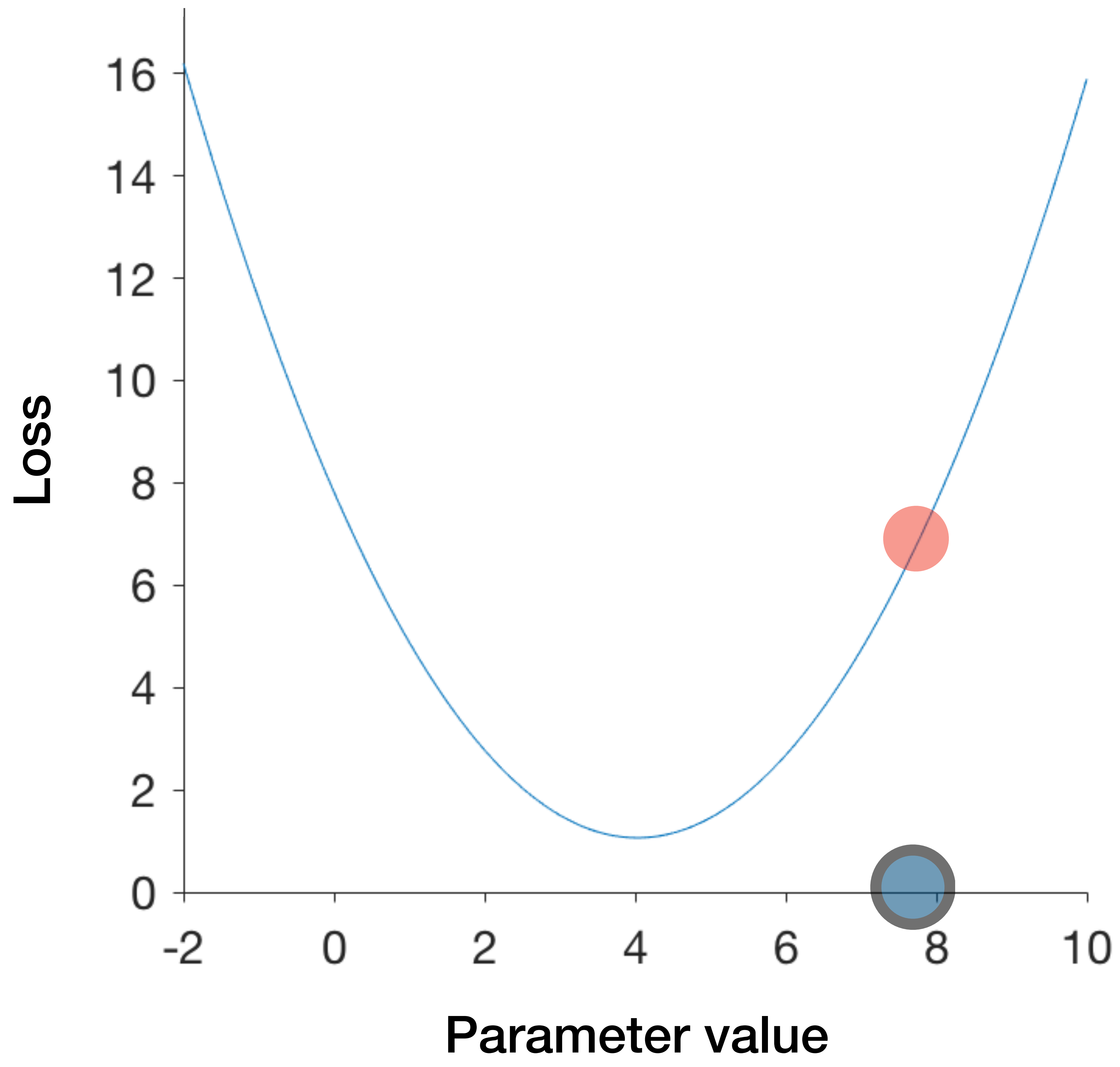


1. Guess a parameter value
2. Evaluate the loss
3. Calculate loss derivative w.r.t. parameter (gradient)
4. Step down the gradient

Gradient descent for one parameter

(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values



1. Guess a parameter value

2. Evaluate the loss

3. Calculate loss derivative w.r.t. parameter (gradient)

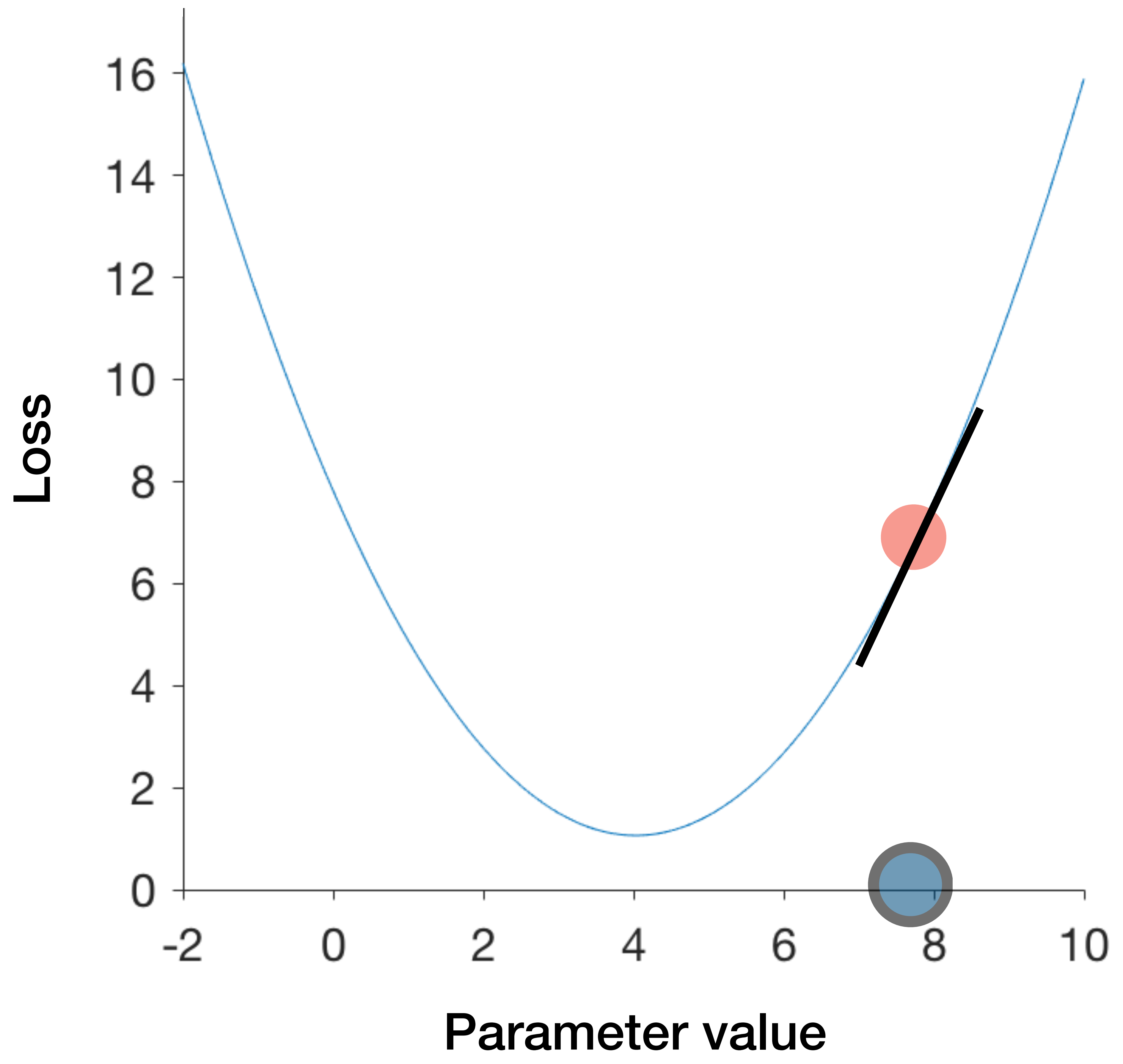
4. Step down the gradient

5. Repeat!

Gradient descent for one parameter

(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values



1. Guess a parameter value

2. Evaluate the loss

3. Calculate loss derivative w.r.t. parameter (gradient)

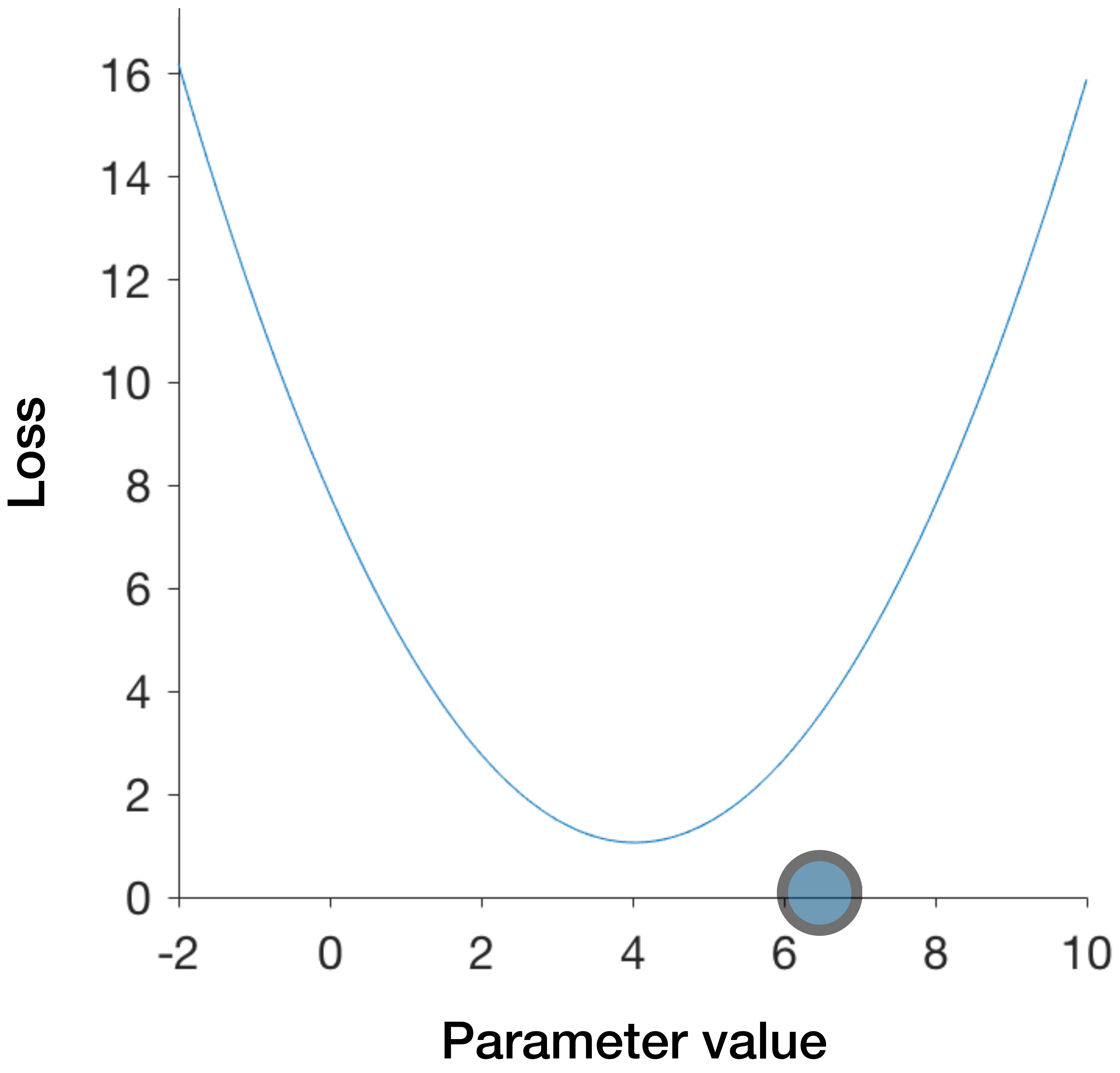
4. Step down the gradient

5. Repeat!

Gradient descent for one parameter

(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values

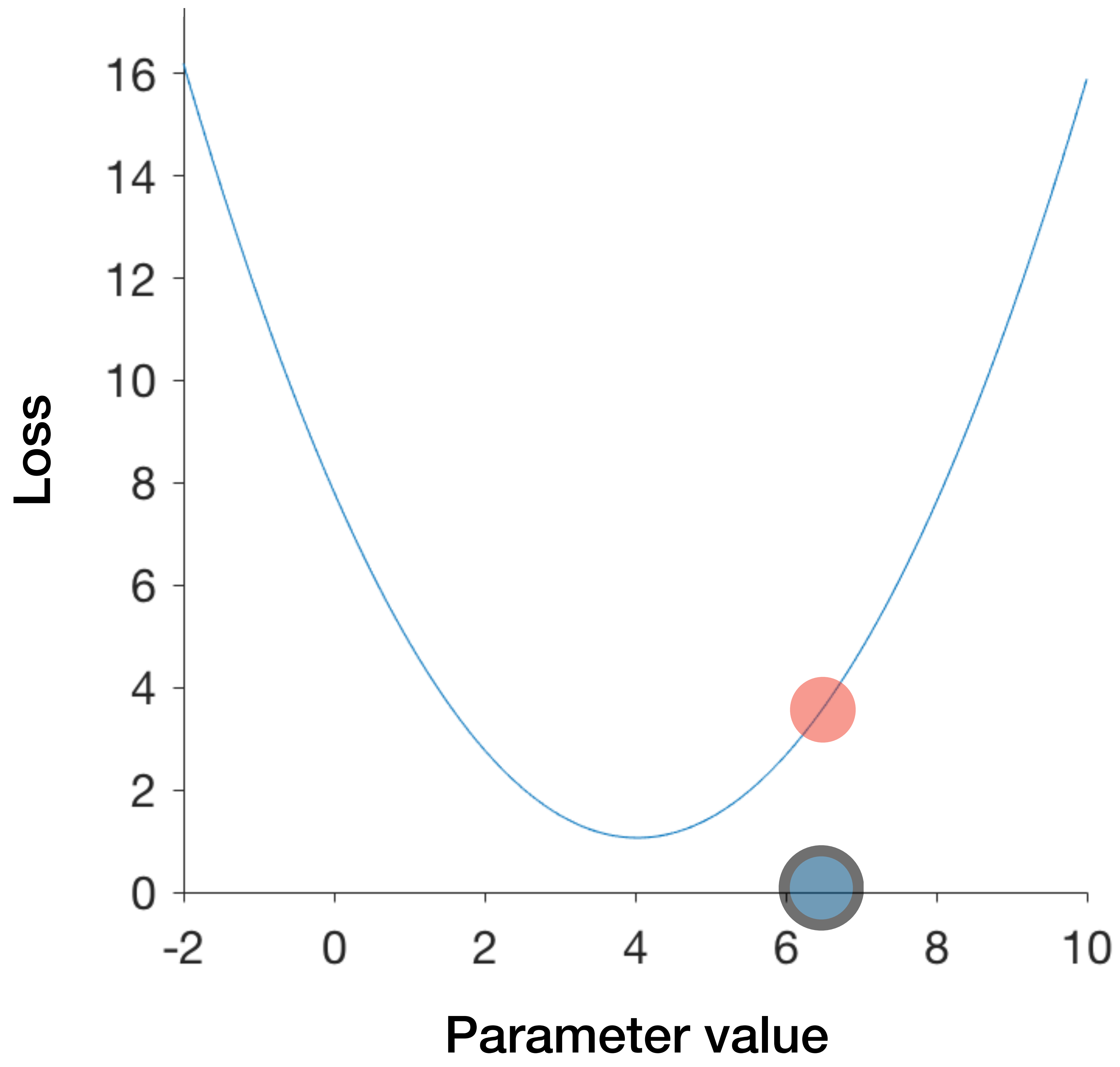


1. Guess a parameter value
2. Evaluate the loss
3. Calculate loss derivative w.r.t. parameter (gradient)
4. Step down the gradient
5. Repeat!

Gradient descent for one parameter

(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values

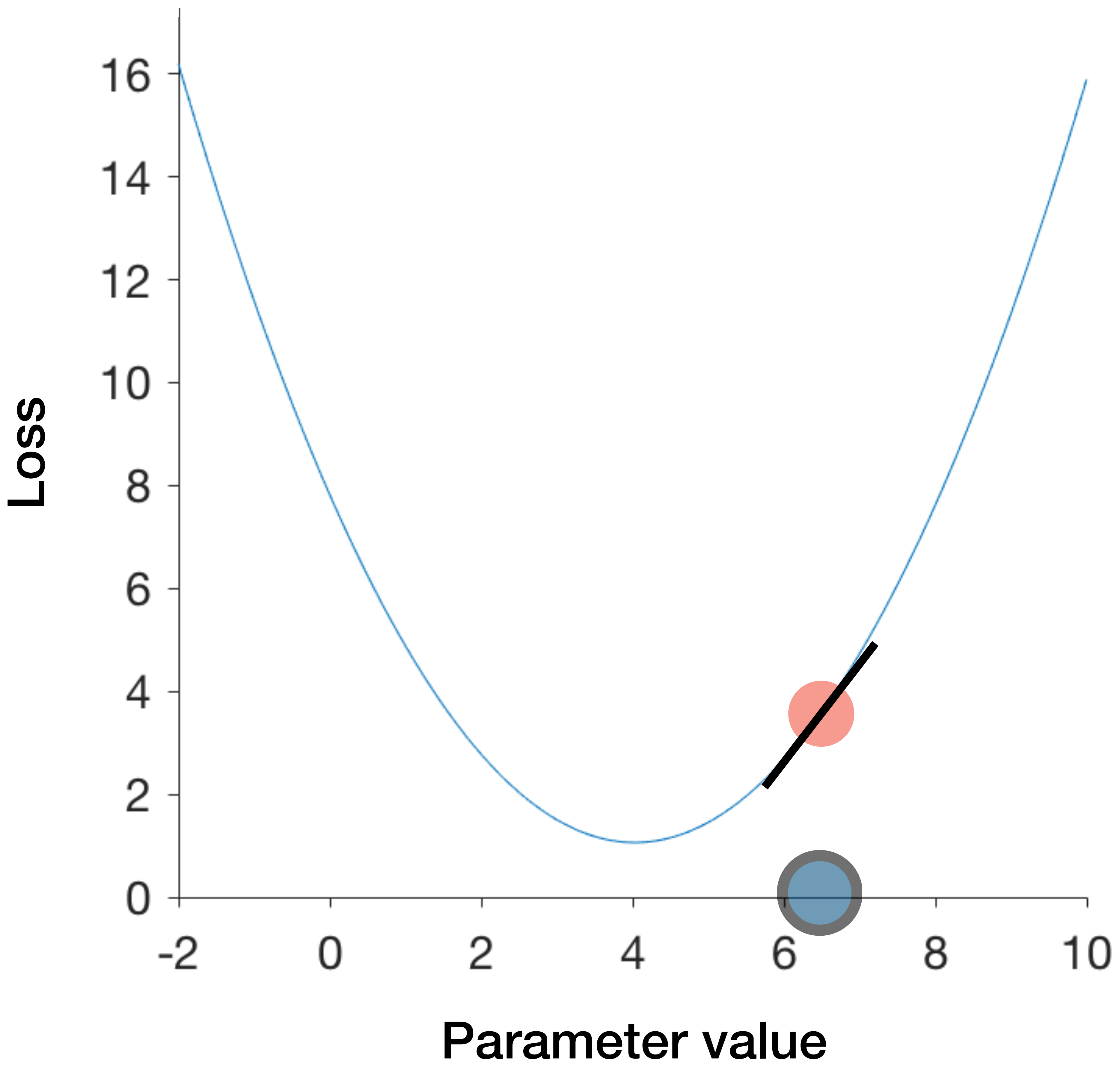


1. Guess a parameter value
2. Evaluate the loss
3. Calculate loss derivative w.r.t. parameter (gradient)
4. Step down the gradient
5. Repeat!

Gradient descent for one parameter

(a.k.a., “try something, then keep trying to make it better”)

Loss landscape = loss for different parameter values



1. Guess a parameter value

2. Evaluate the loss

3. Calculate loss derivative w.r.t. parameter (gradient)

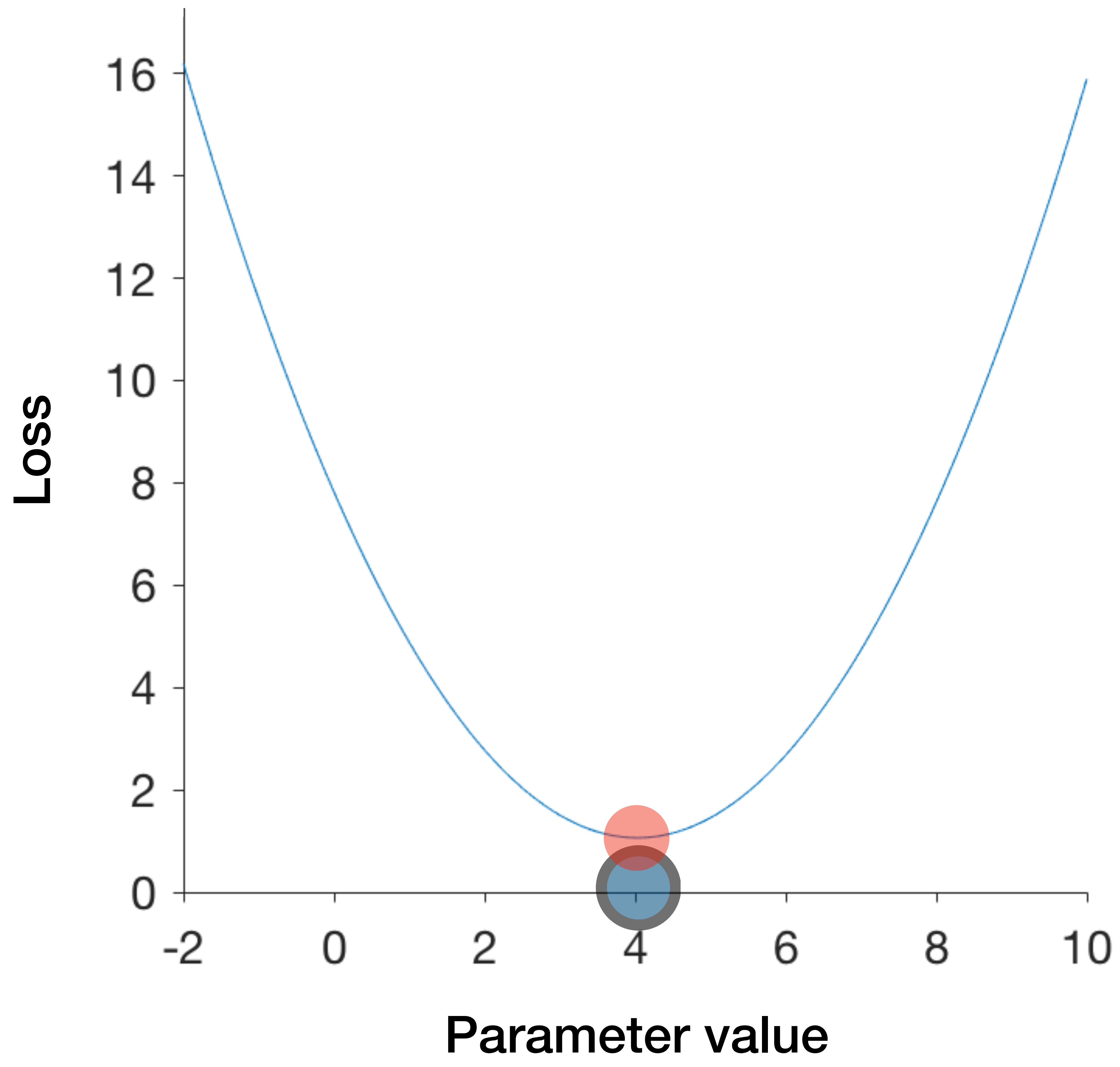
4. Step down the gradient

5. Repeat!

Gradient descent for one parameter

(a.k.a., “try something, then keep trying to make it better”)

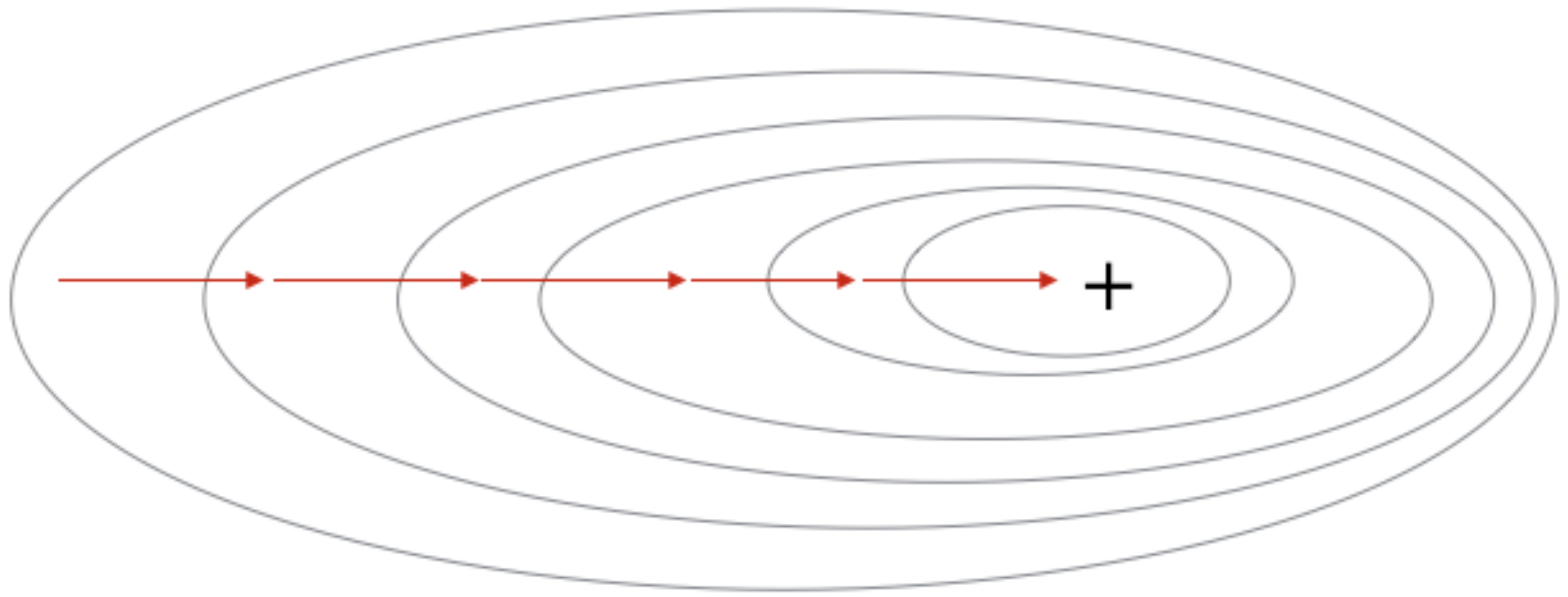
Loss landscape = loss for different parameter values



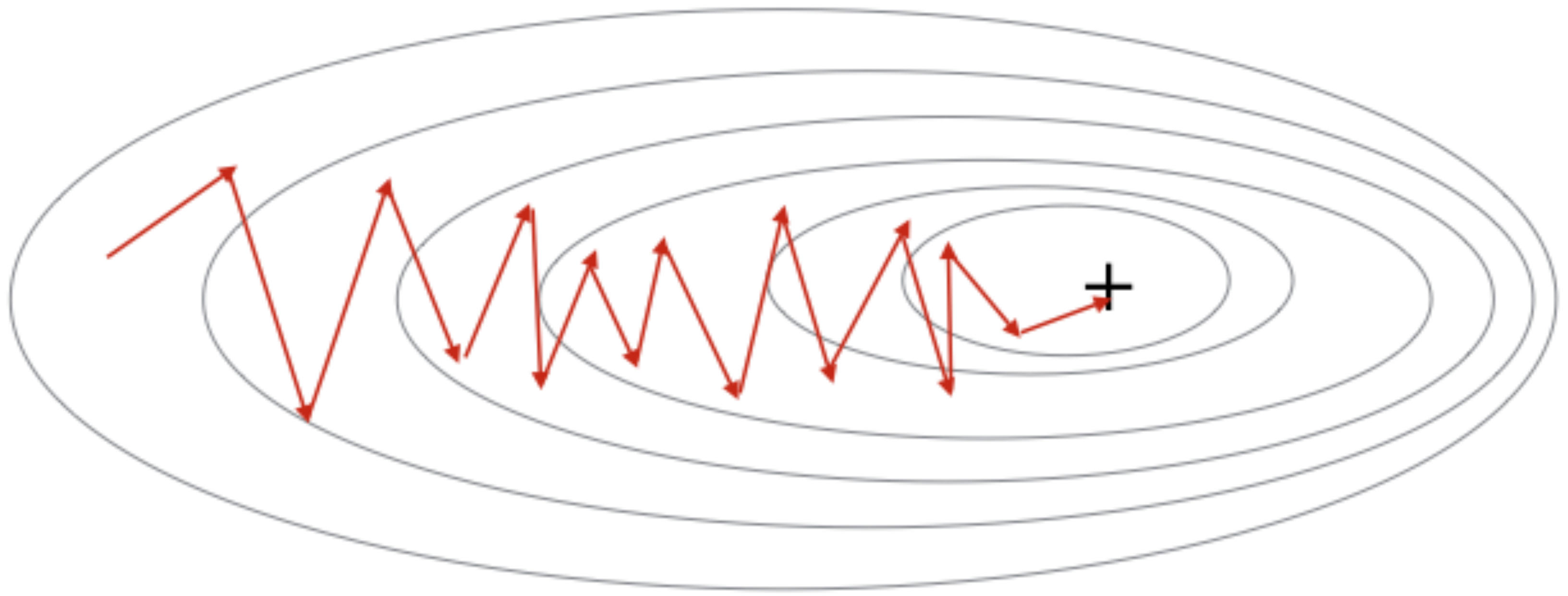
1. Guess a parameter value
2. Evaluate the loss
3. Calculate loss derivative w.r.t. parameter (gradient)
4. Step down the gradient
5. Repeat!

Gradient Descent and Stochastic Gradient Descent

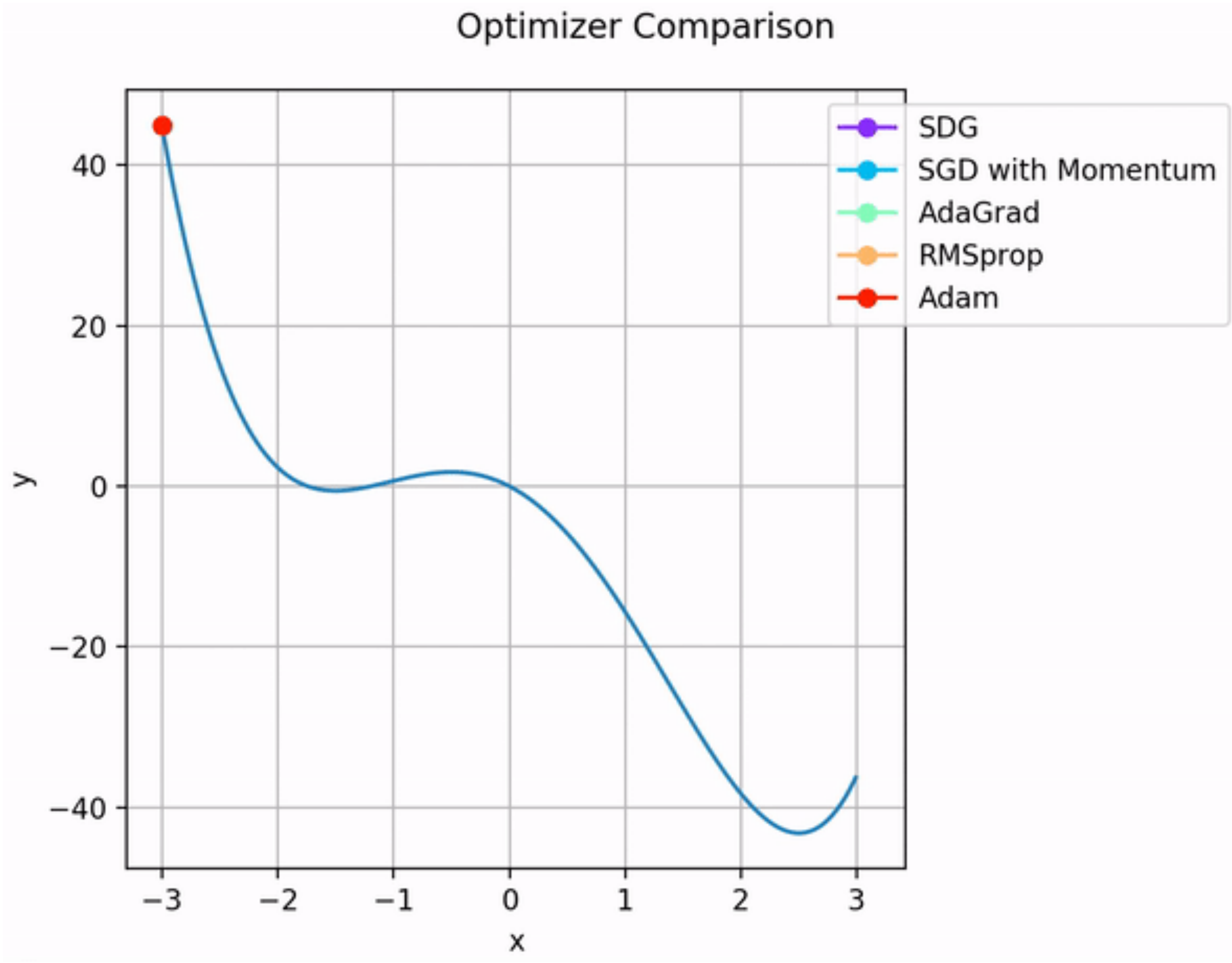
Gradient Descent



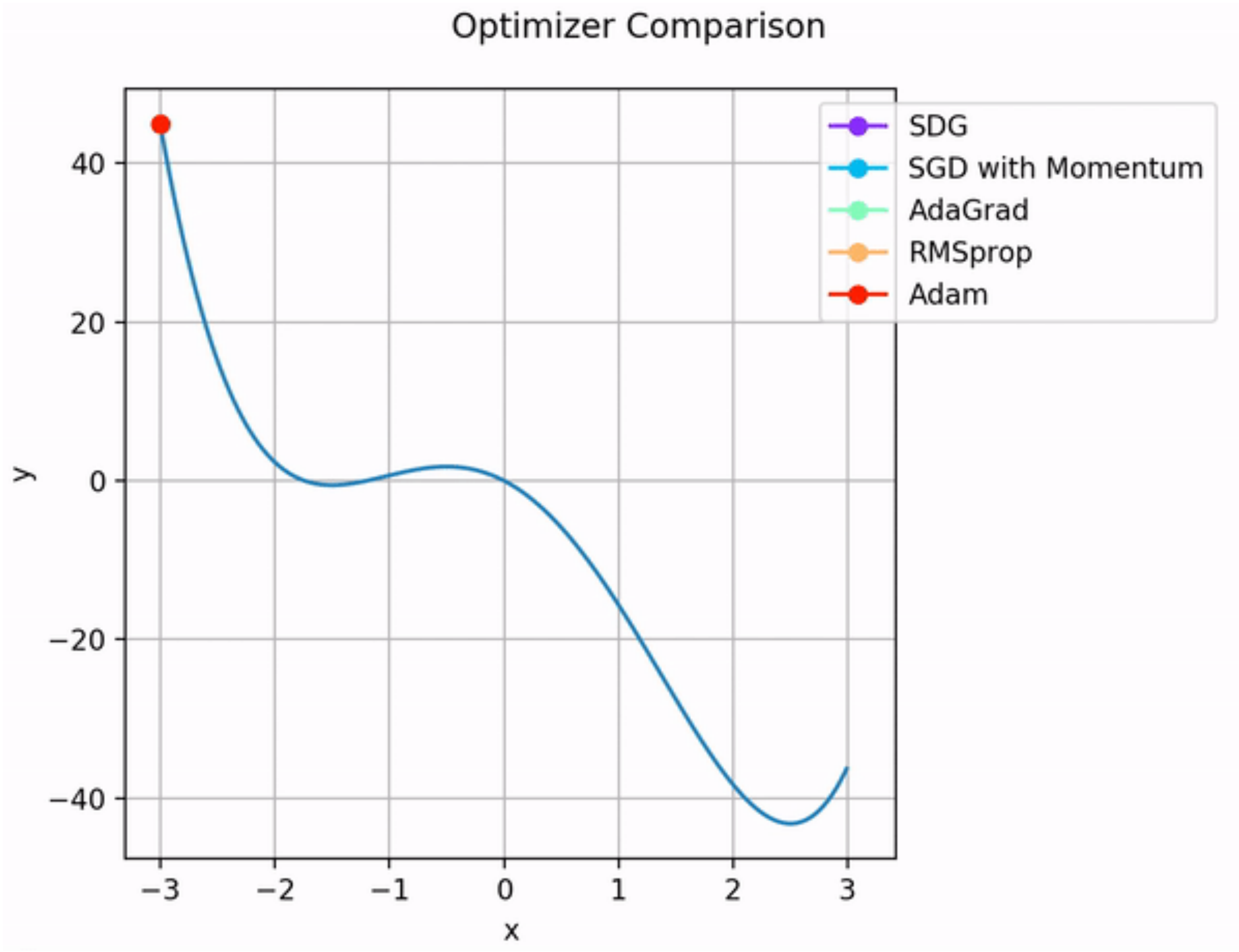
Stochastic Gradient Descent



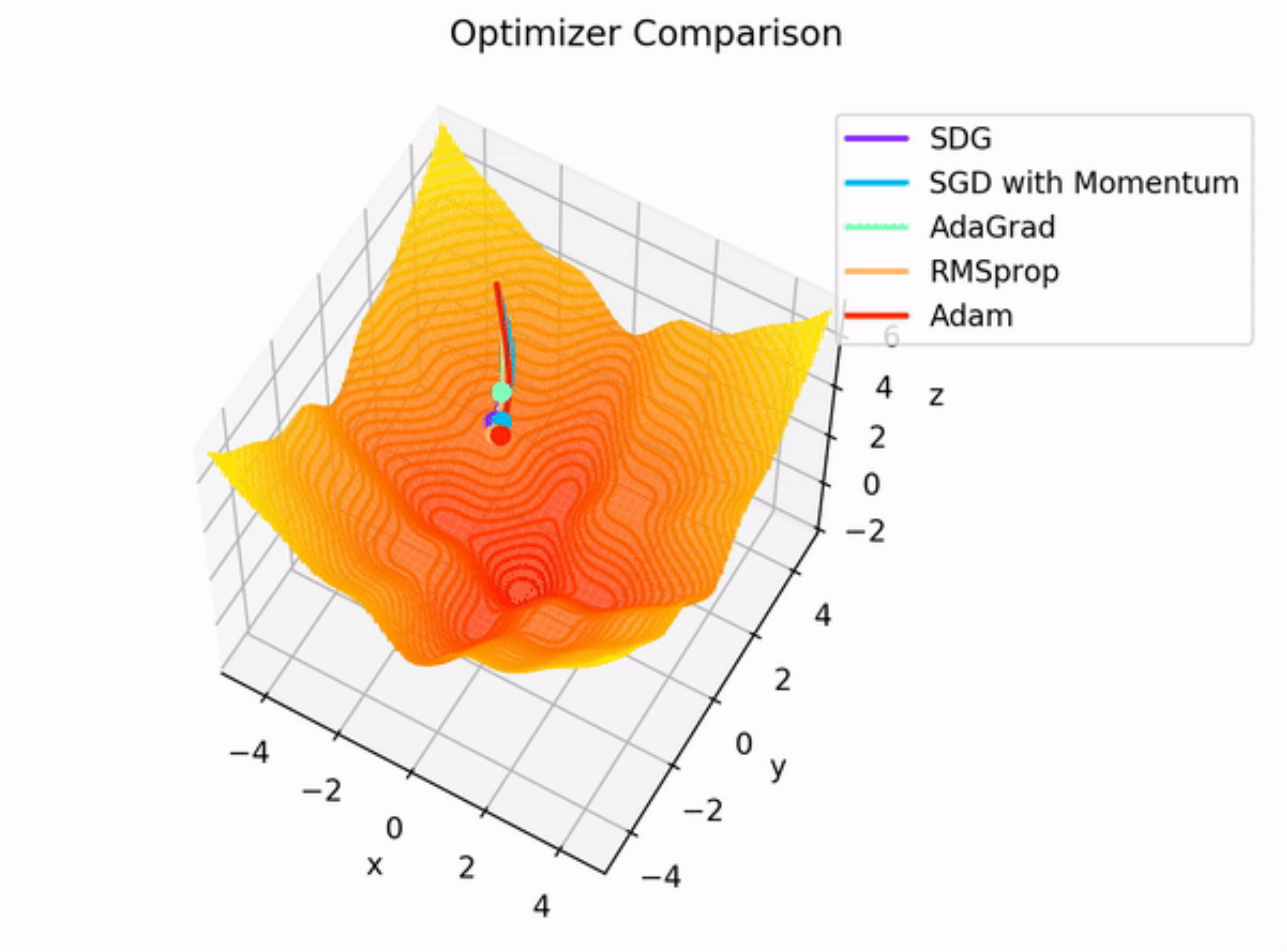
https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d



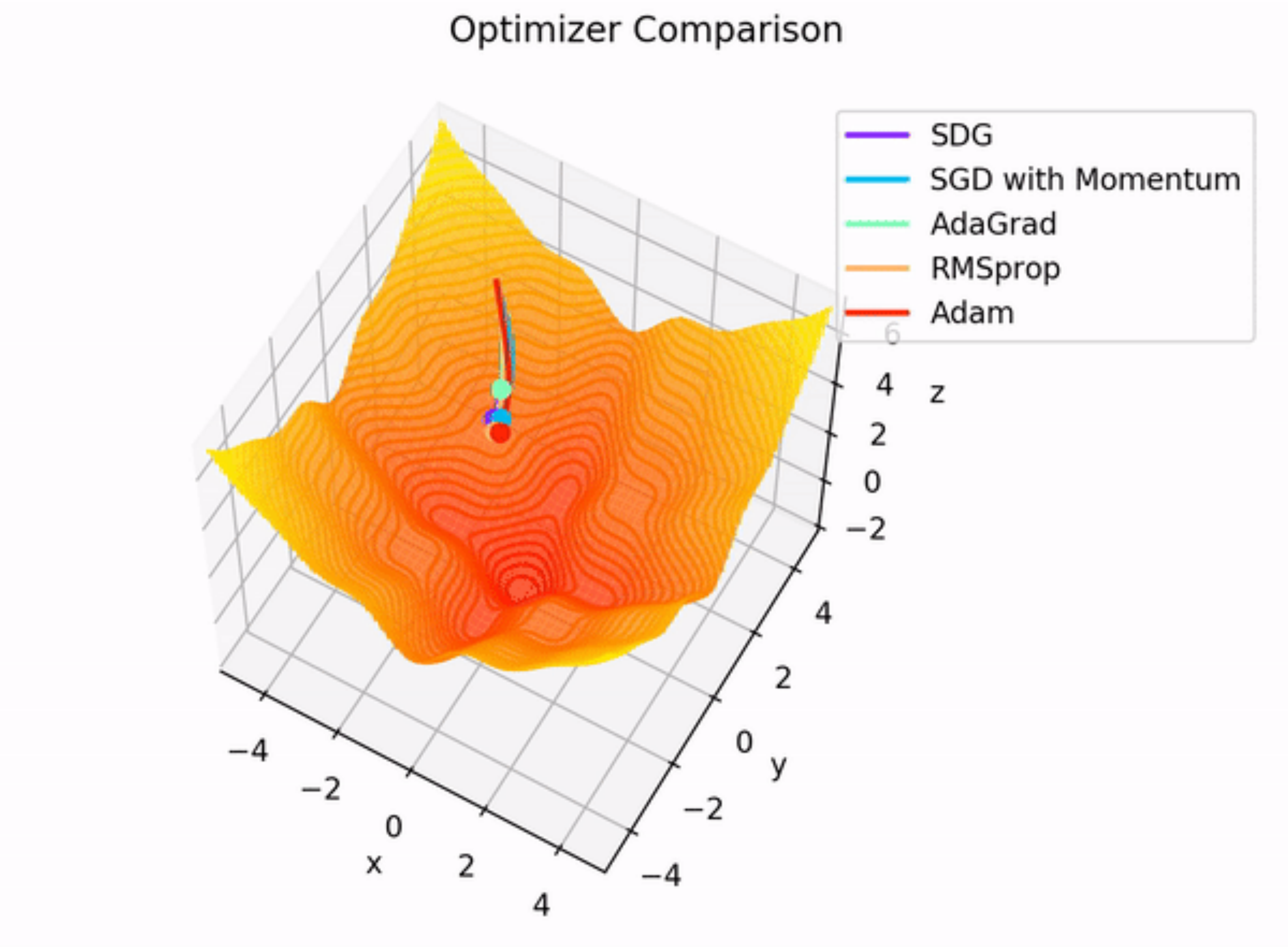
https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d



https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d

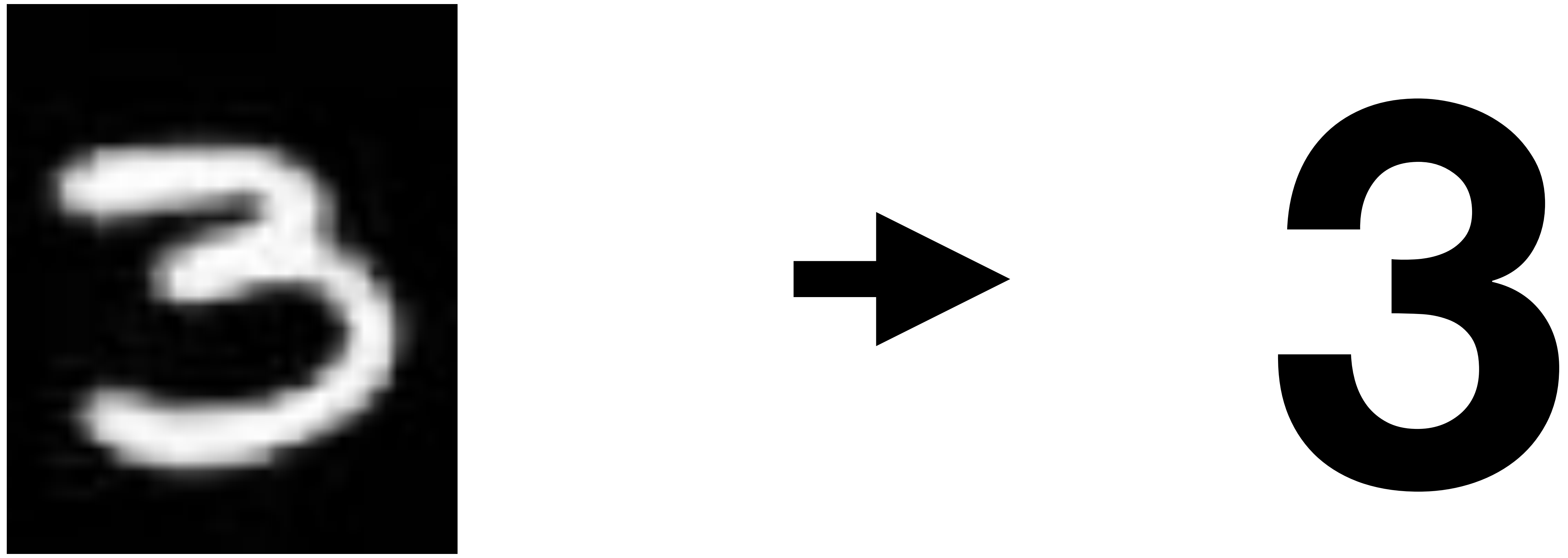


https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d



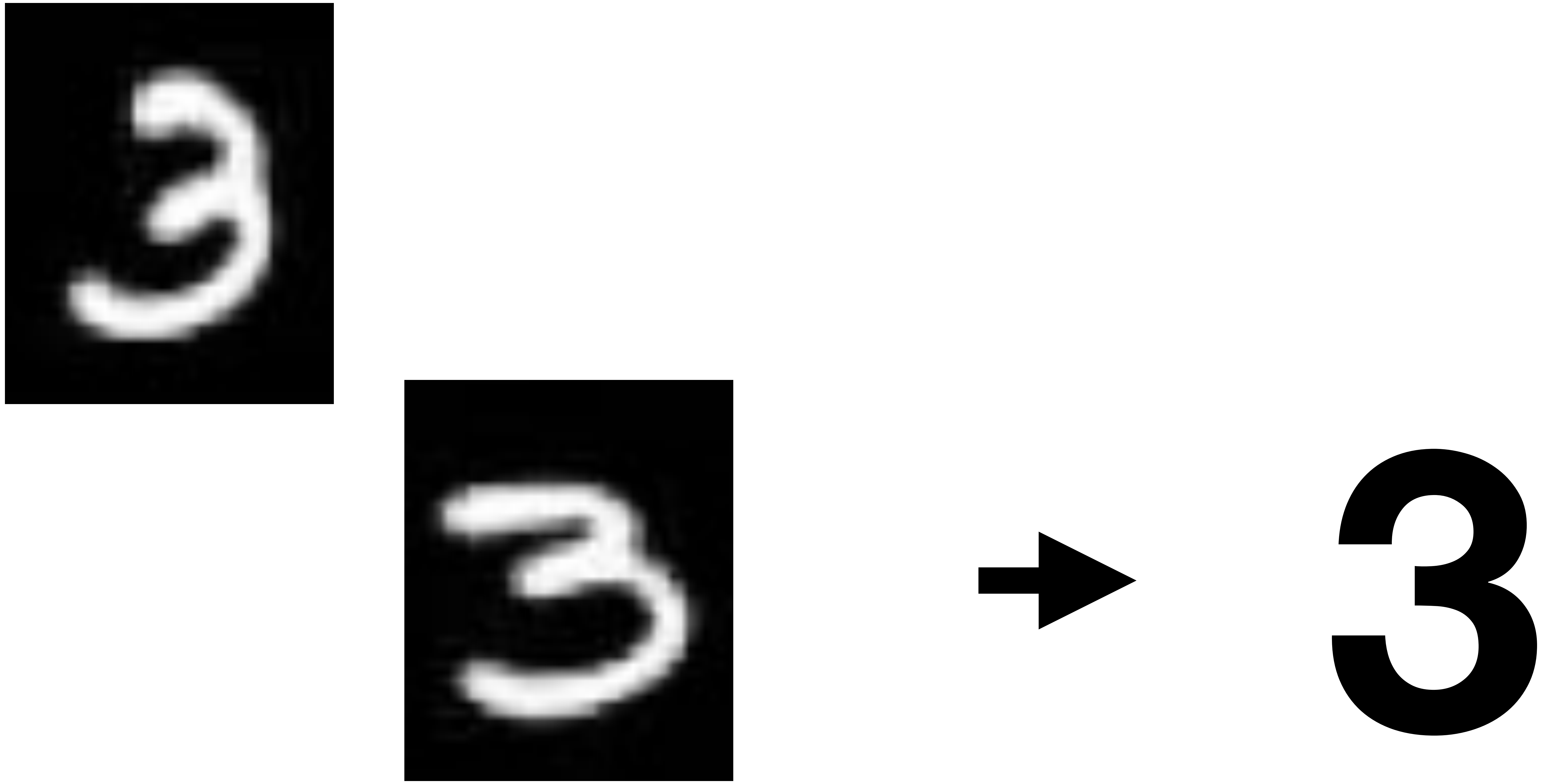
https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d

High-D problems



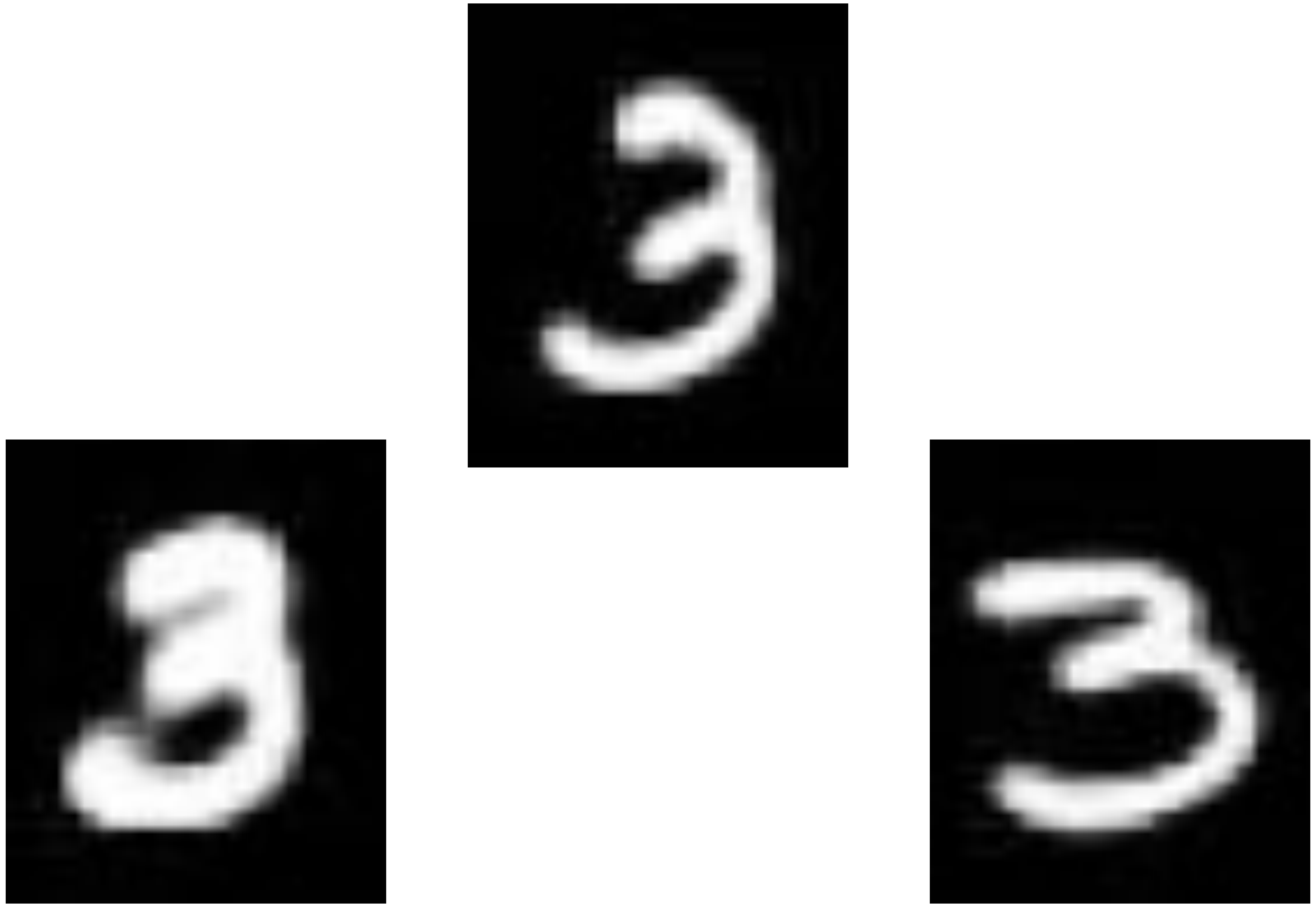
MNIST digits-
http://yann.lecun.com/exdb/mnist/

High-D problems



MNIST digits-
http://yann.lecun.com/exdb/mnist/

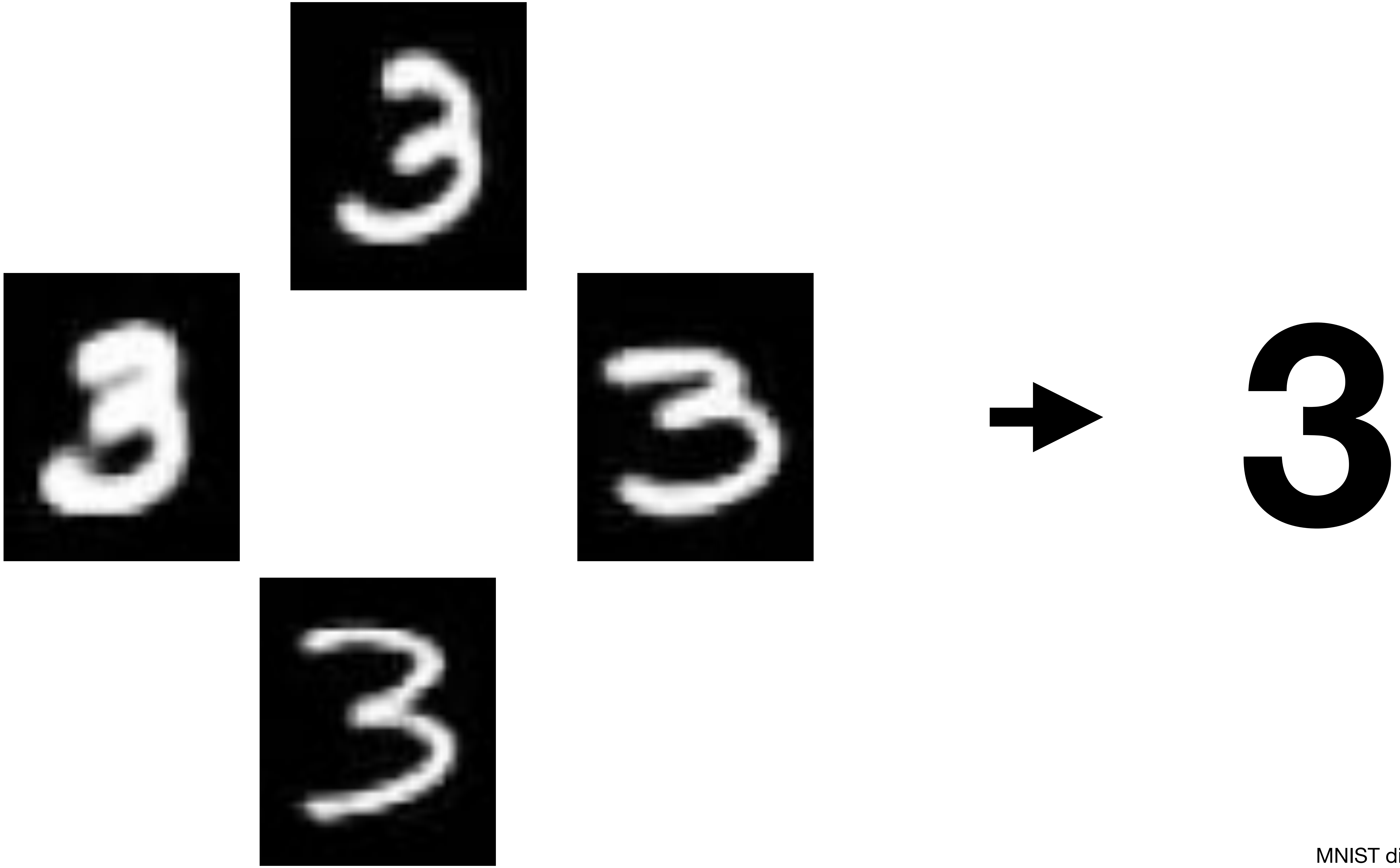
High-D problems



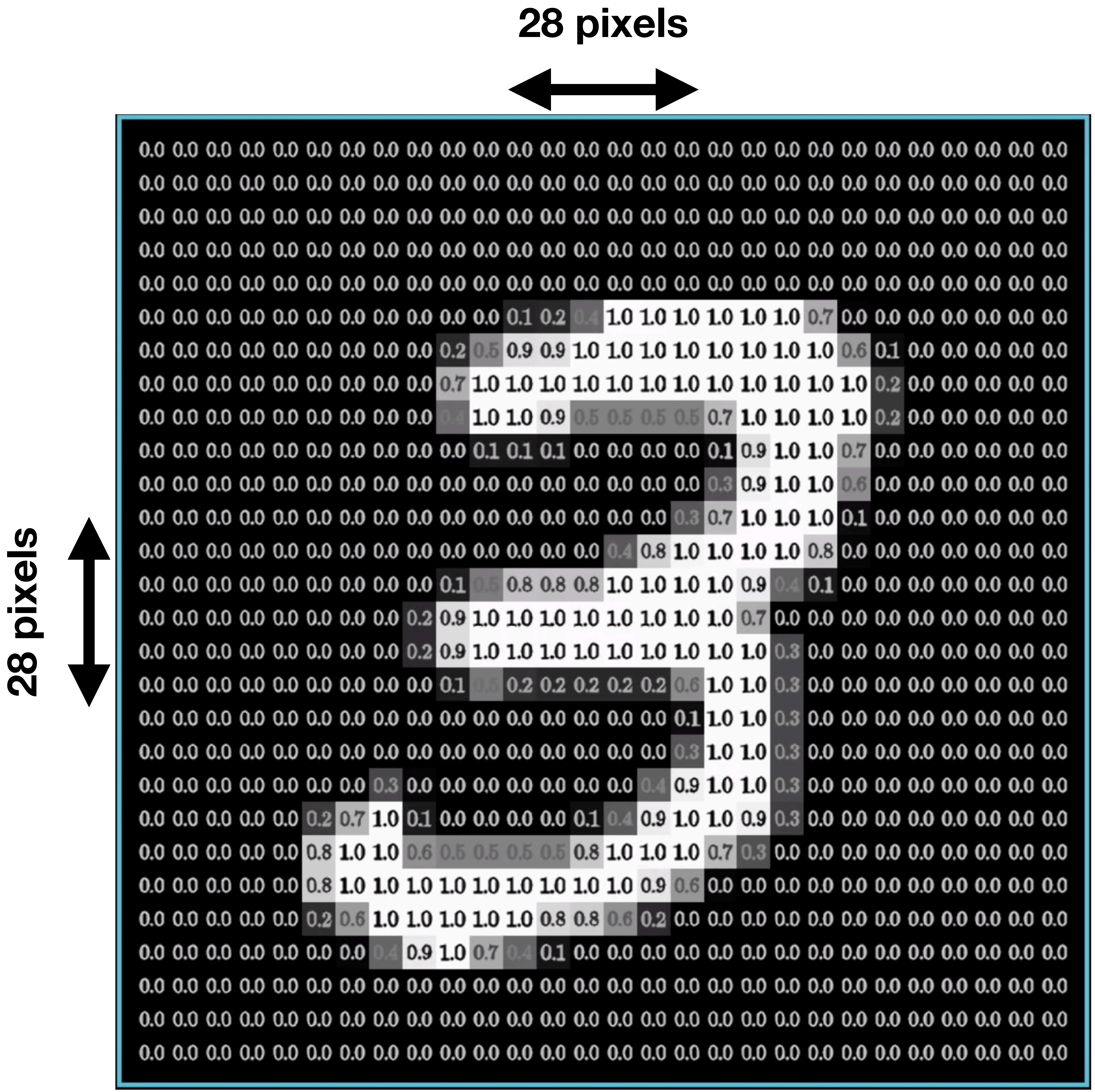
3

MNIST digits-
http://yann.lecun.com/exdb/mnist/

High-D problems

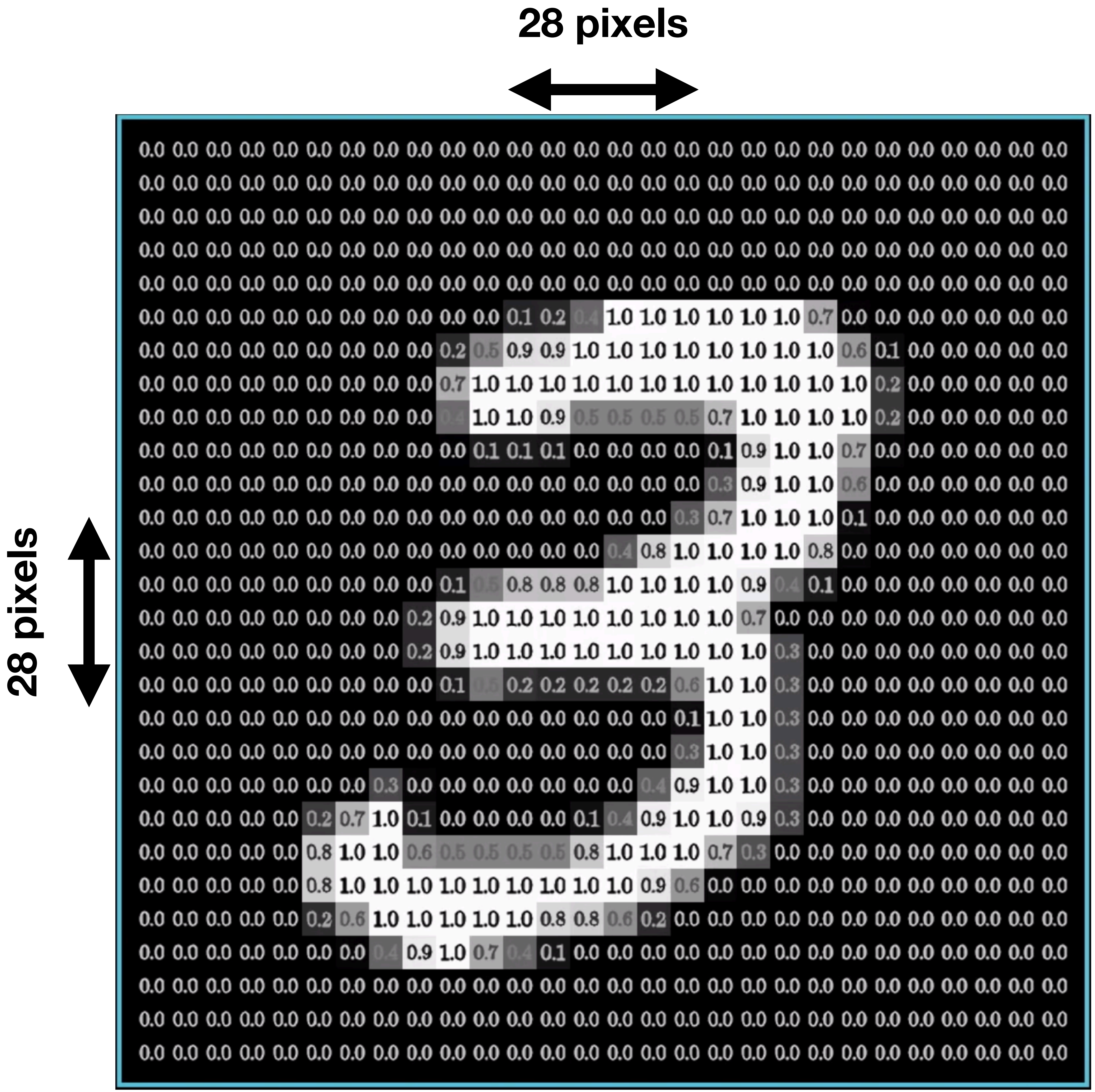


MNIST digits-
http://yann.lecun.com/exdb/mnist/



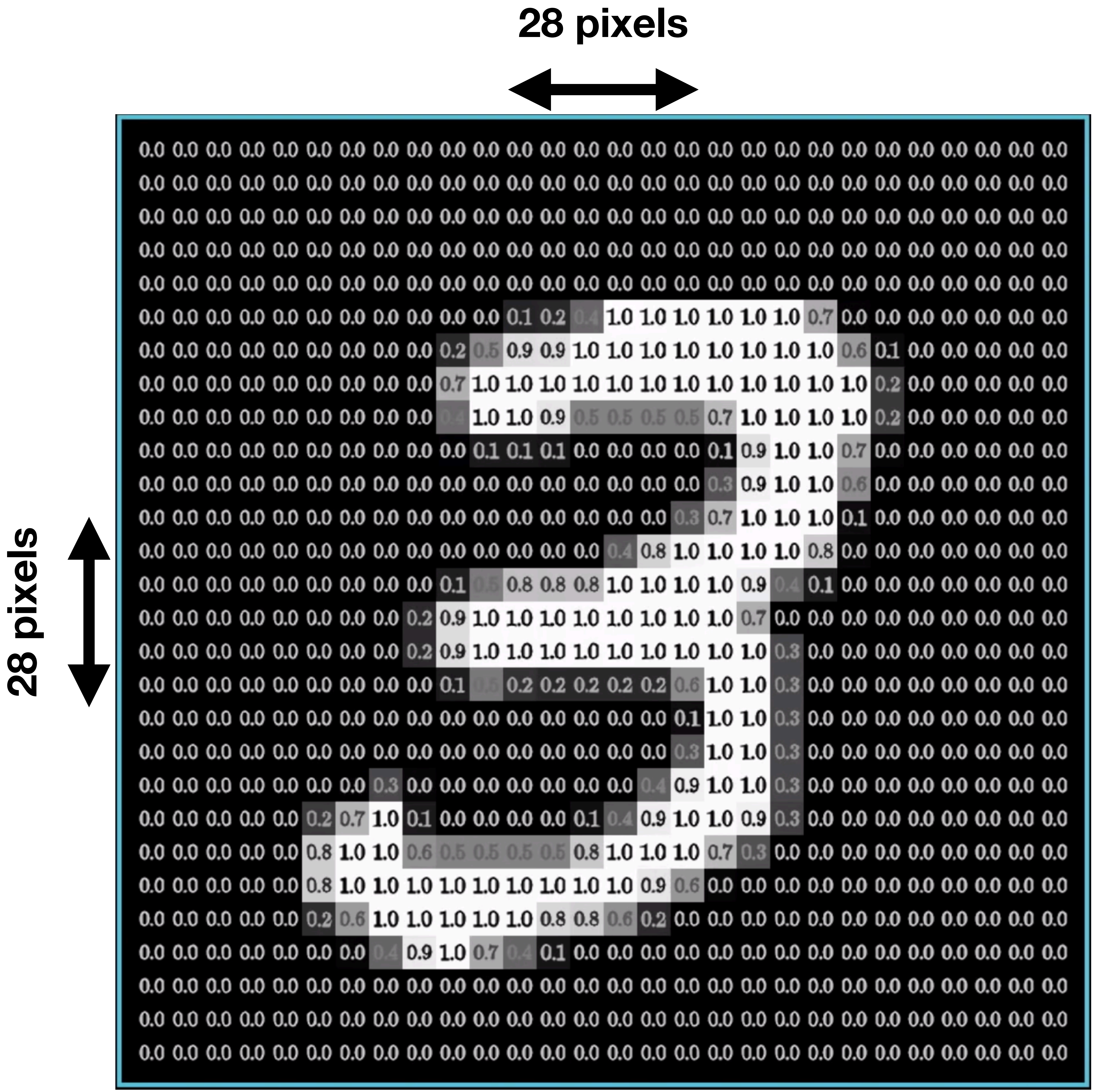
MNIST - 784 total pixels
each ranges from 0 to 1

3Blue1Brown (YouTube)
But what is a Neural Network? | Deep learning, chapter 1

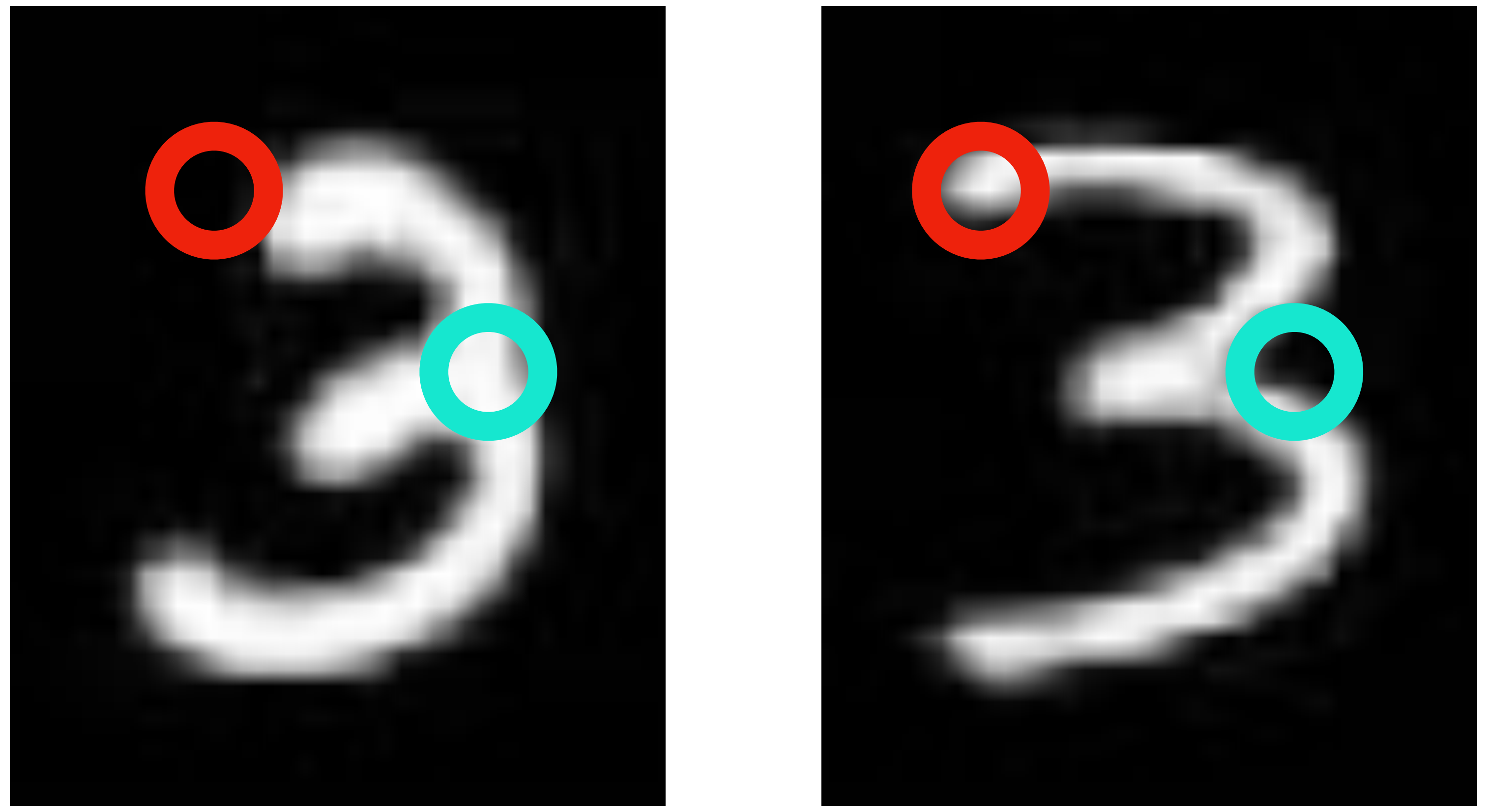


MNIST - 784 total pixels each ranges from 0 to 1

3Blue1Brown (YouTube)
But what is a Neural Network? | Deep learning, chapter 1



MNIST - 784 total pixels each ranges from 0 to 1

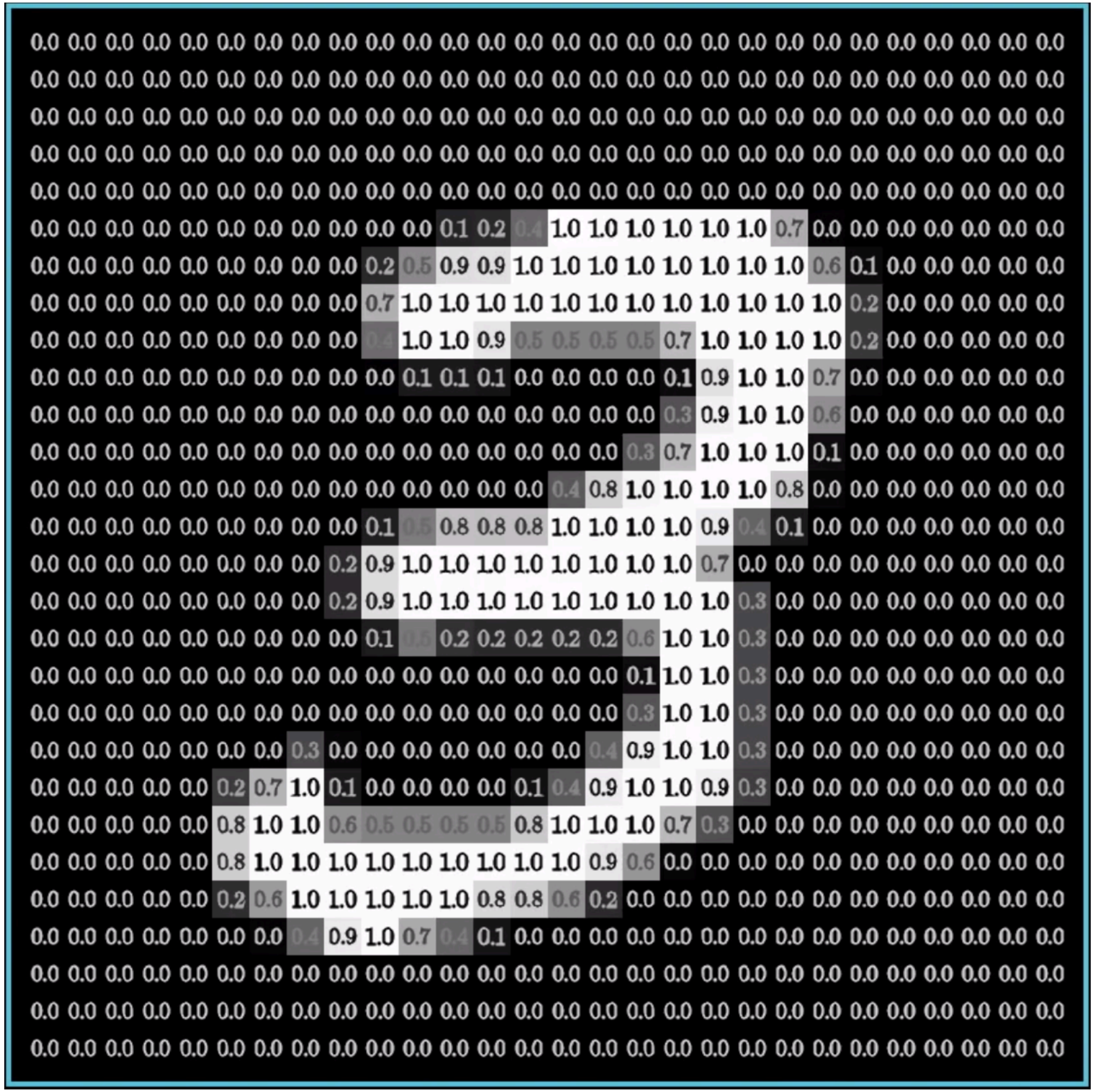


Specific pixel values are very different between images

3Blue1Brown (YouTube)
But what is a Neural Network? | Deep learning, chapter 1

28 pixels

28 pixels

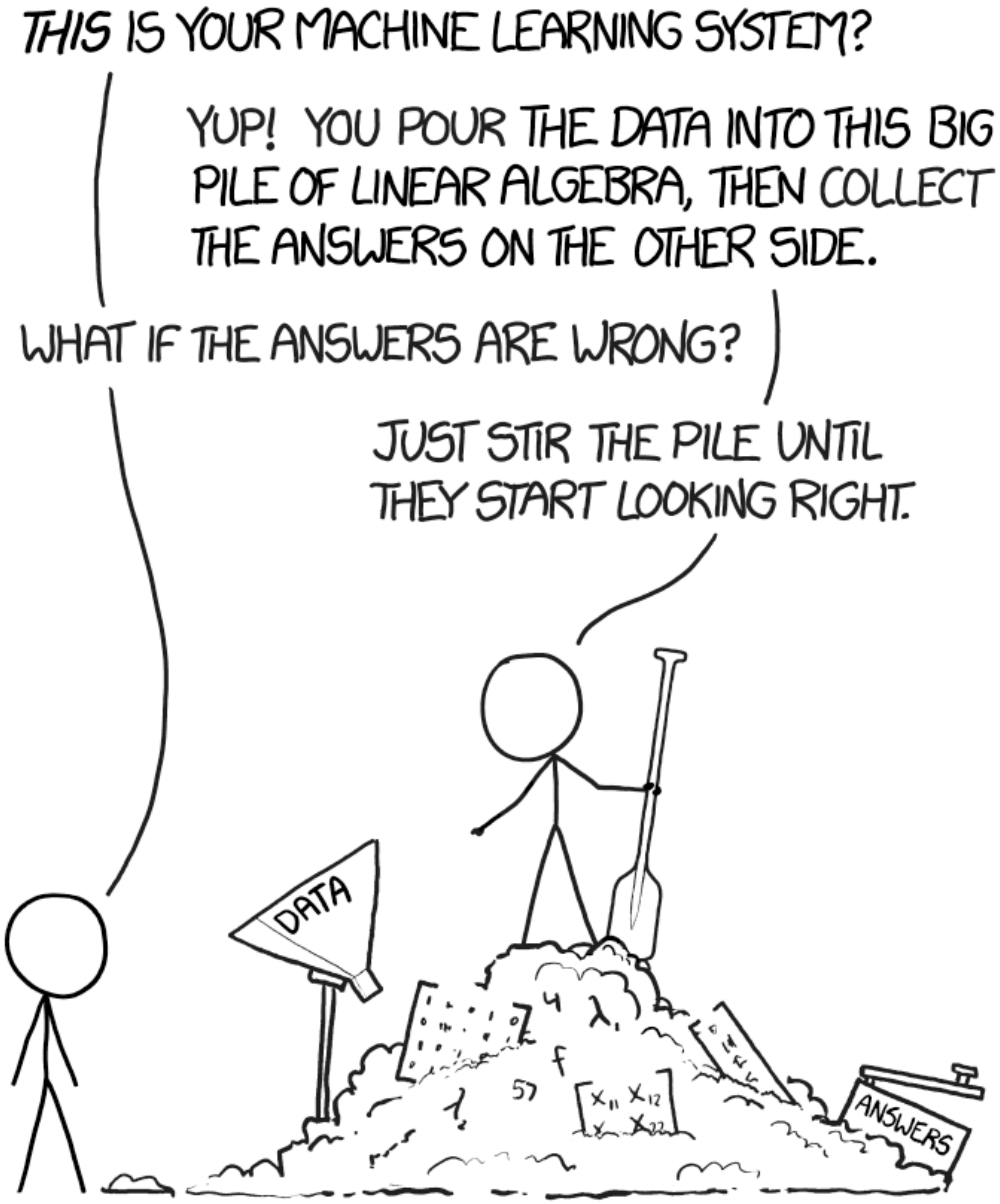


MNIST - 784 total pixels each ranges from 0 to 1

Specific pixel values are very different between images

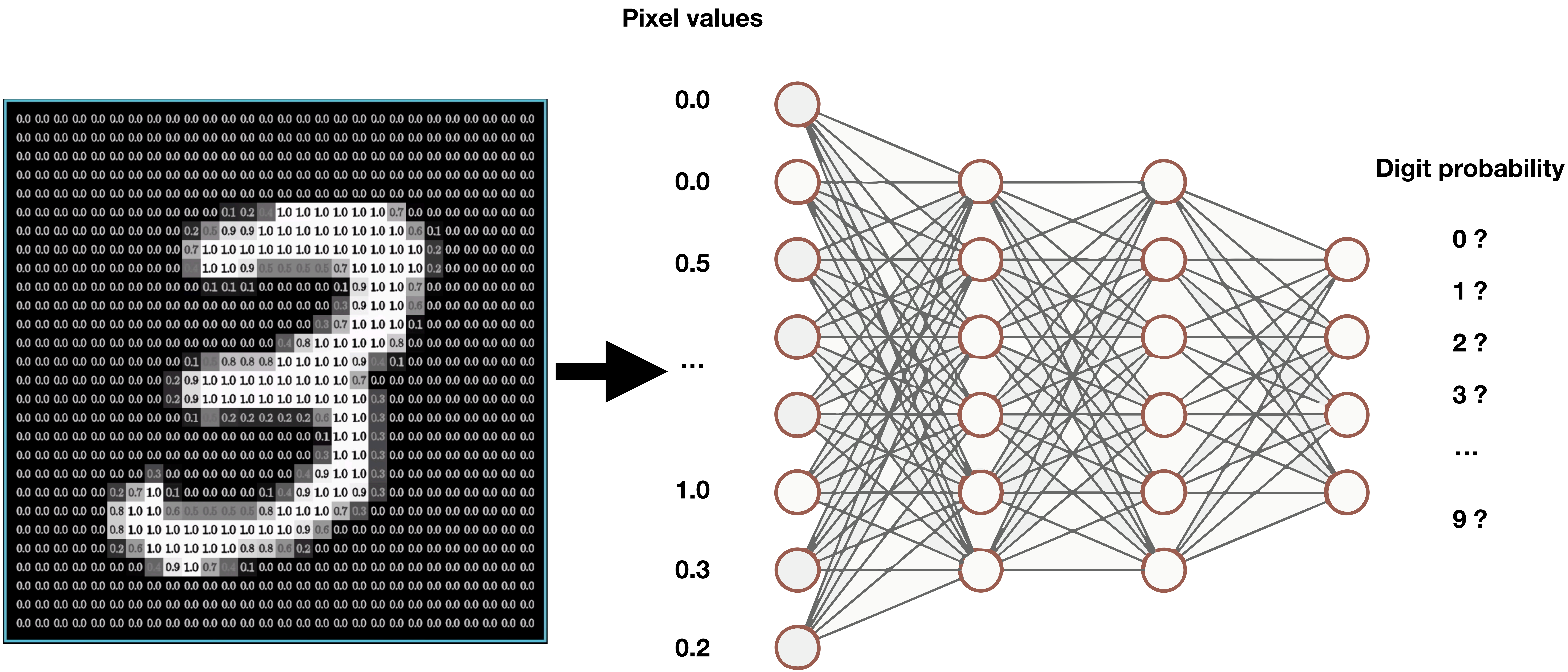
3Blue1Brown (YouTube)

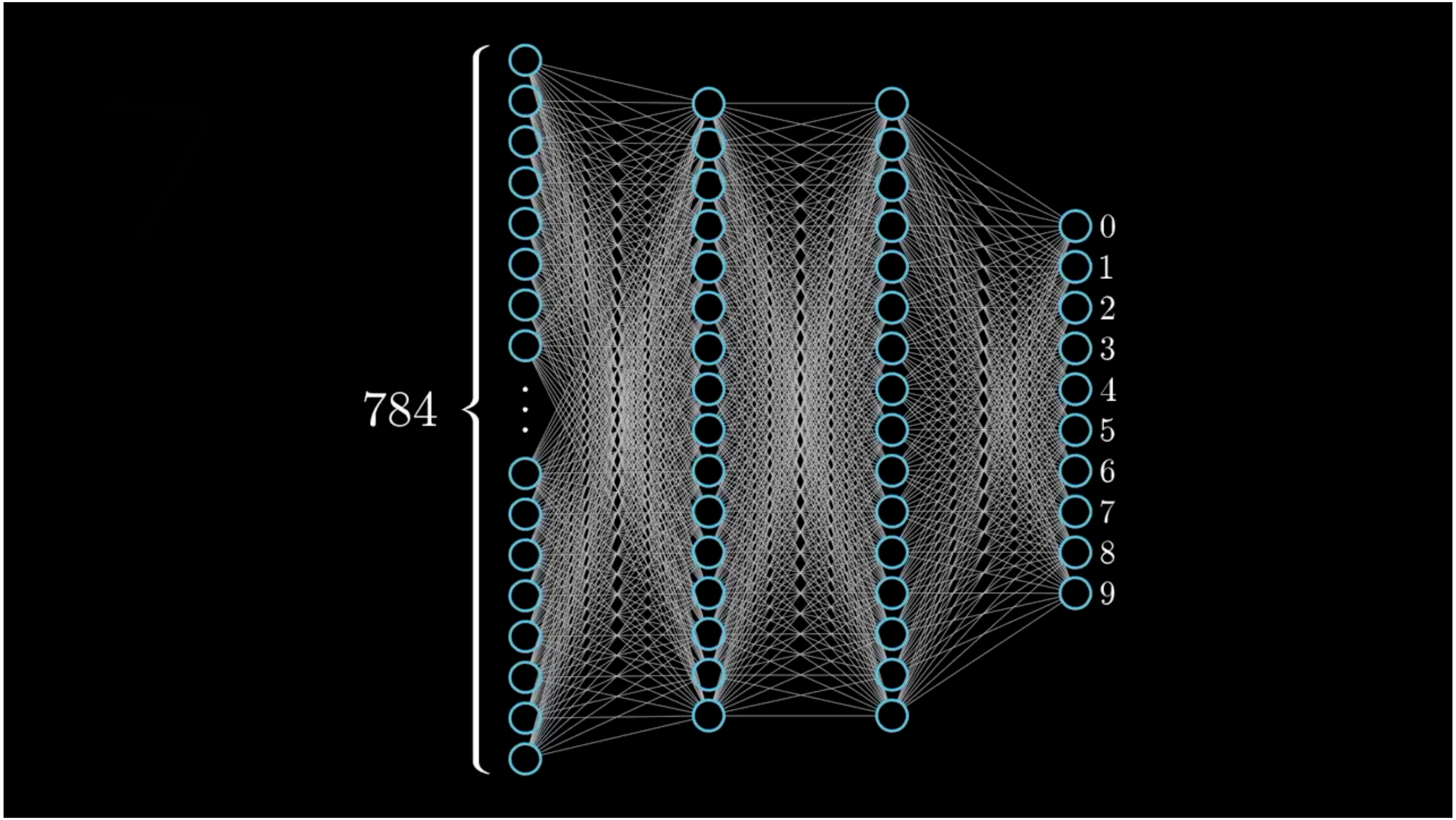
But what is a Neural Network? | Deep learning, chapter 1



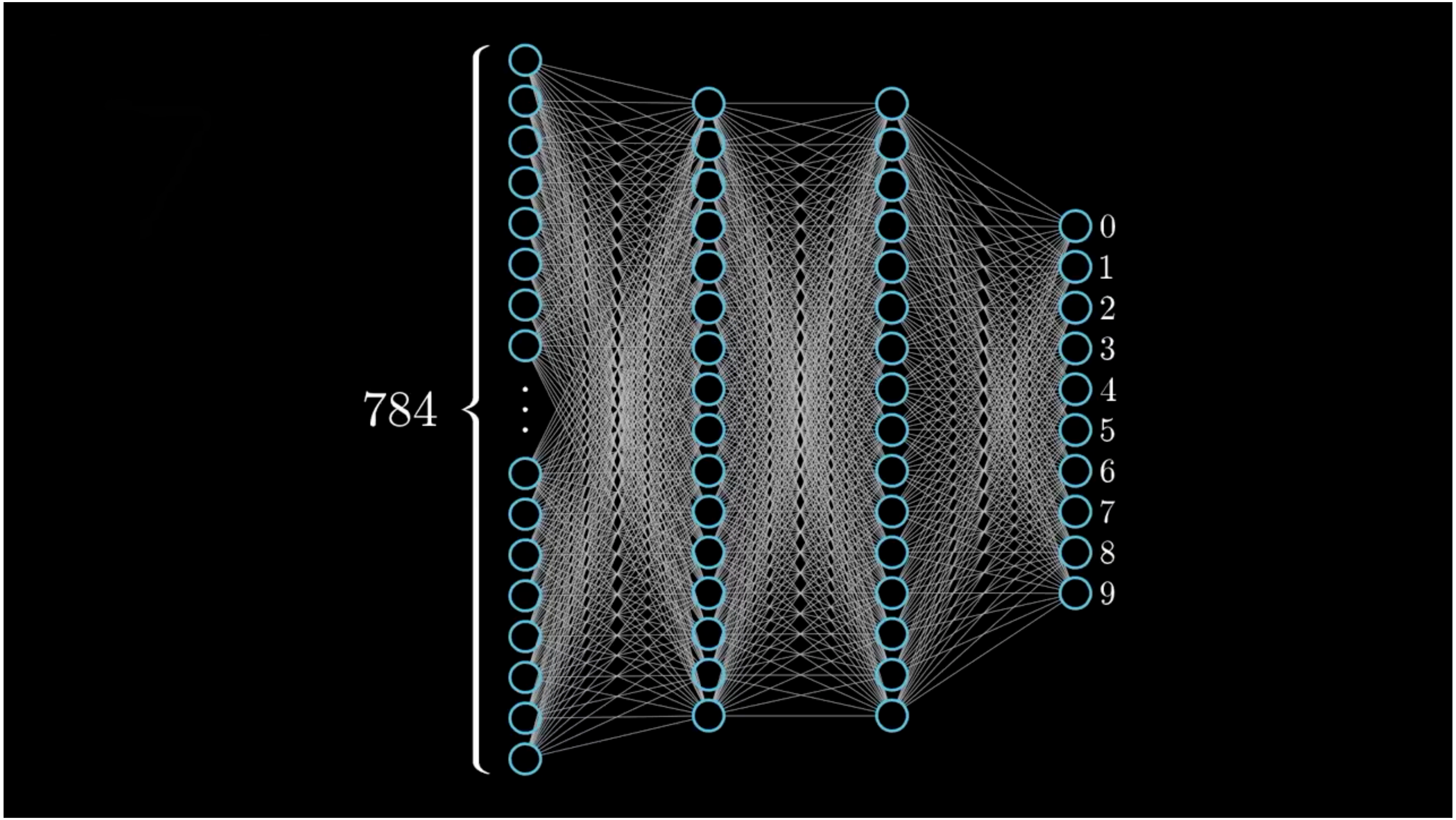
https://xkcd.com/1838/

MLP for digit classification





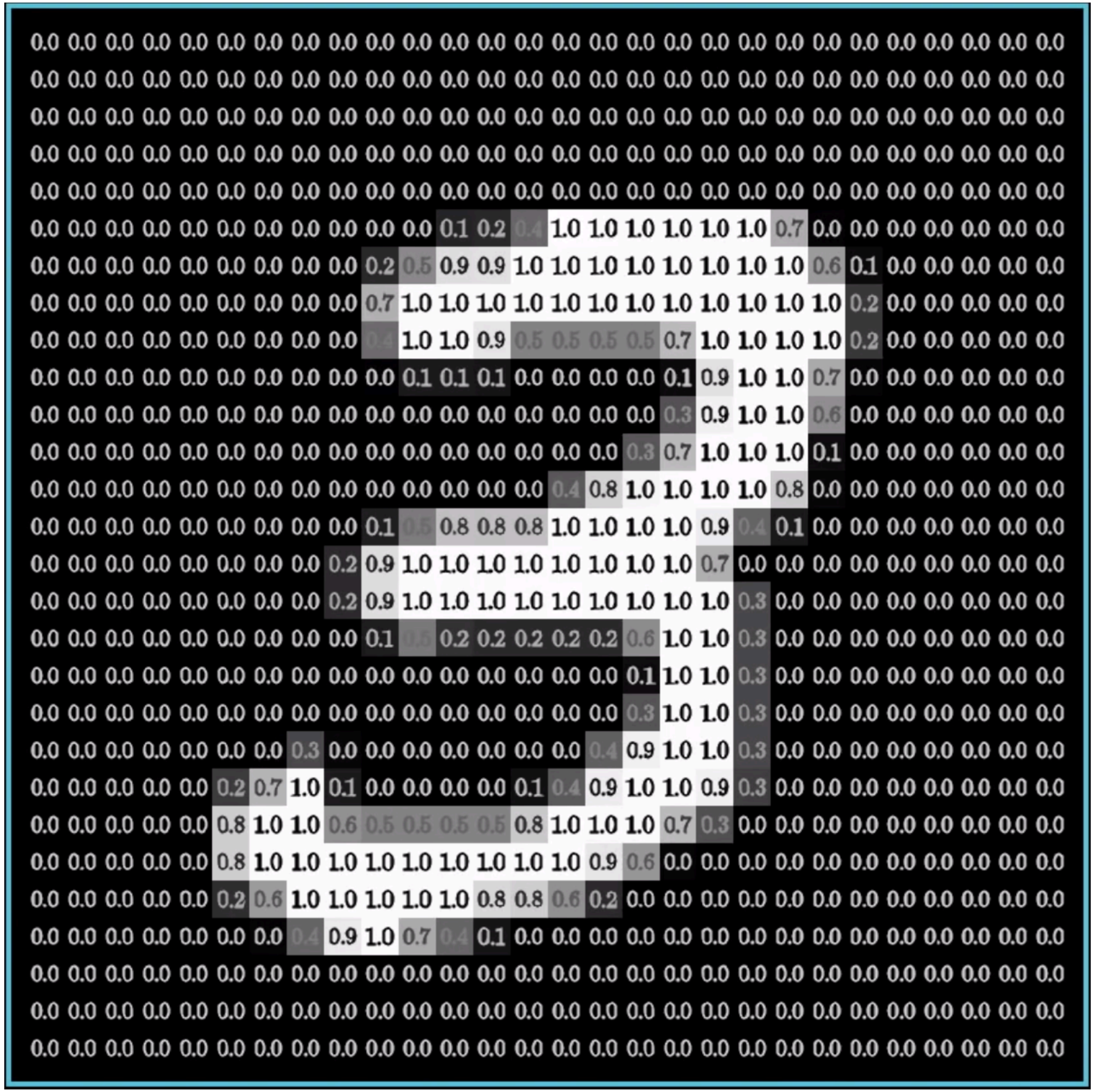
3Blue1Brown (YouTube)
But what is a Neural Network? | Deep learning, chapter 1



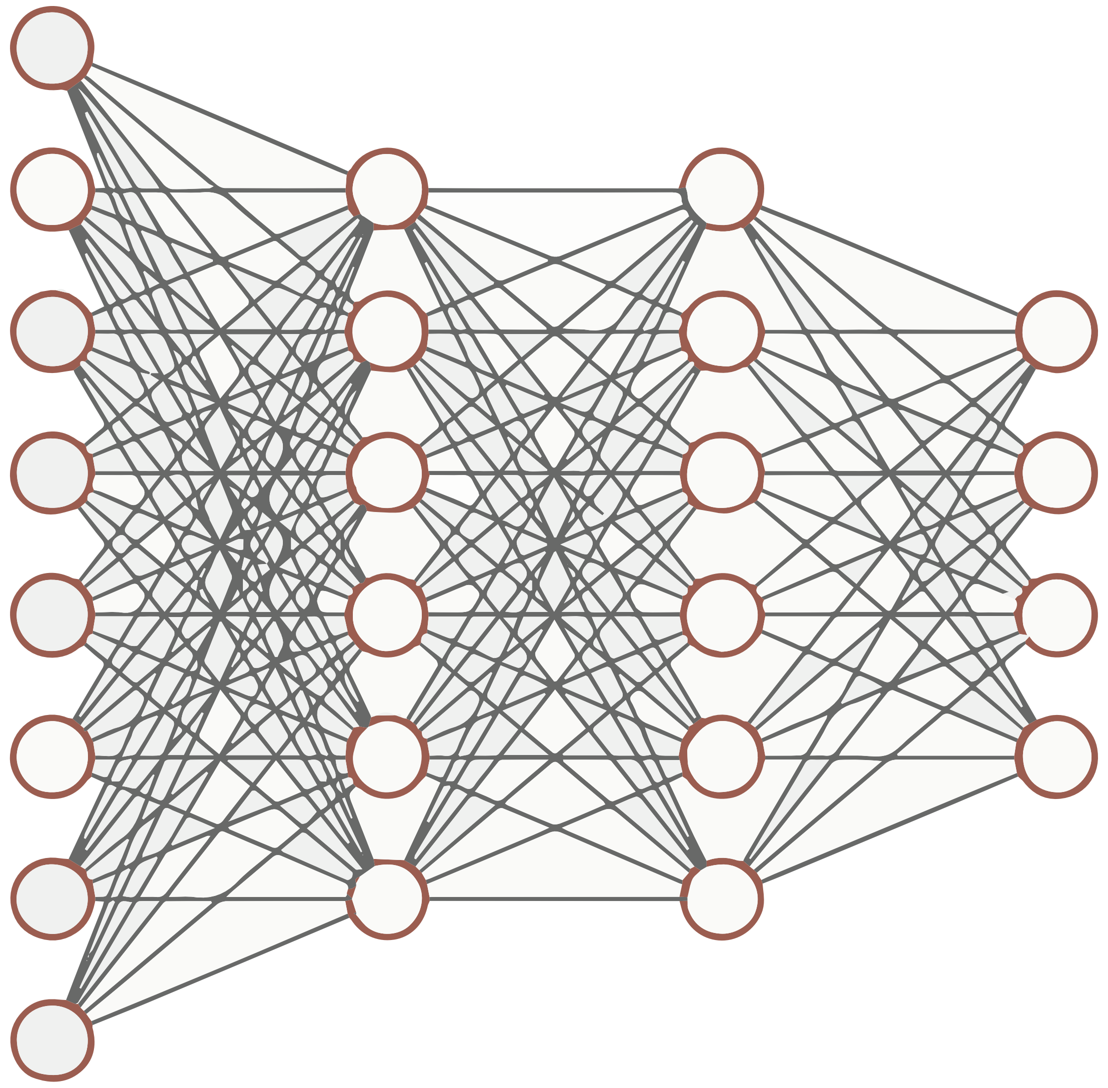
3Blue1Brown (YouTube)
But what is a Neural Network? | Deep learning, chapter 1

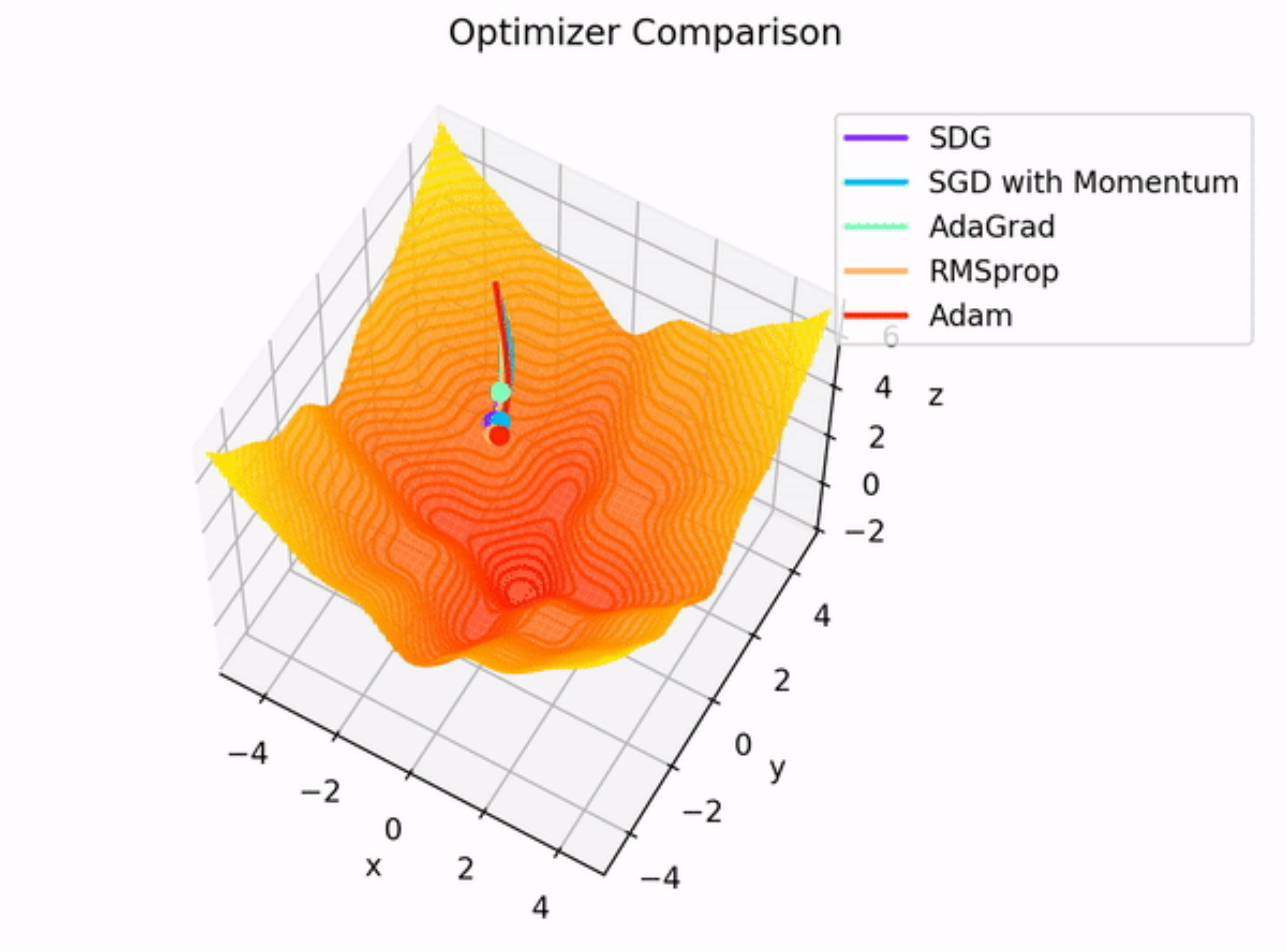
28 pixels

28 pixels

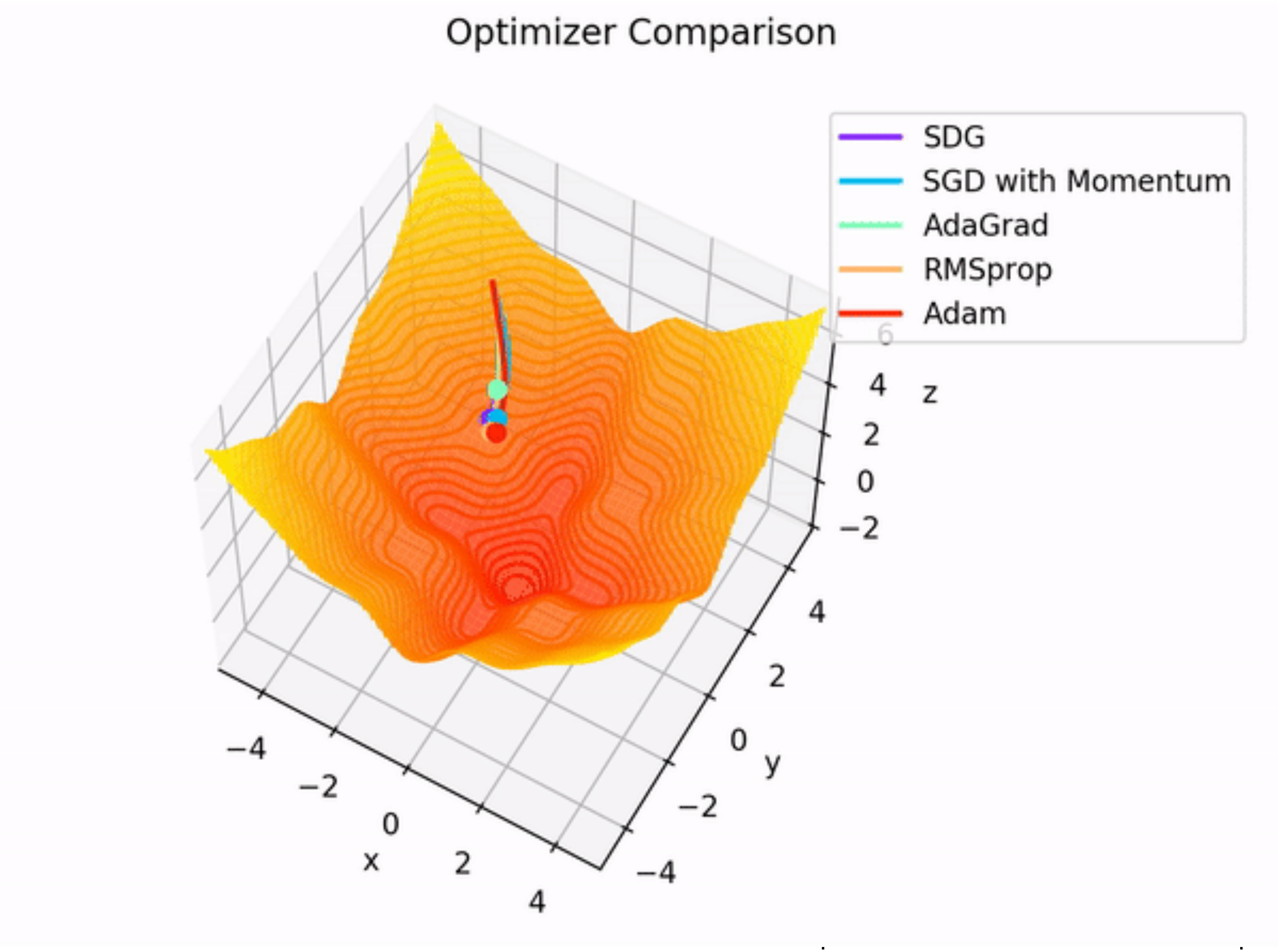


784-dimensional input





te-guide-to-adam-
optimization-1e5f29532c3d



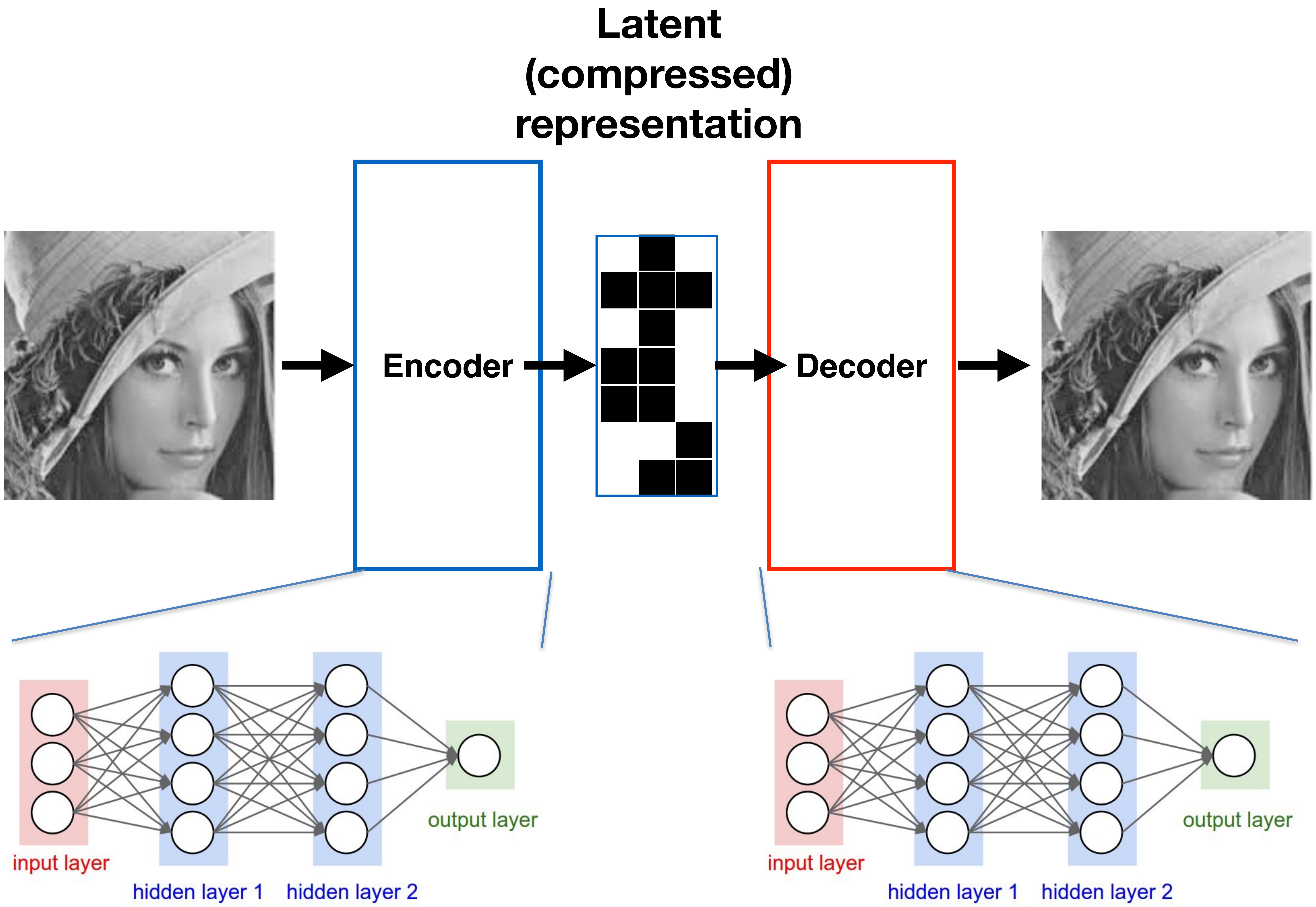
te-guide-to-adam-
optimization-1e5f29532c3d

What you'll hear about in this lecture

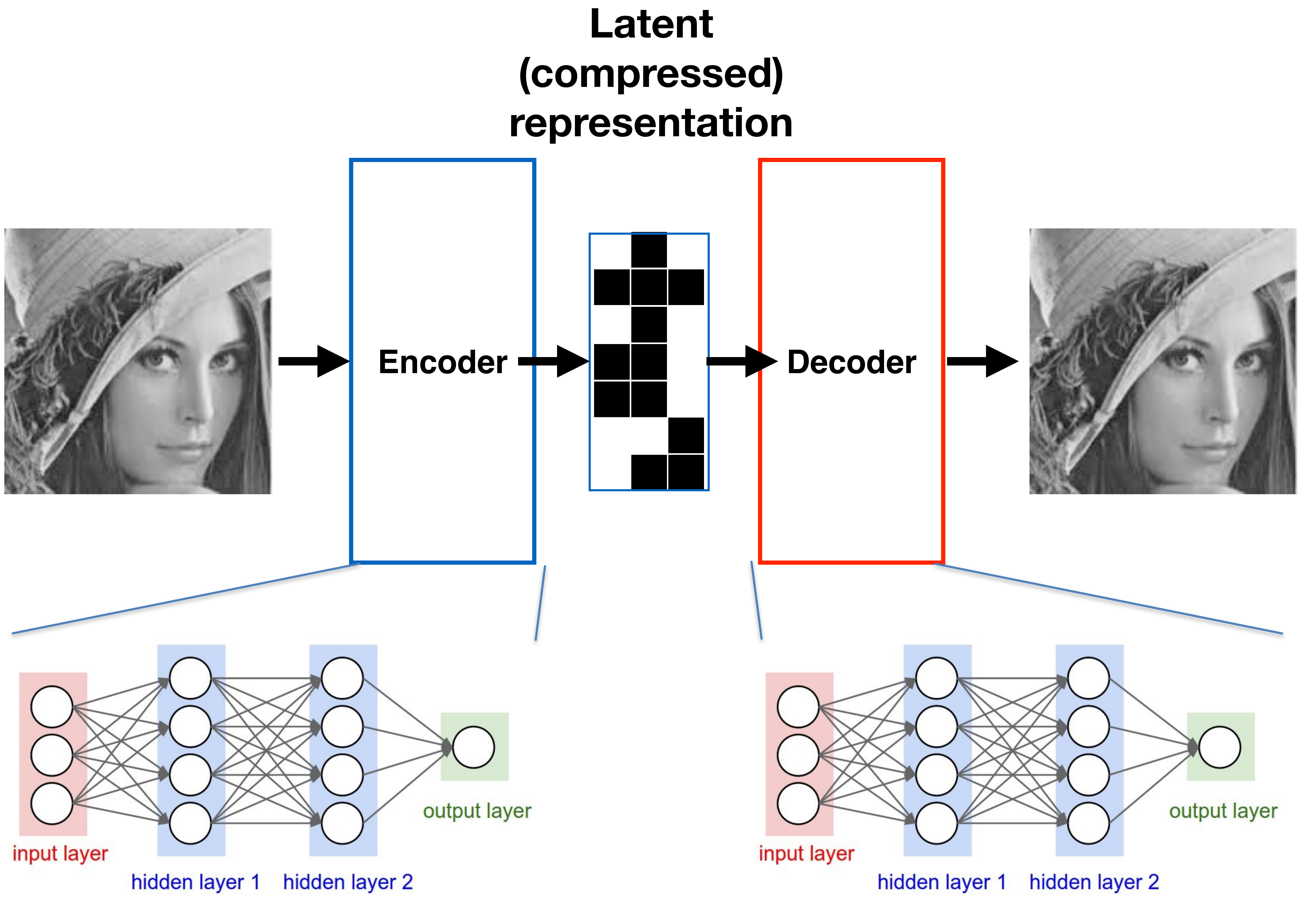
Neural network basics

Deep autoencoders

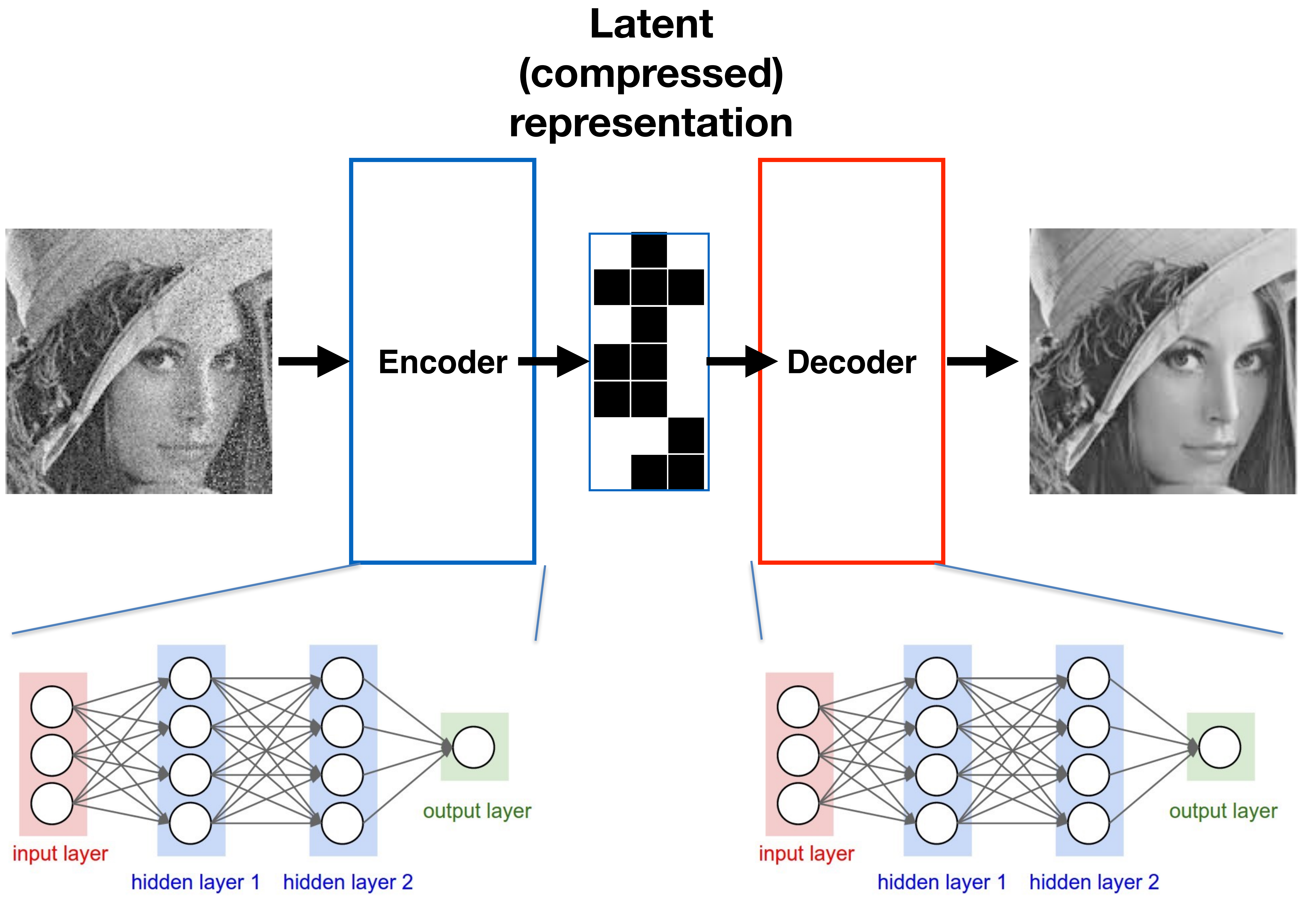
Intro to neural population dynamics



- Autoencoders consist of a pair of networks: an “encoder” and a “decoder”
- The encoder compresses the input down to a low-D representation. The decoder attempts to reconstruct the same input based on the low-D representation



- The compressed representation is a bottleneck - the encoder is forced to throw out irrelevant information and extract the “true” structure underlying the data
- Completely unsupervised - there are no data labels

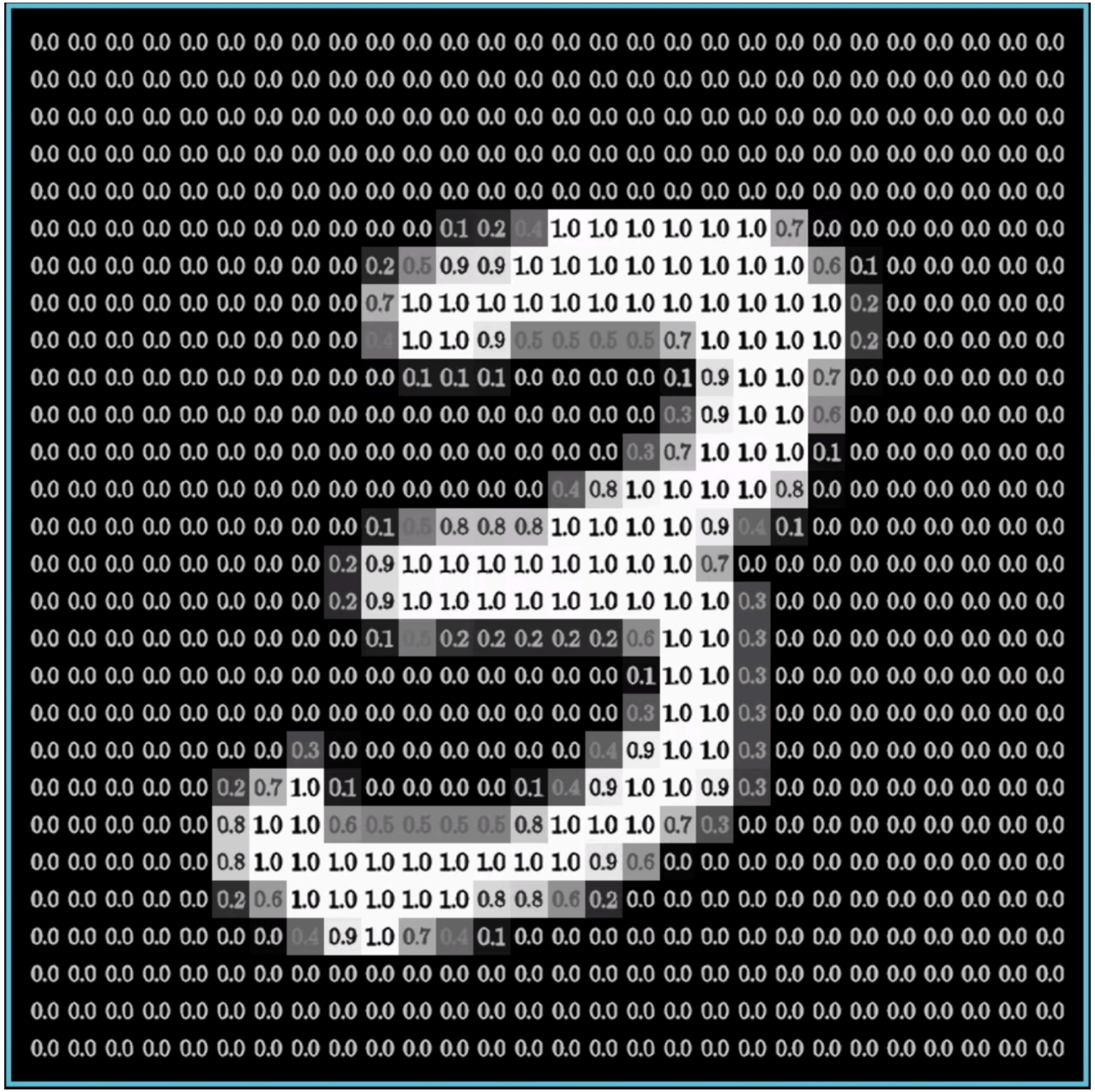


Applications

- An autoencoder can reject noise, that is, it filters out signals that don't correspond to the structure underlying the data

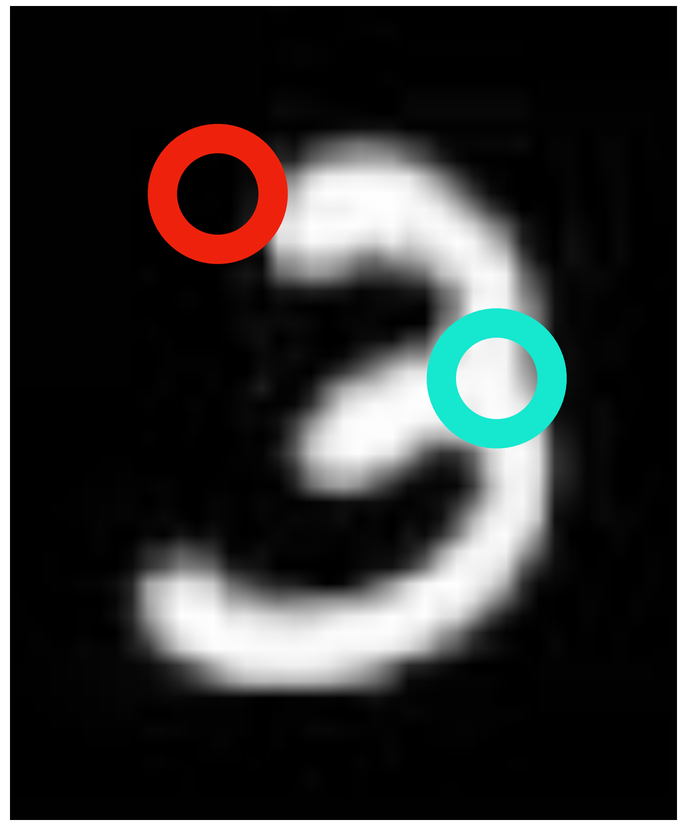
- The decoder must also learn to generate the data, i.e., it is a a generative model

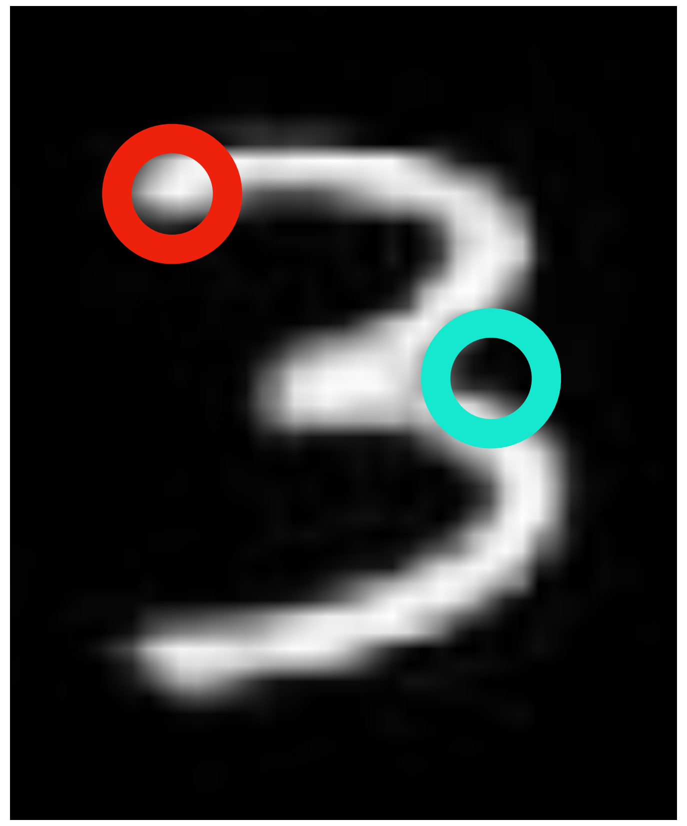
28 pixels



28 pixels

MNIST - 784 total pixels each ranges from 0 to 1

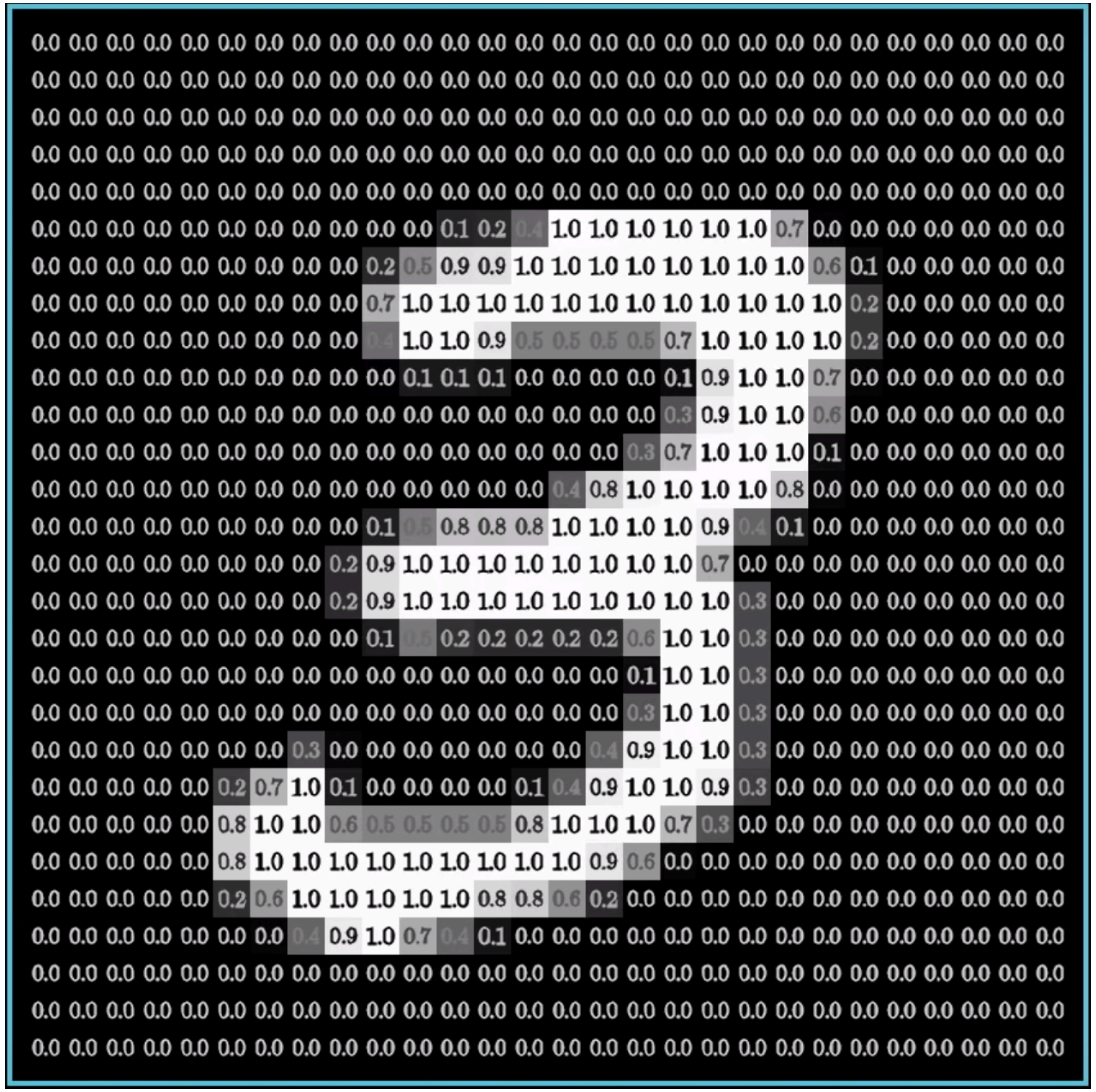


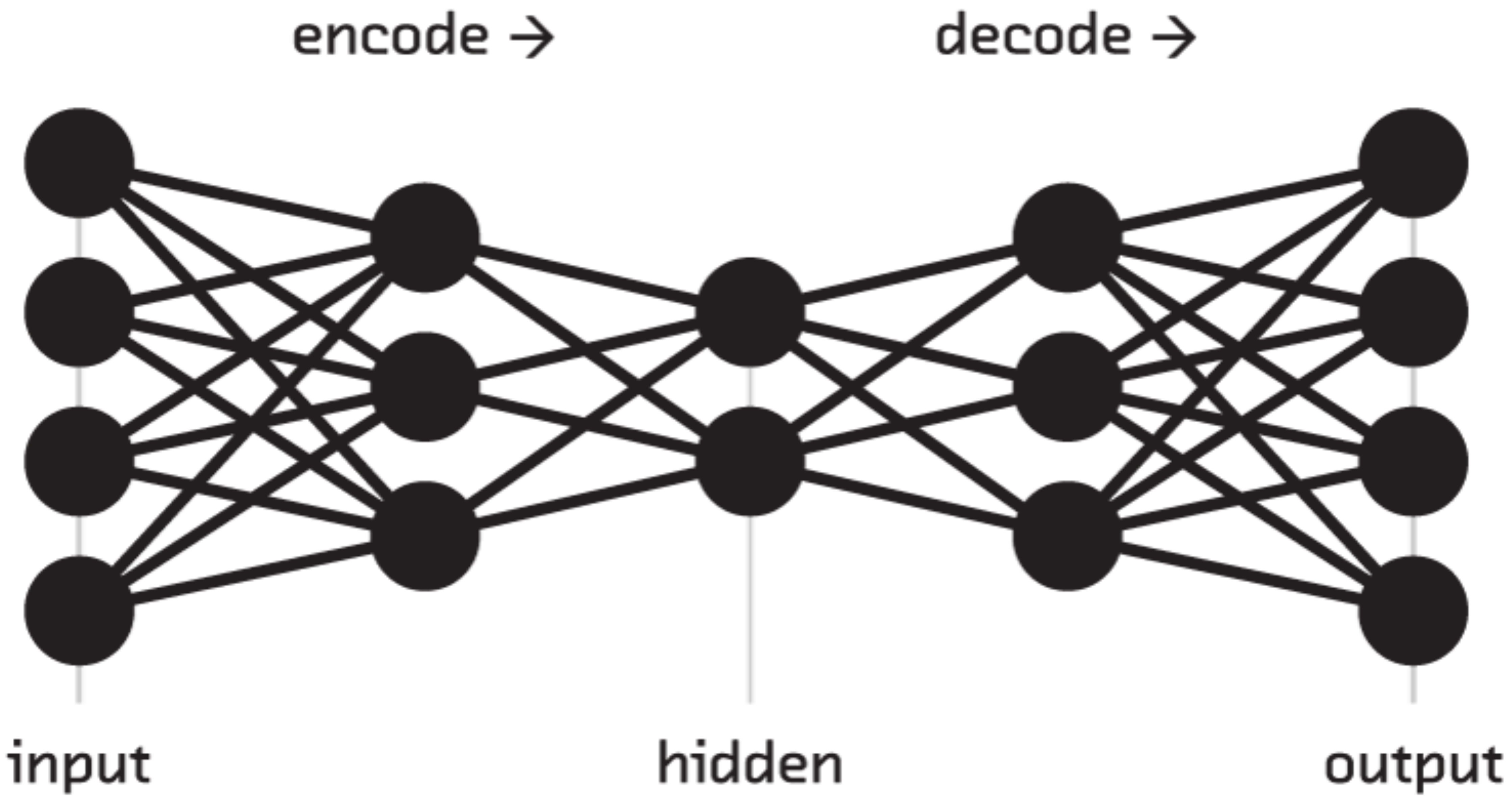


Specific pixel values are very different between images

3Blue1Brown (YouTube)

But what is a Neural Network? | Deep learning, chapter 1

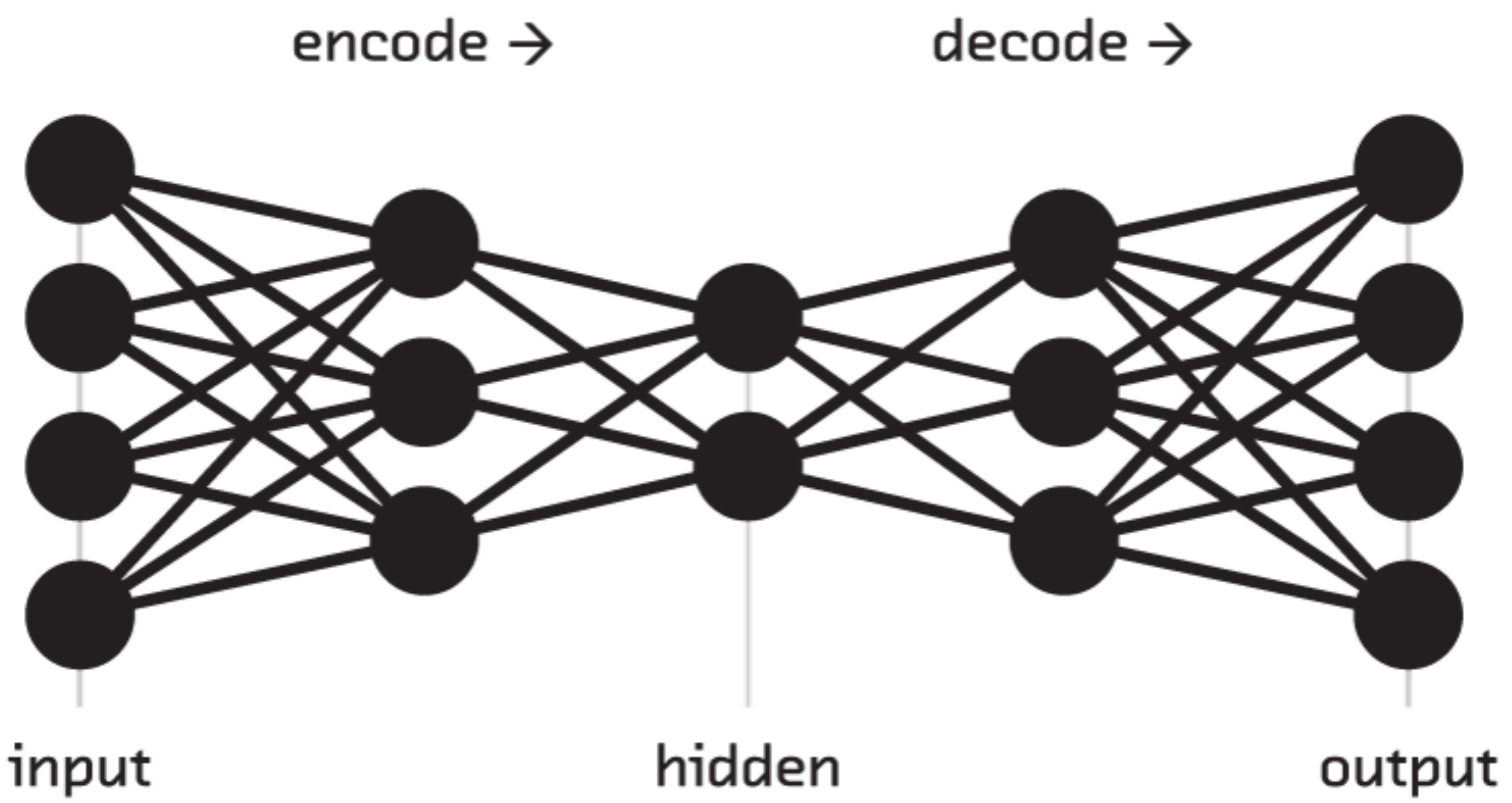




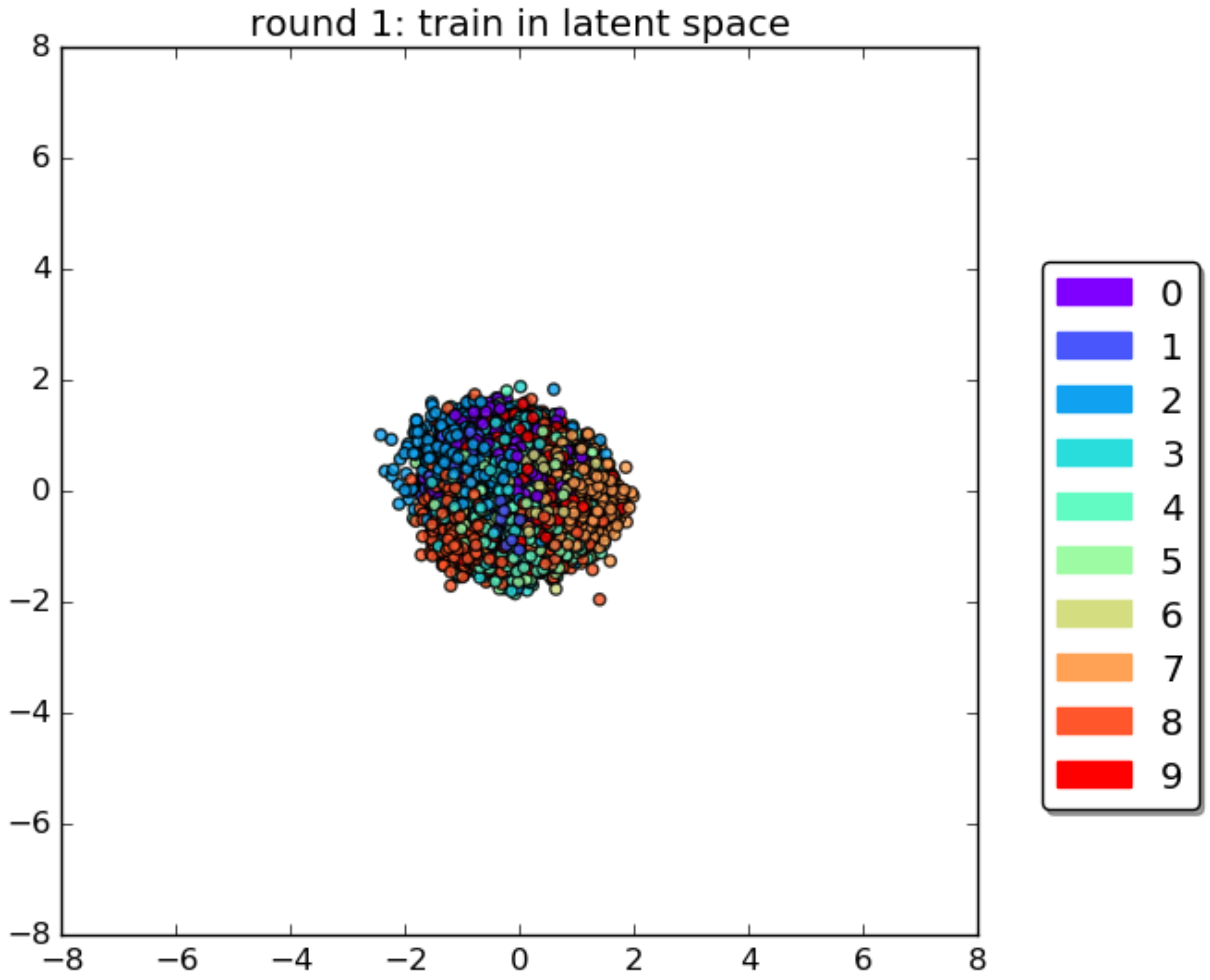
- Network weights are typically initialized completely randomly

- Through the course of training, the latent representation begins to exhibit structure that matches the structure in the data

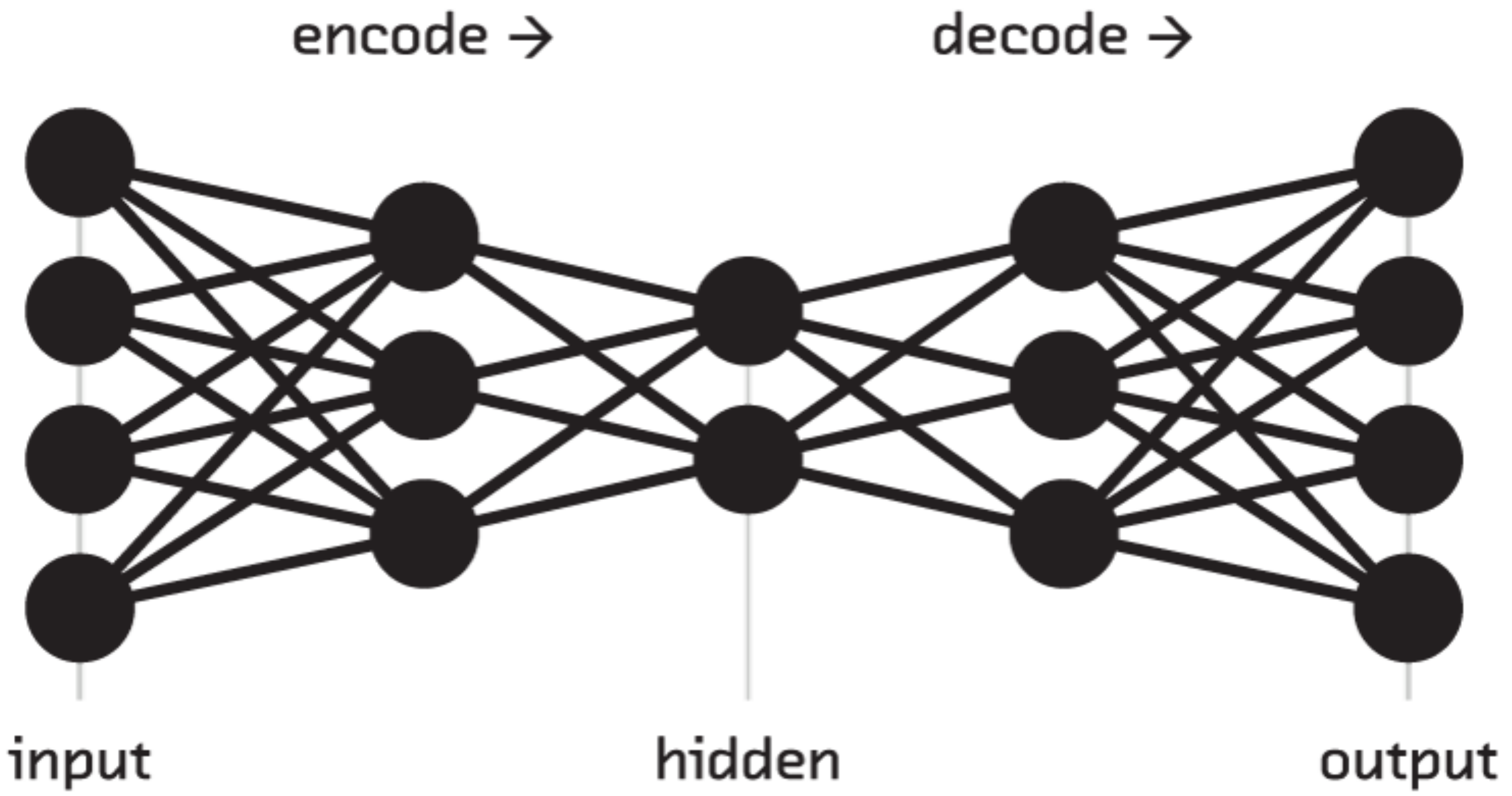
https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and.html

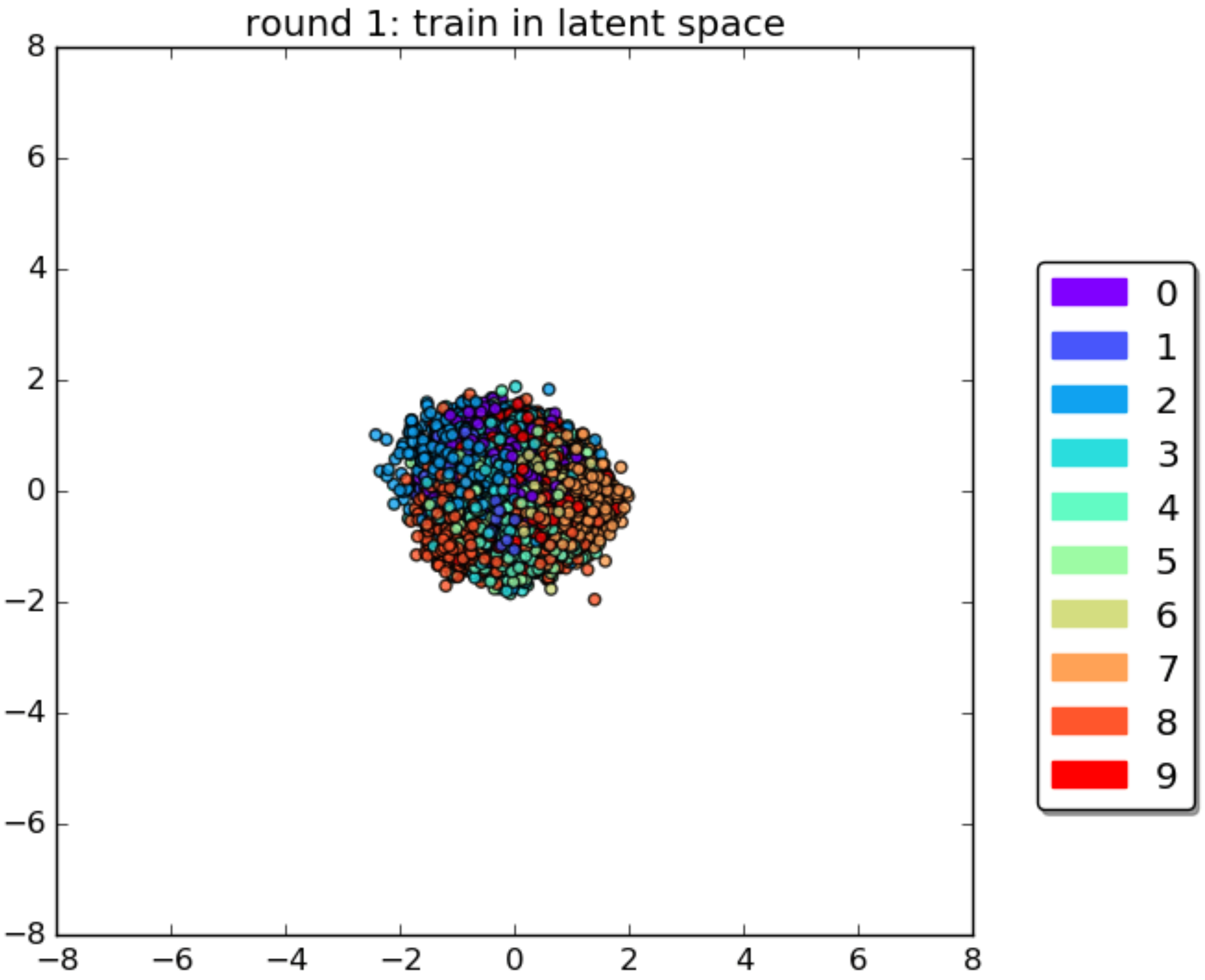


- Through the course of training, the latent representation begins to exhibit structure that matches the structure in the data



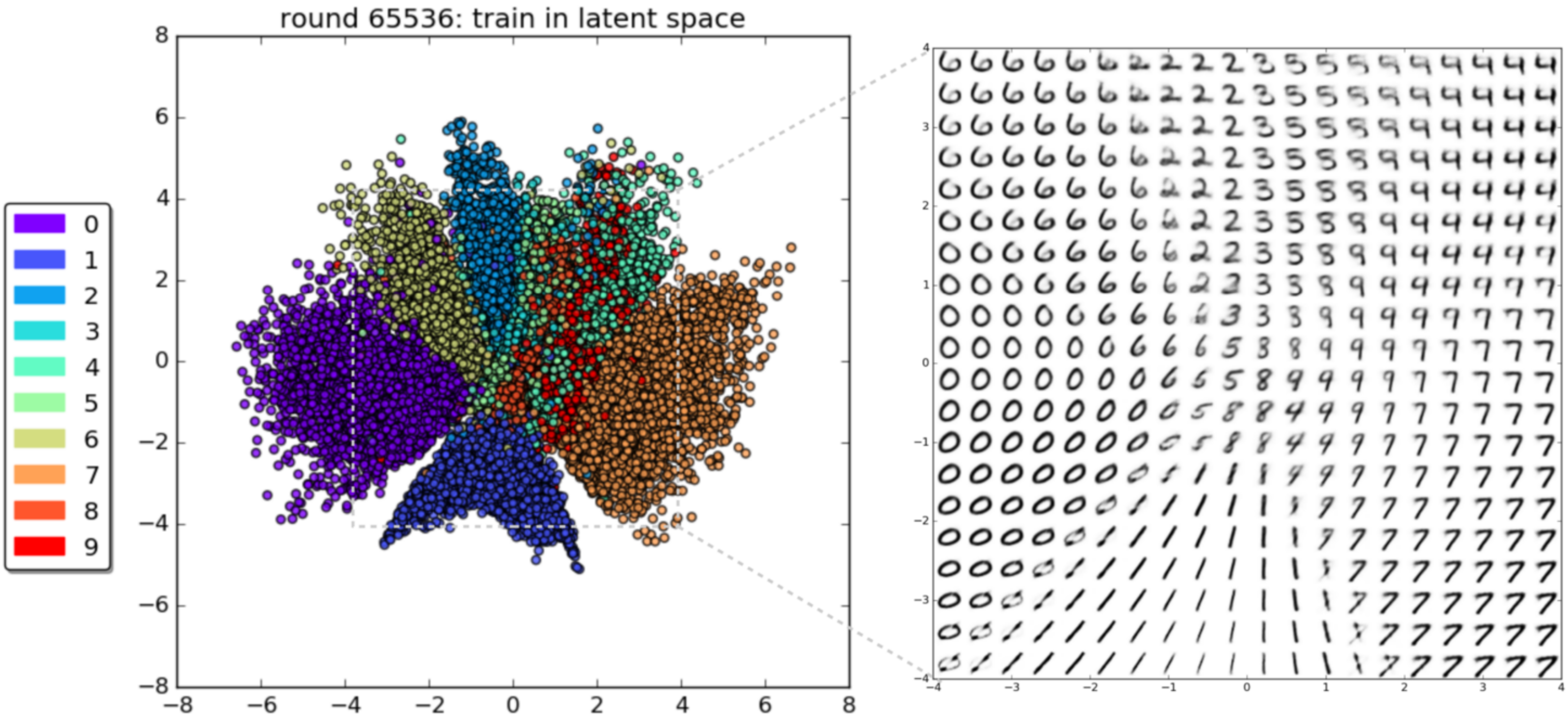
https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and.html





- Through the course of training, the latent representation begins to exhibit structure that matches the structure in the data

https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and.html



https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and.html

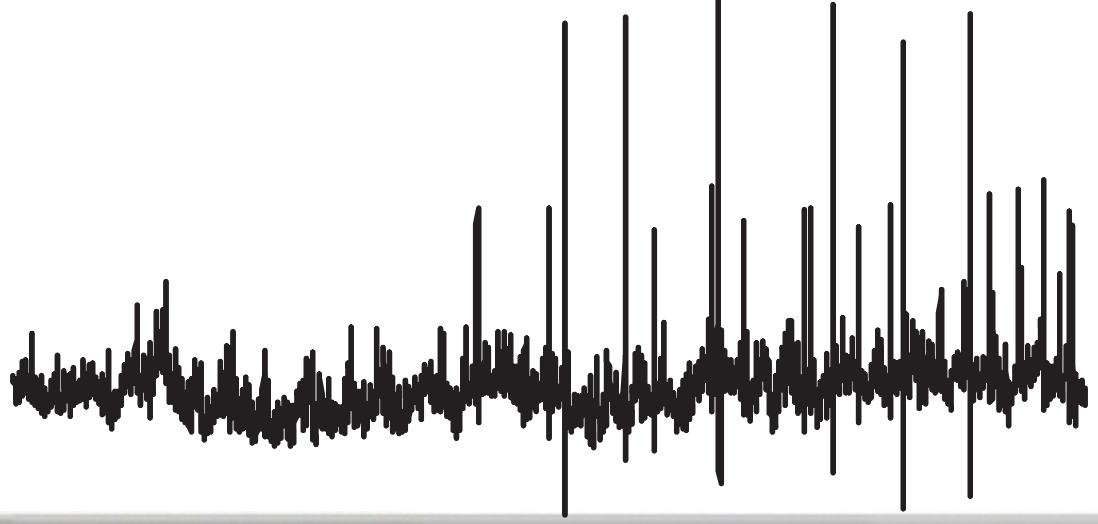
What you'll hear about in this lecture

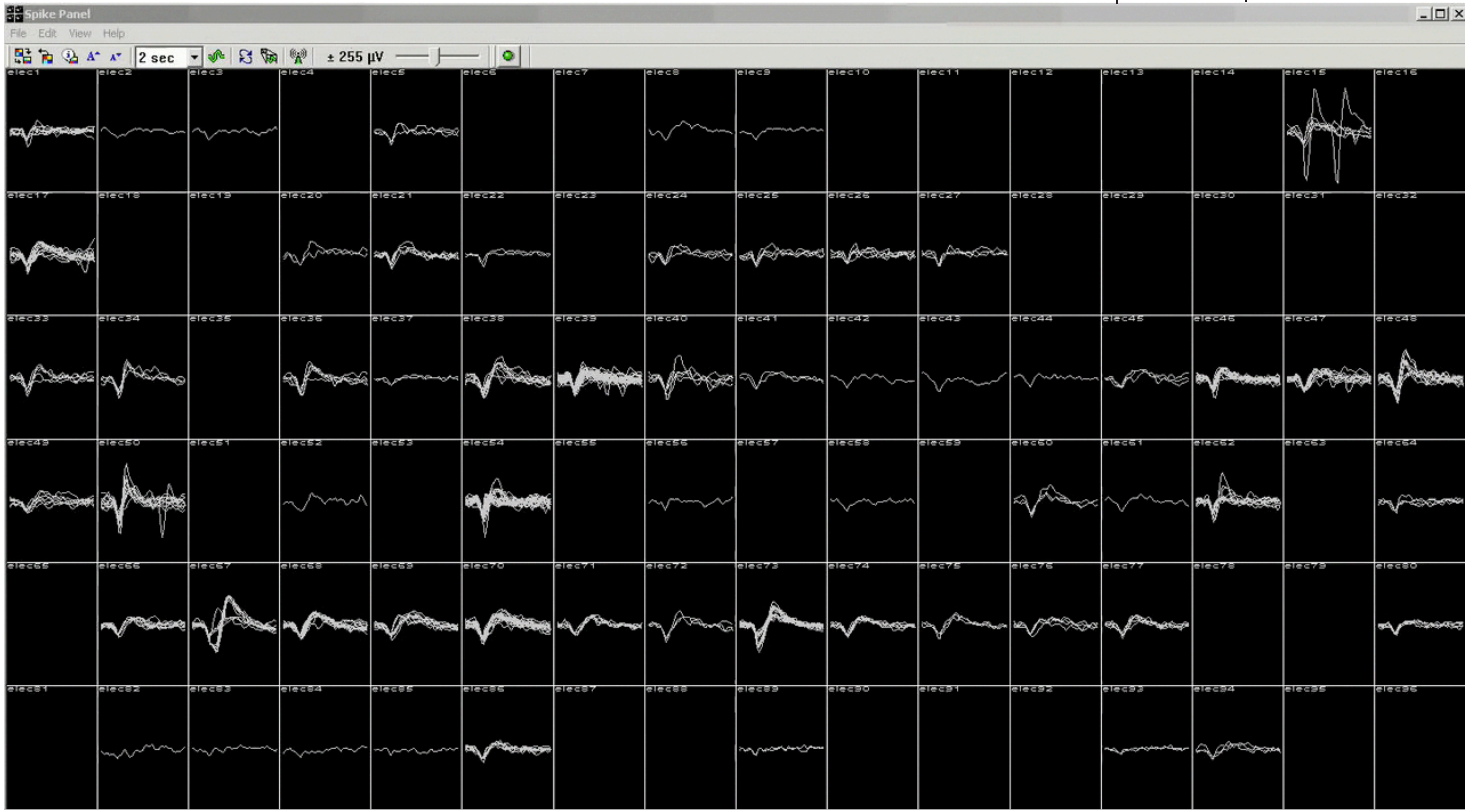
Neural network basics

Deep autoencoders

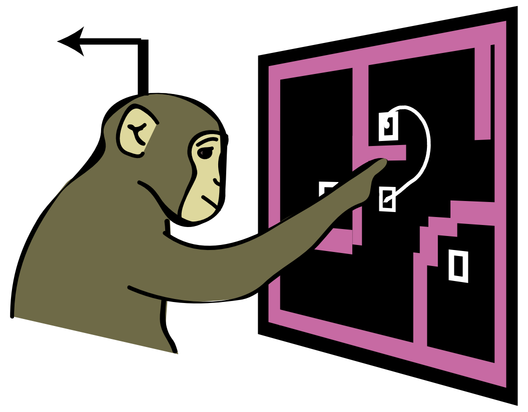
Intro to neural population dynamics

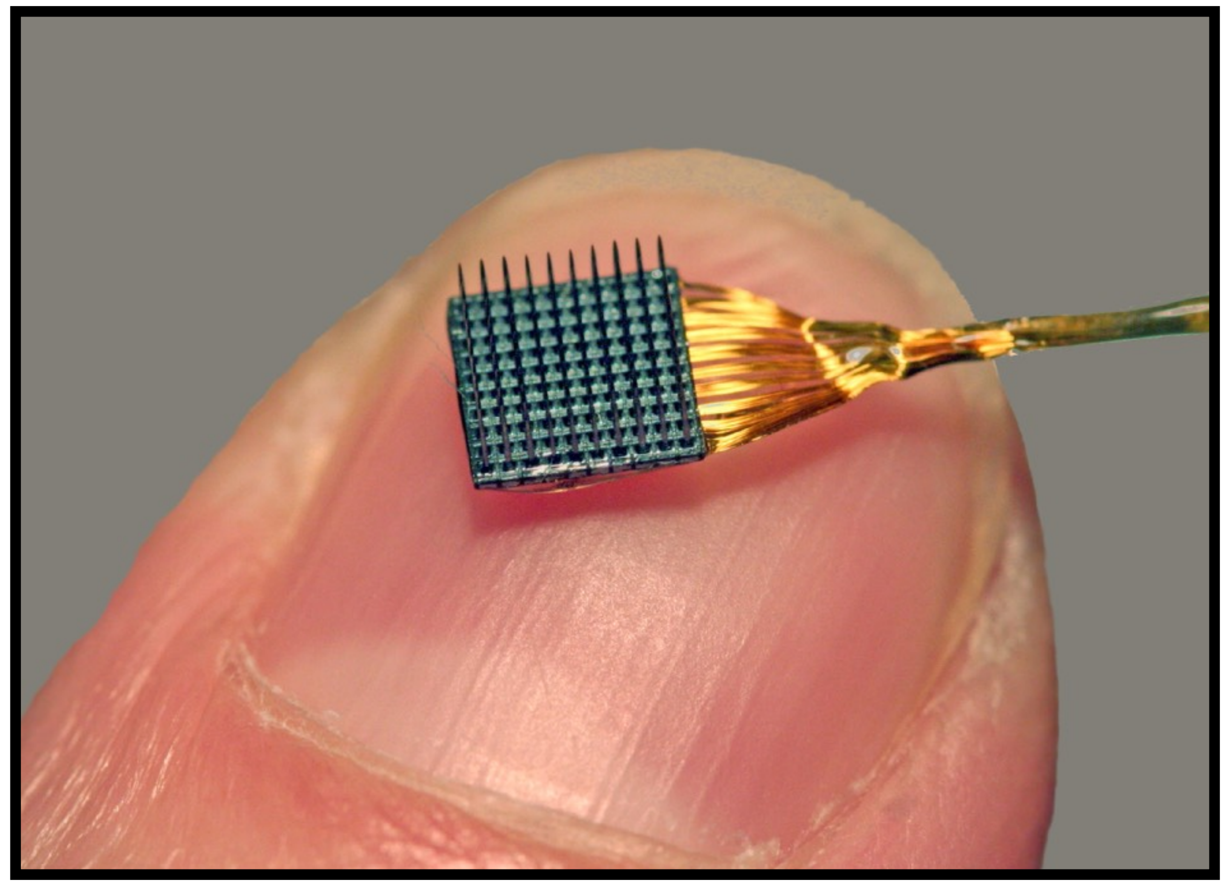
Intracortical electrophysiological recordings



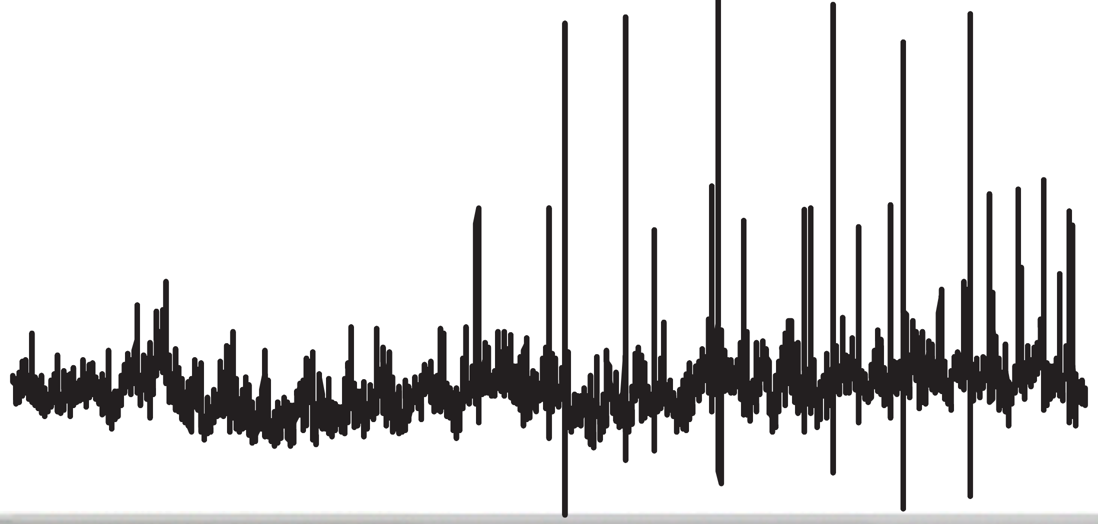


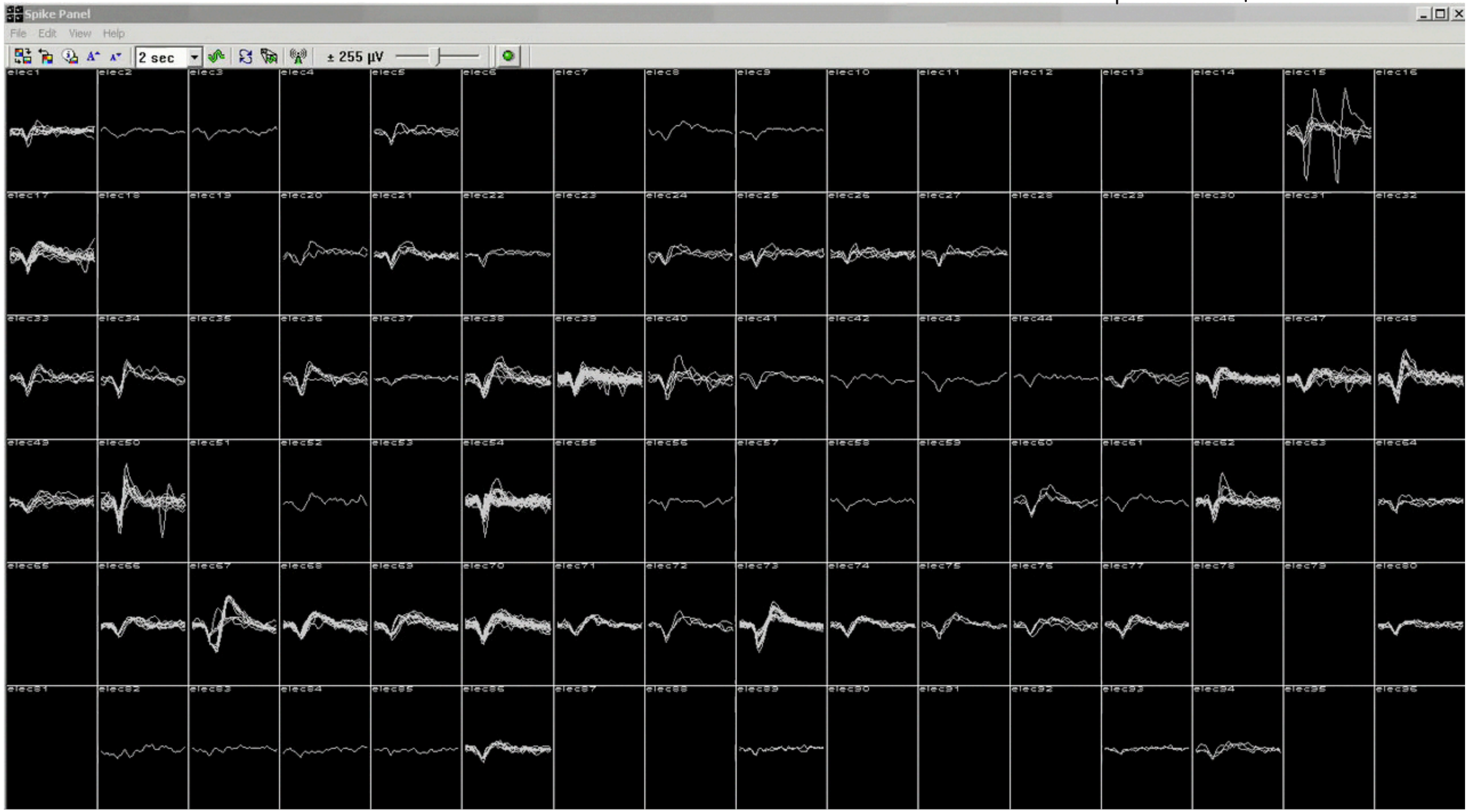
M1/
PMd



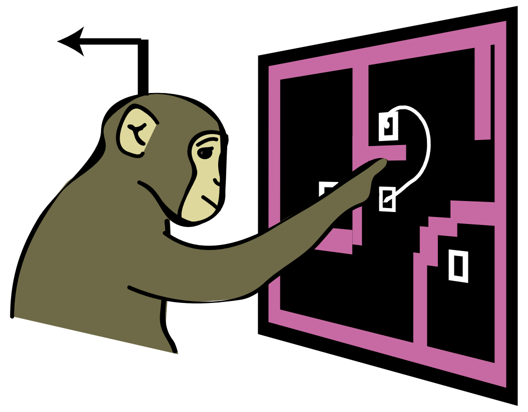


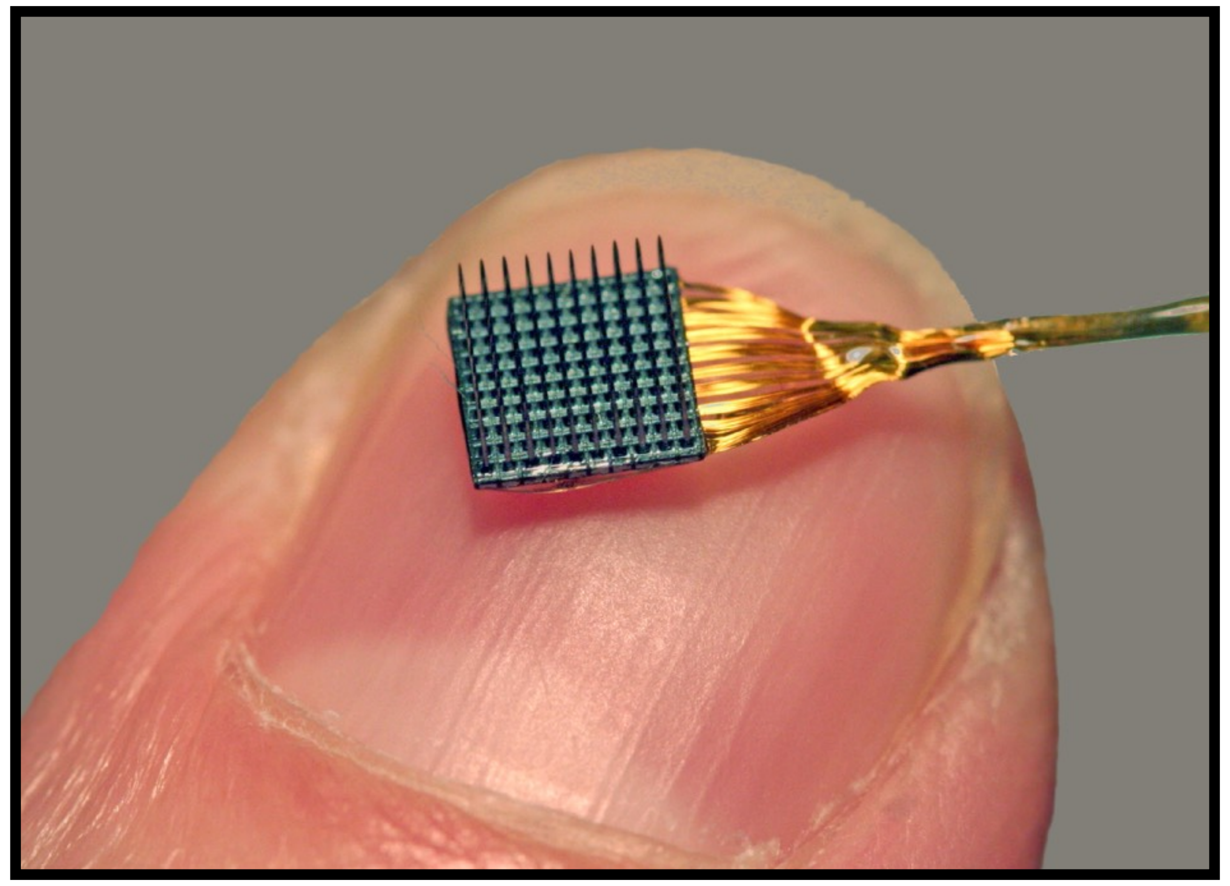
Intracortical electrophysiological recordings



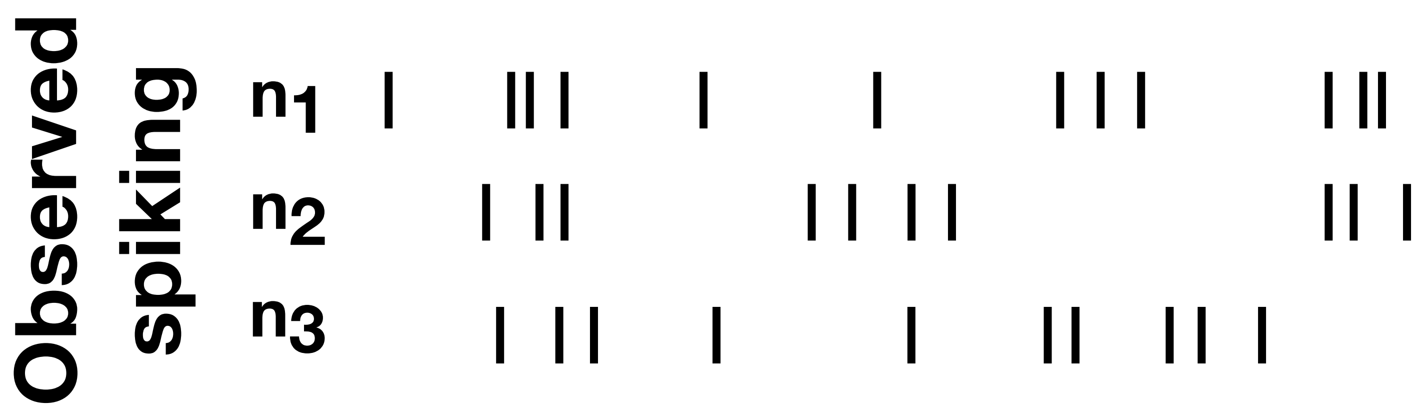


M1/
PMd

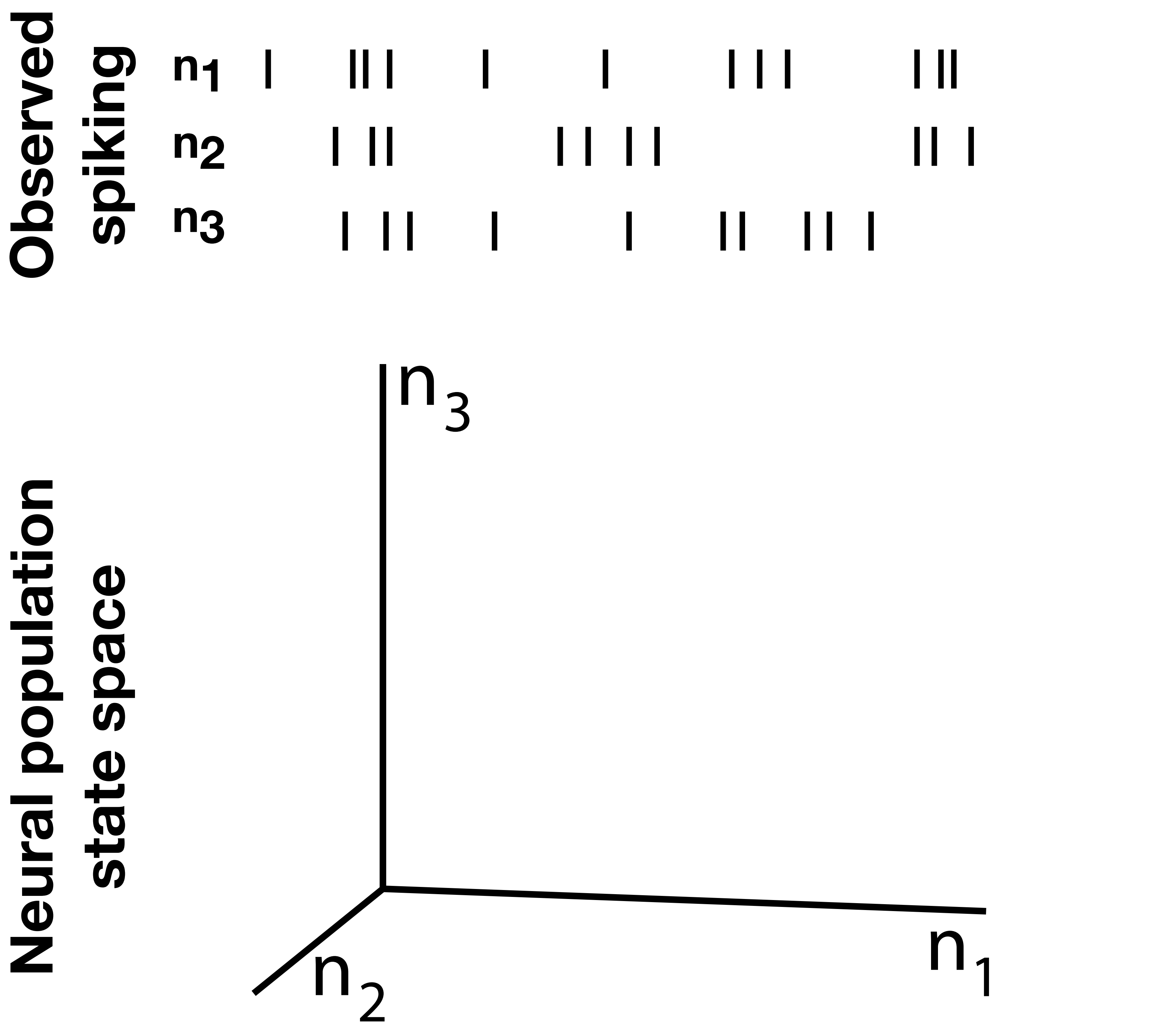




Neural population state spaces

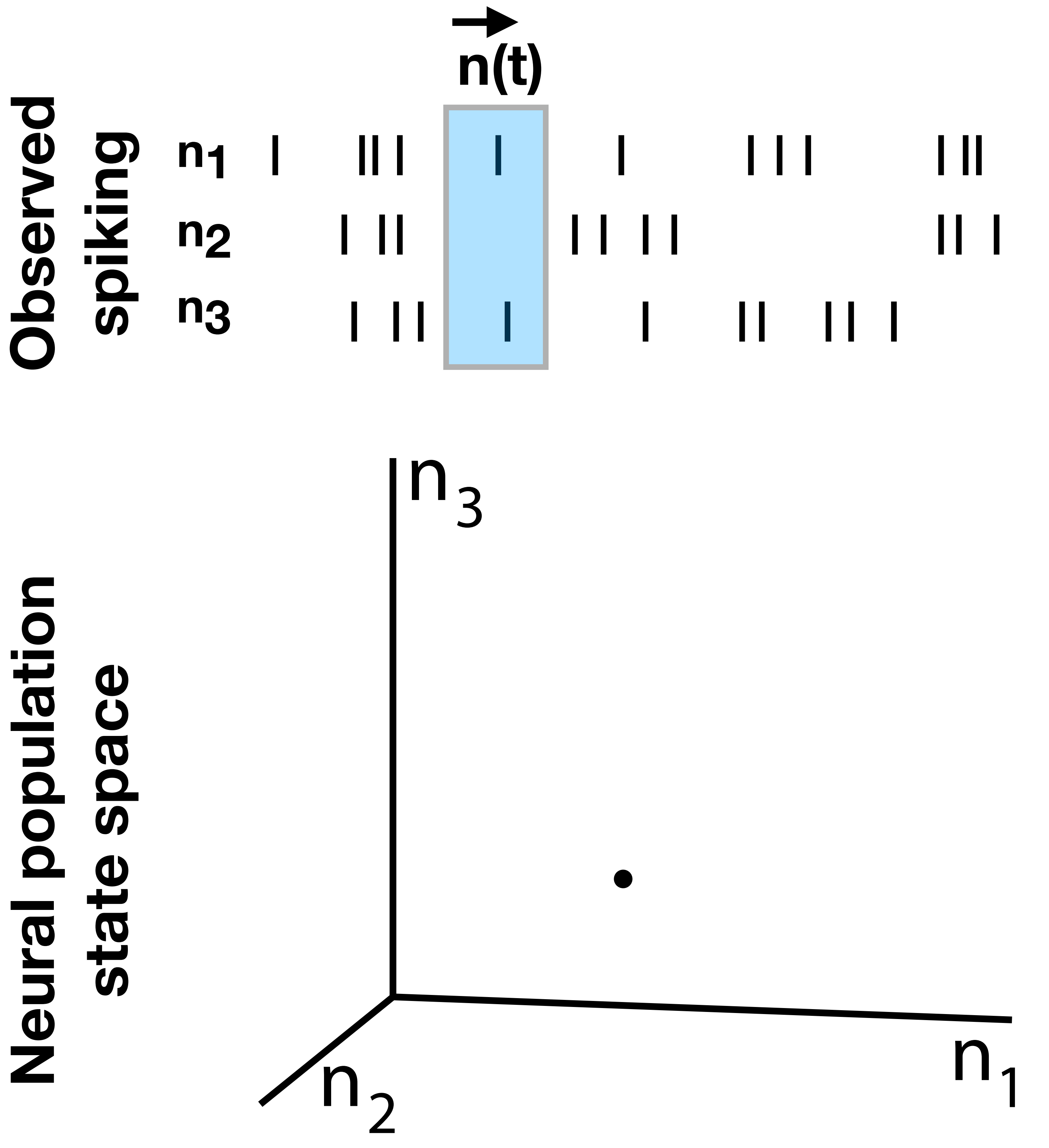


Neural population state spaces



Adapted from Cunningham & Yu, *Nat Neuro* 2014
Dimensionality reduction for large-scale neural recordings

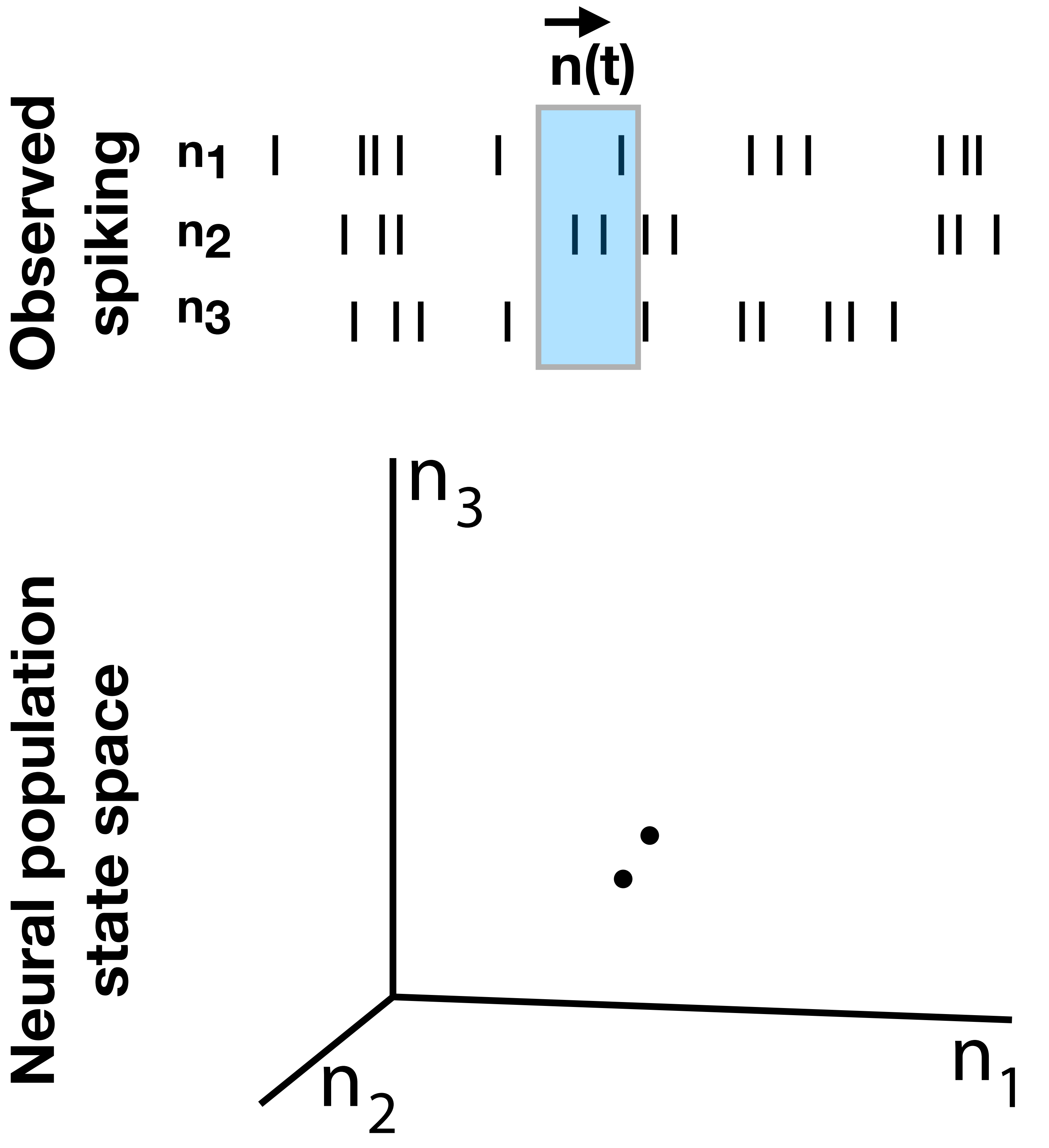
Neural population state spaces



Adapted from Cunningham & Yu, *Nat Neuro* 2014
Dimensionality reduction for large-scale neural recordings

- The current state of the system is captured by the firing rates of all of its neurons

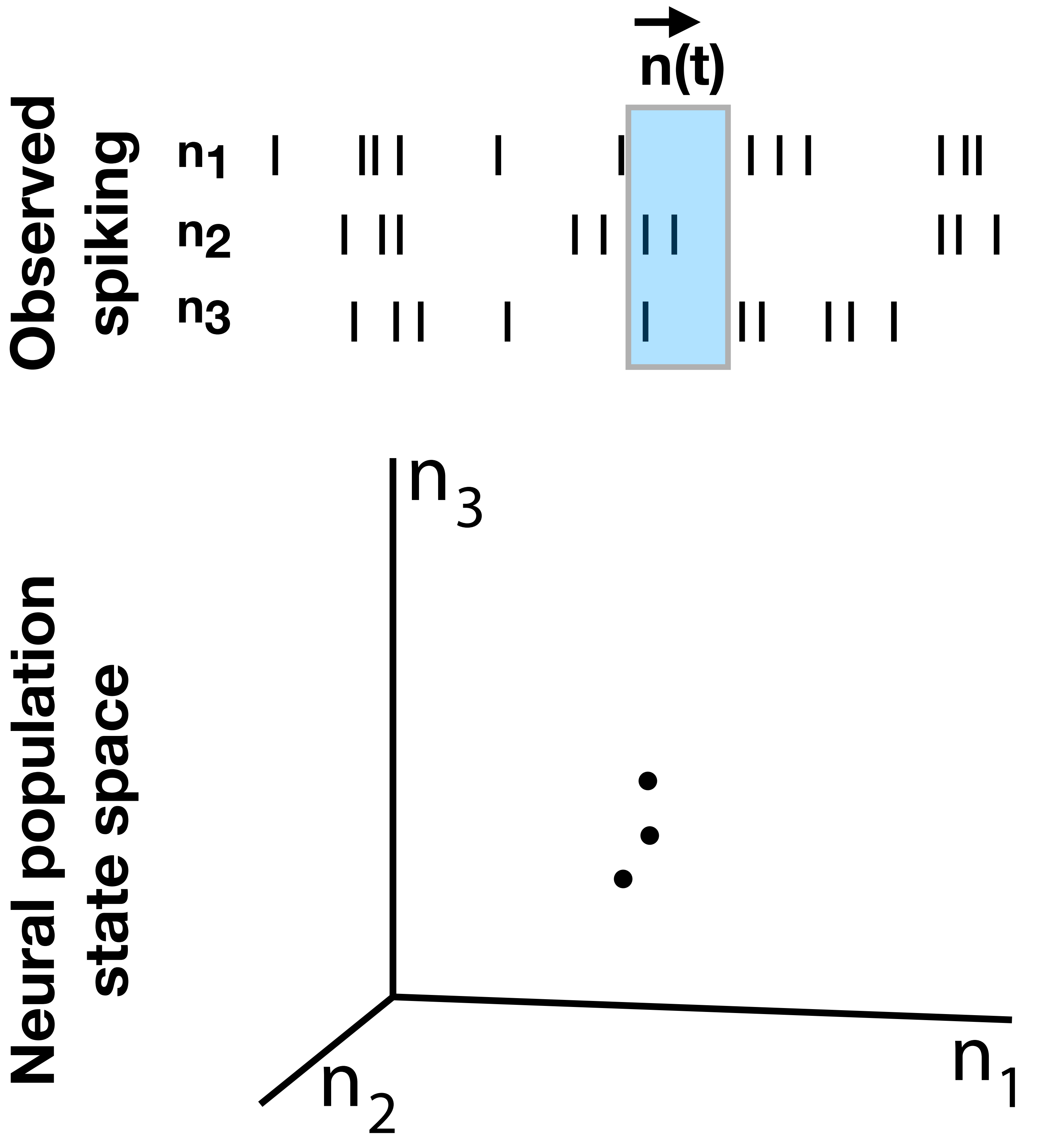
Neural population state spaces



Adapted from Cunningham & Yu, *Nat Neuro* 2014
Dimensionality reduction for large-scale neural recordings

- The current state of the system is captured by the firing rates of all of its neurons

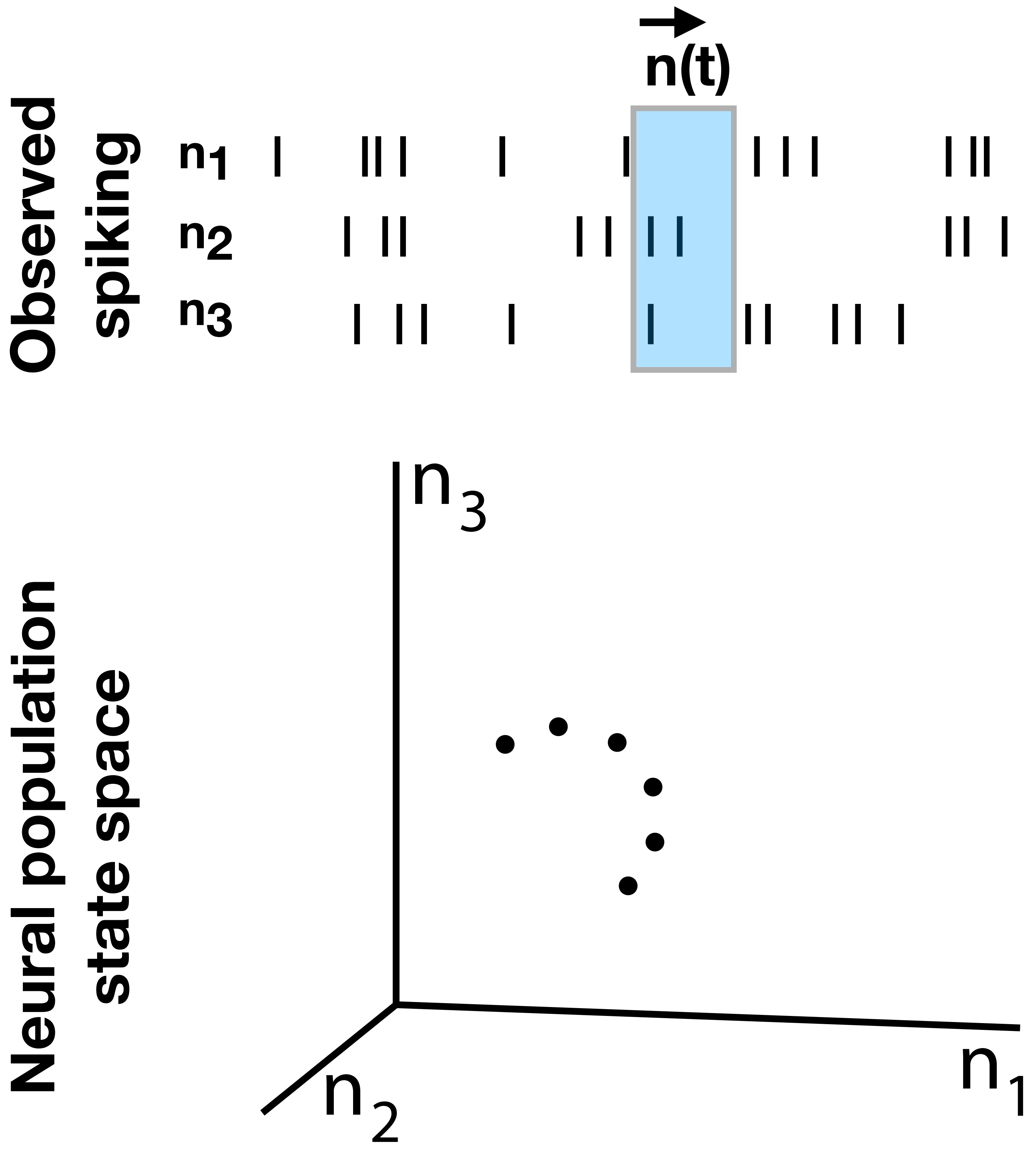
Neural population state spaces



Adapted from Cunningham & Yu, *Nat Neuro* 2014
Dimensionality reduction for large-scale neural recordings

- The current state of the system is captured by the firing rates of all of its neurons

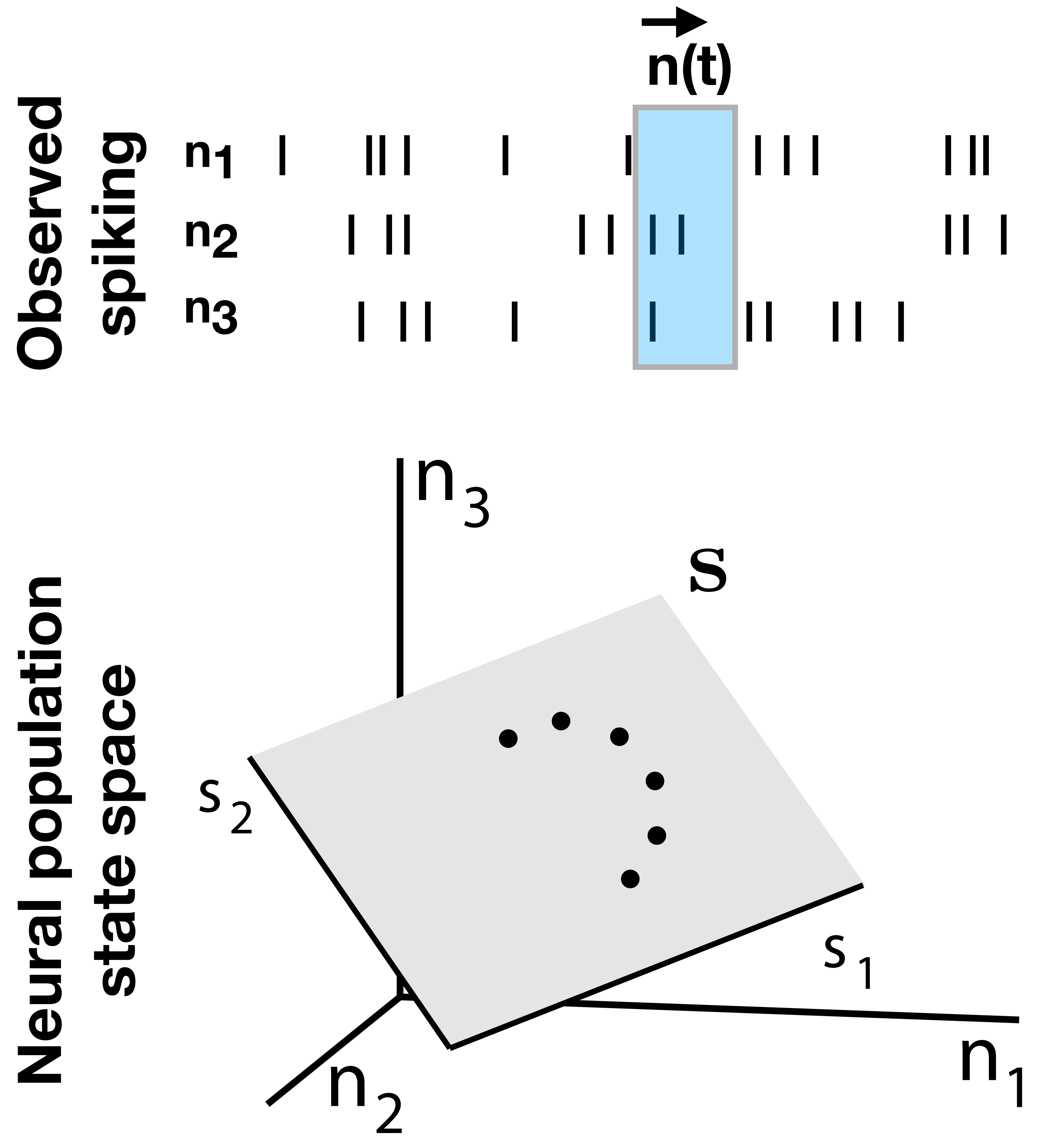
Neural population state spaces



Adapted from Cunningham & Yu, *Nat Neuro* 2014
Dimensionality reduction for large-scale neural recordings

- The current state of the system is captured by the firing rates of all of its neurons
- Activity does not uniformly fill the space - i.e., not all patterns of neural firing rates are equally likely

Neural population state spaces

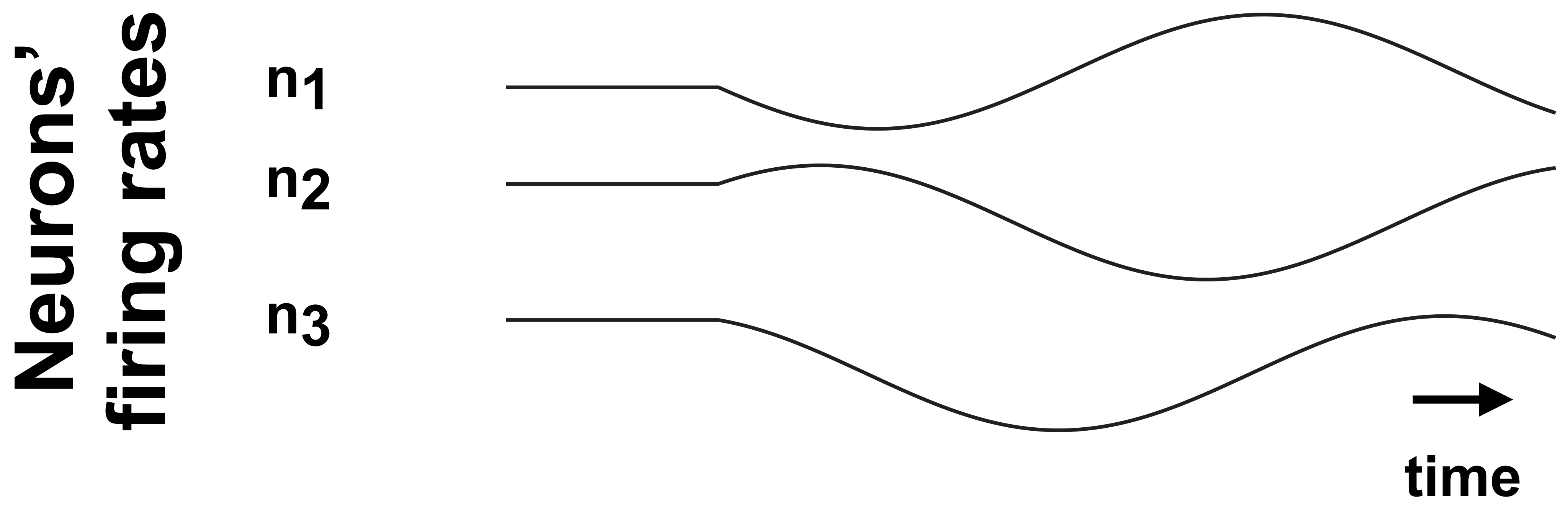


Adapted from Cunningham & Yu, *Nat Neuro* 2014
Dimensionality reduction for large-scale neural recordings

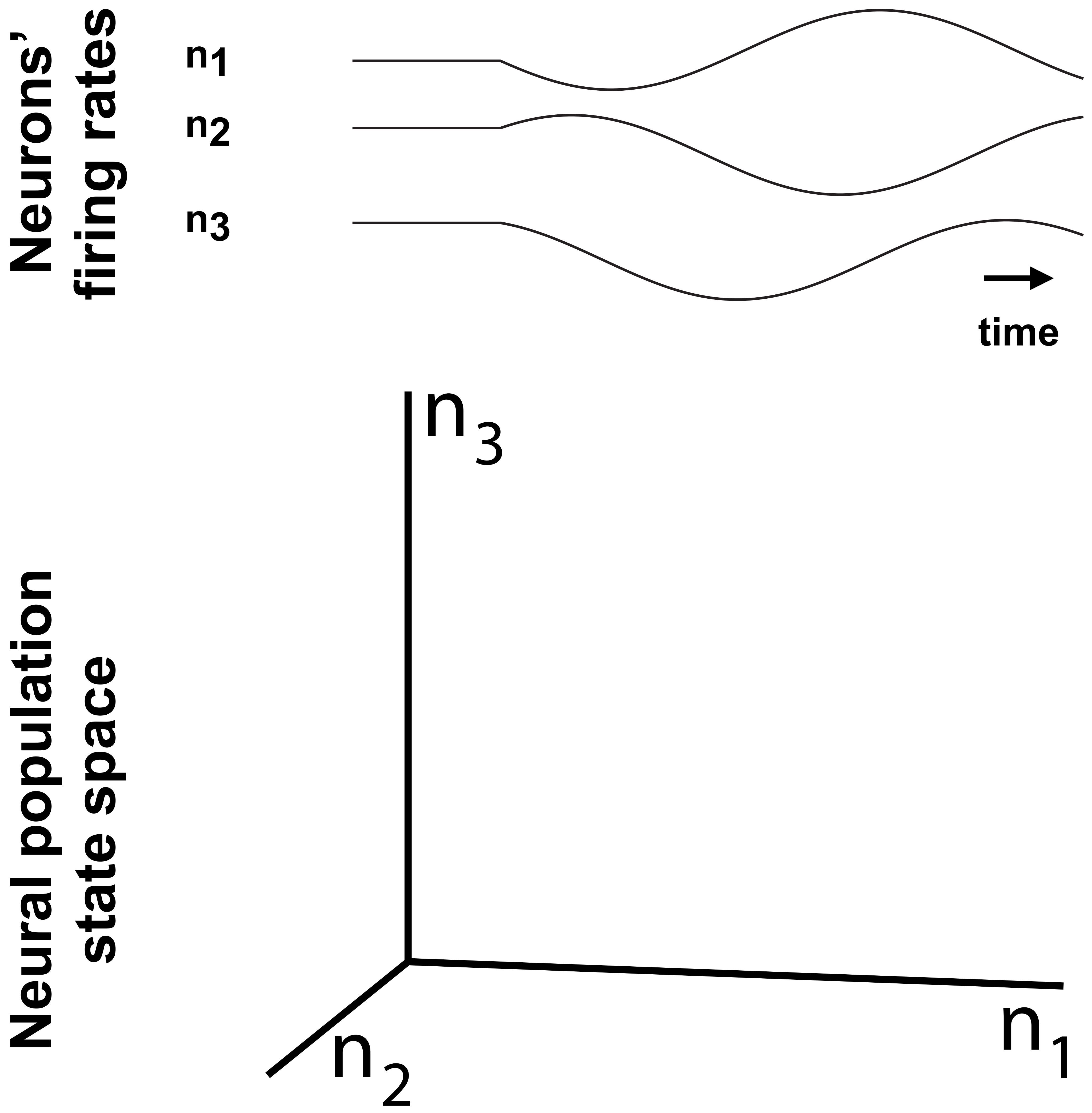
- The current state of the system is captured by the firing rates of all of its neurons
- Activity does not uniformly fill the space - i.e., not all patterns of neural firing rates are equally likely
- We can usually identify shared structure underlying the firing rates of the recorded neurons

Changes in neurons’ firing rates are coordinated

Changes in neurons' firing rates are coordinated

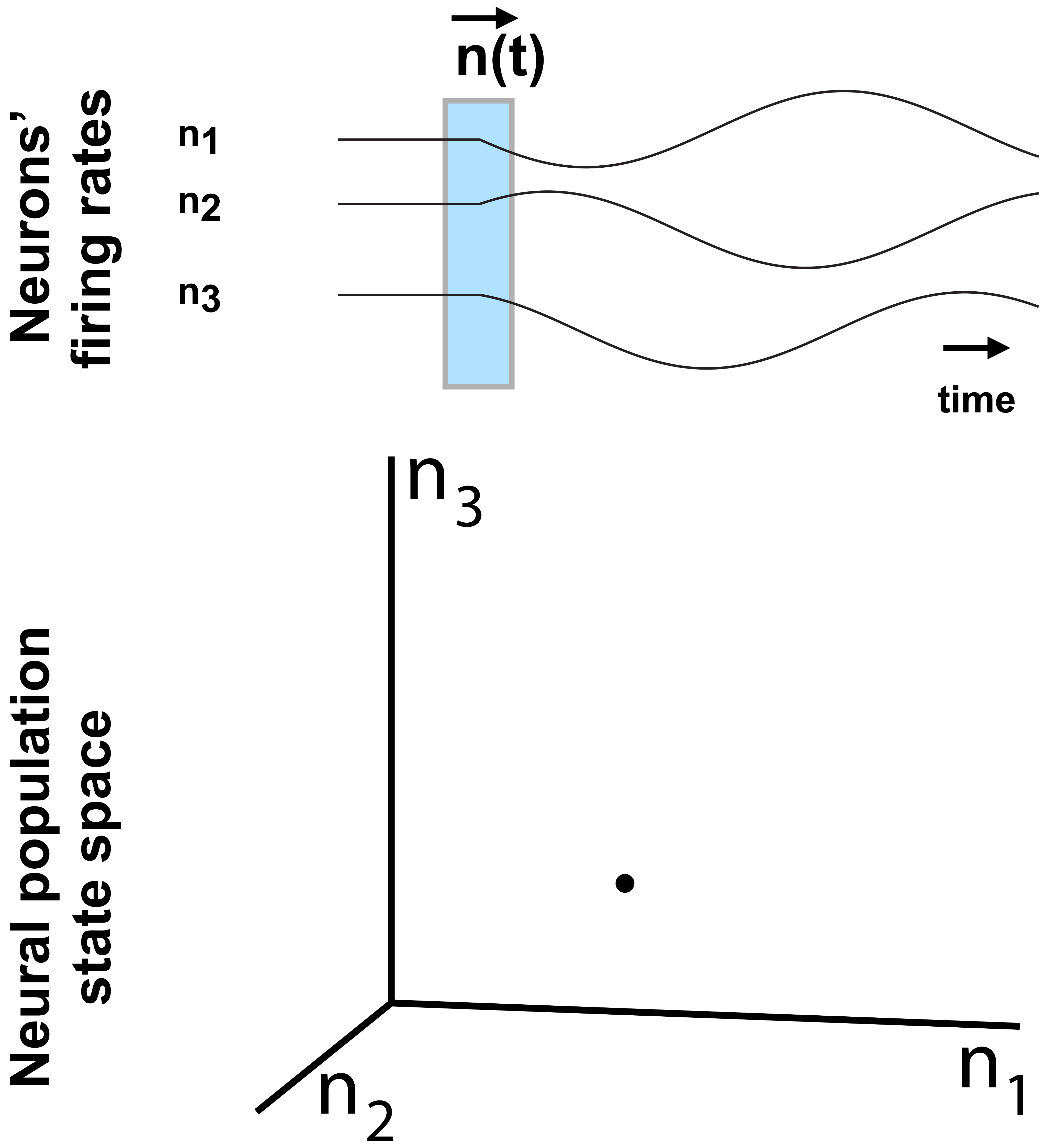


Changes in neurons’ firing rates are coordinated



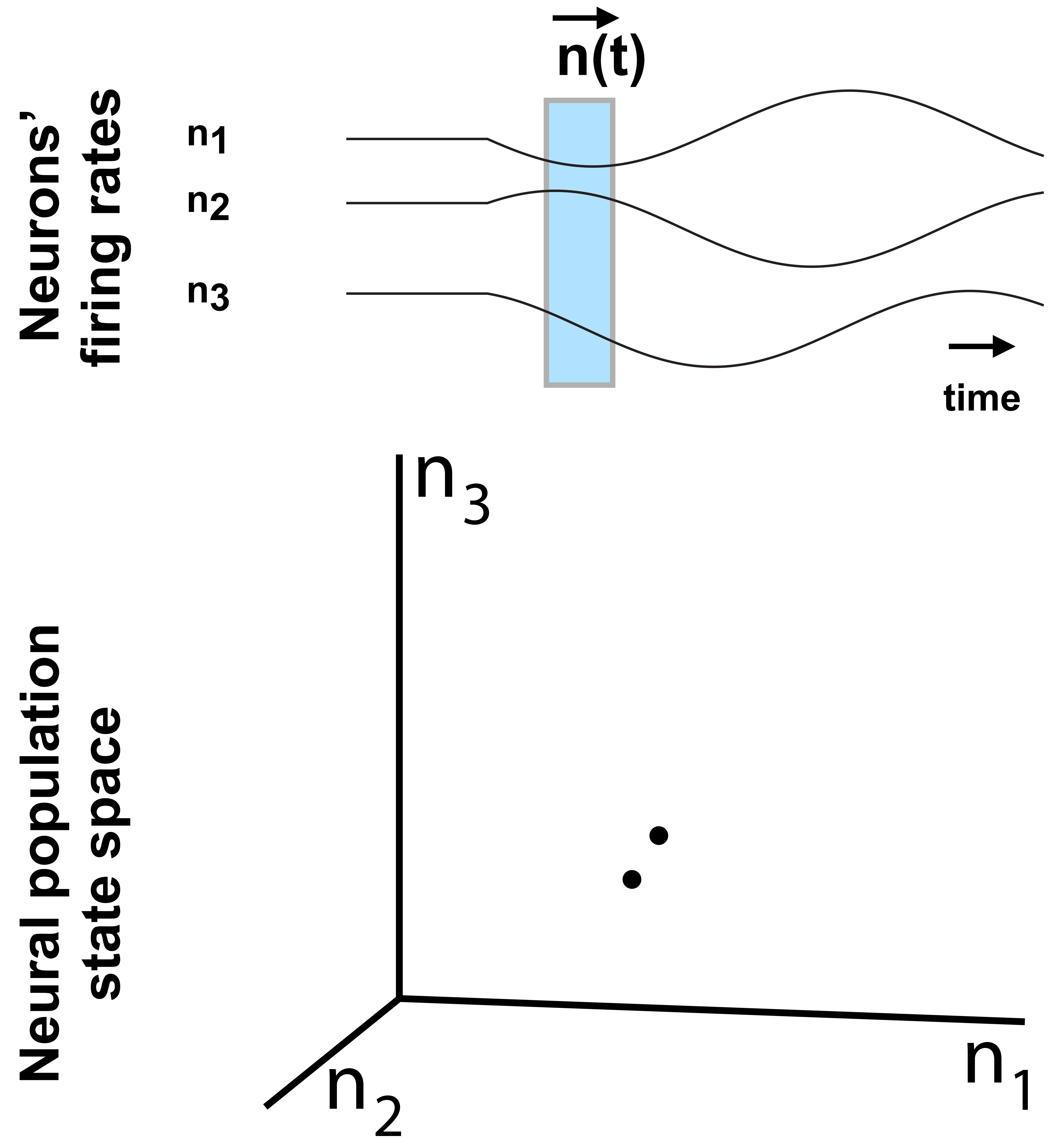
Adapted from Cunningham & Yu, *Nat Neuro* 2014

Changes in neurons' firing rates are coordinated



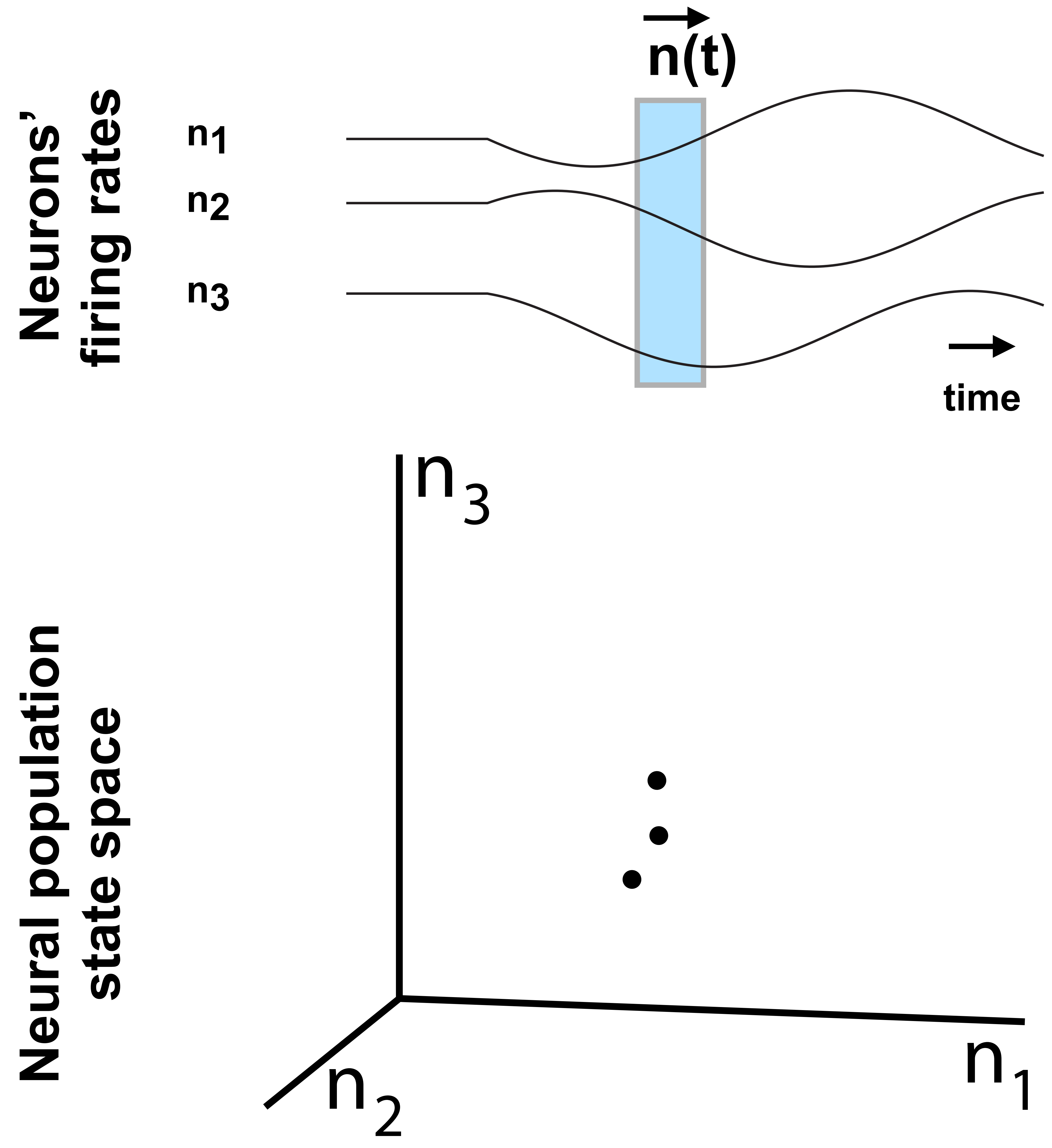
Adapted from Cunningham & Yu, *Nat Neuro* 2014

Changes in neurons’ firing rates are coordinated



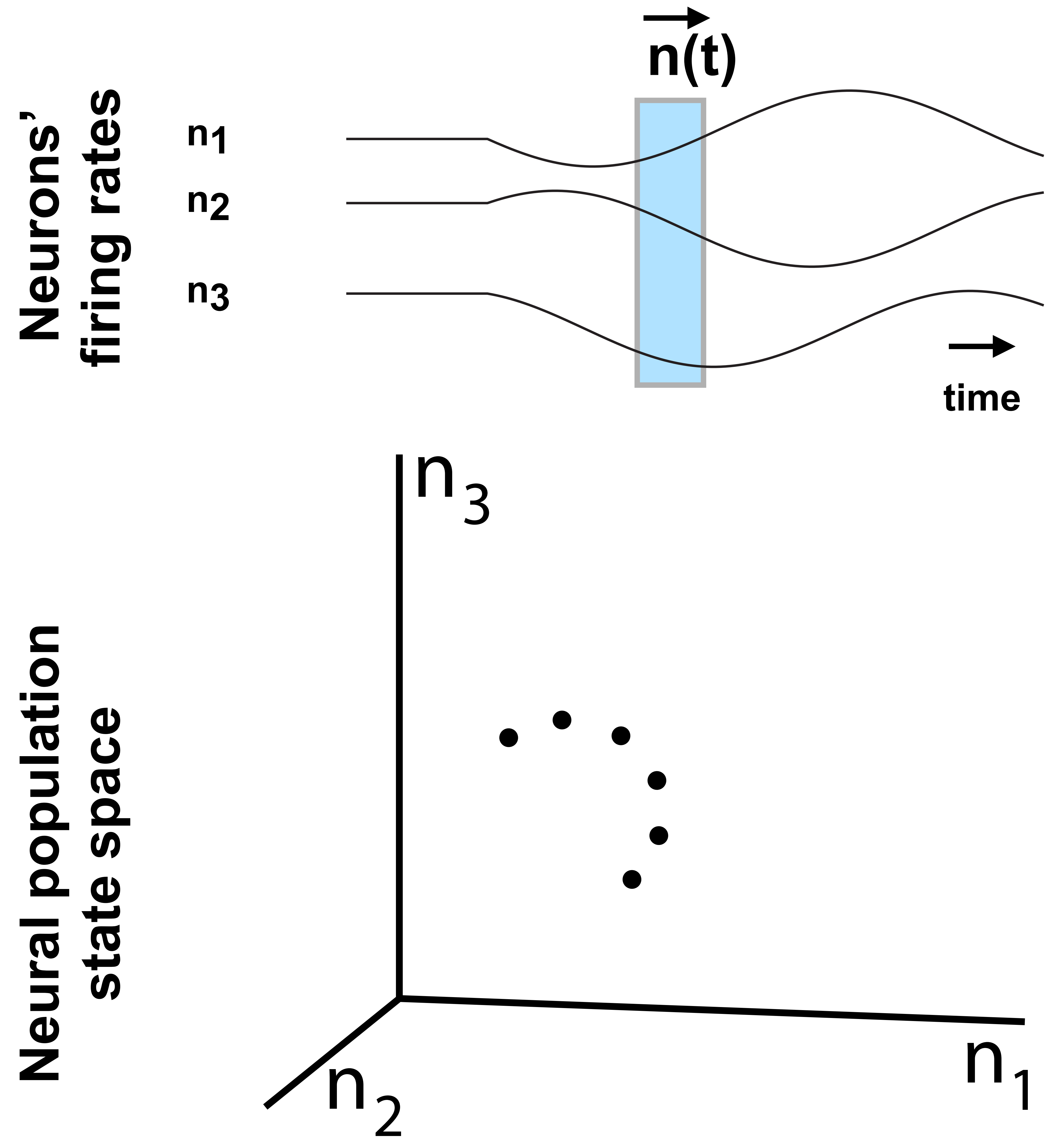
Adapted from Cunningham & Yu, *Nat Neuro* 2014

Changes in neurons' firing rates are coordinated



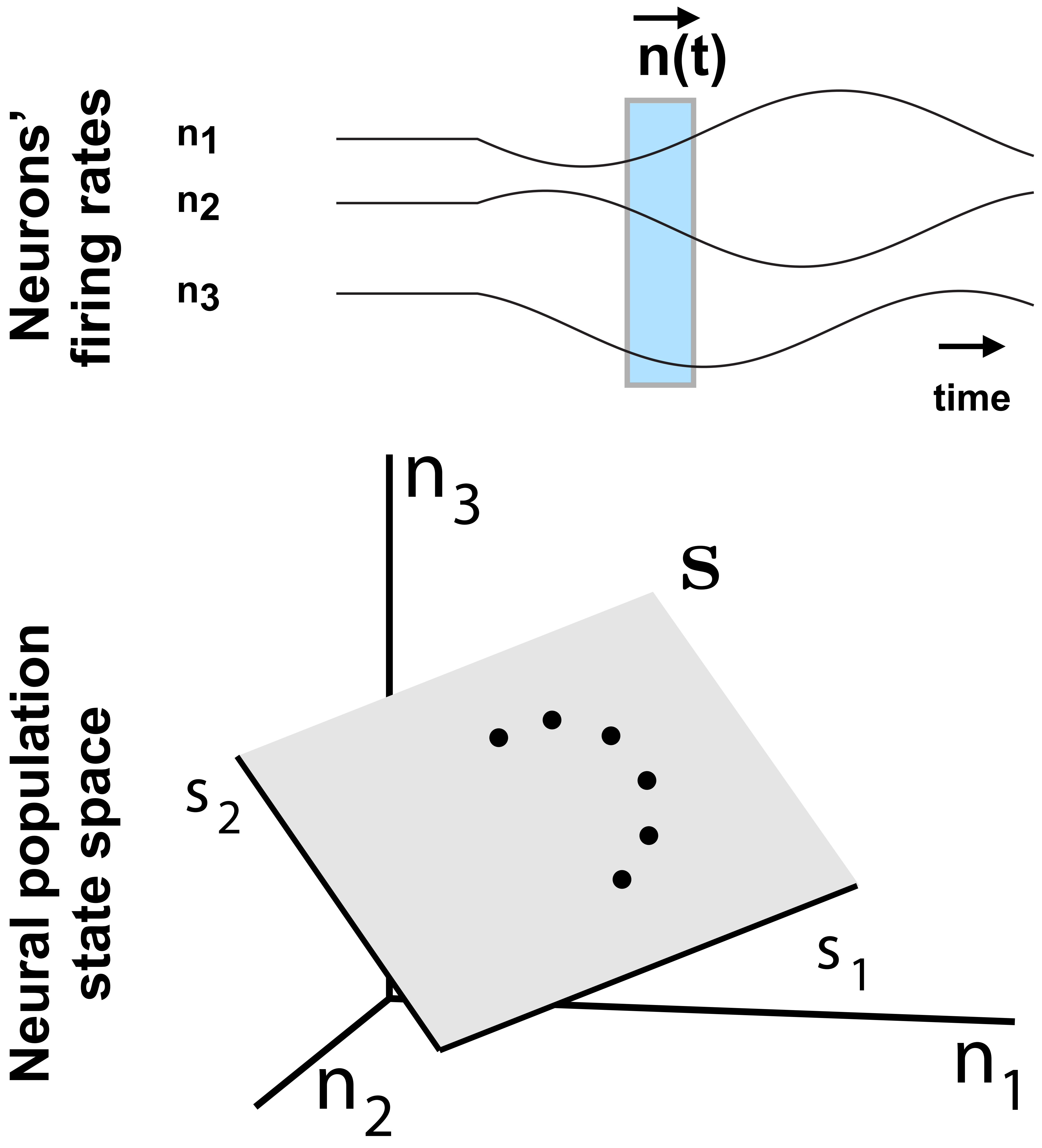
Adapted from Cunningham & Yu, *Nat Neuro* 2014

Changes in neurons' firing rates are coordinated



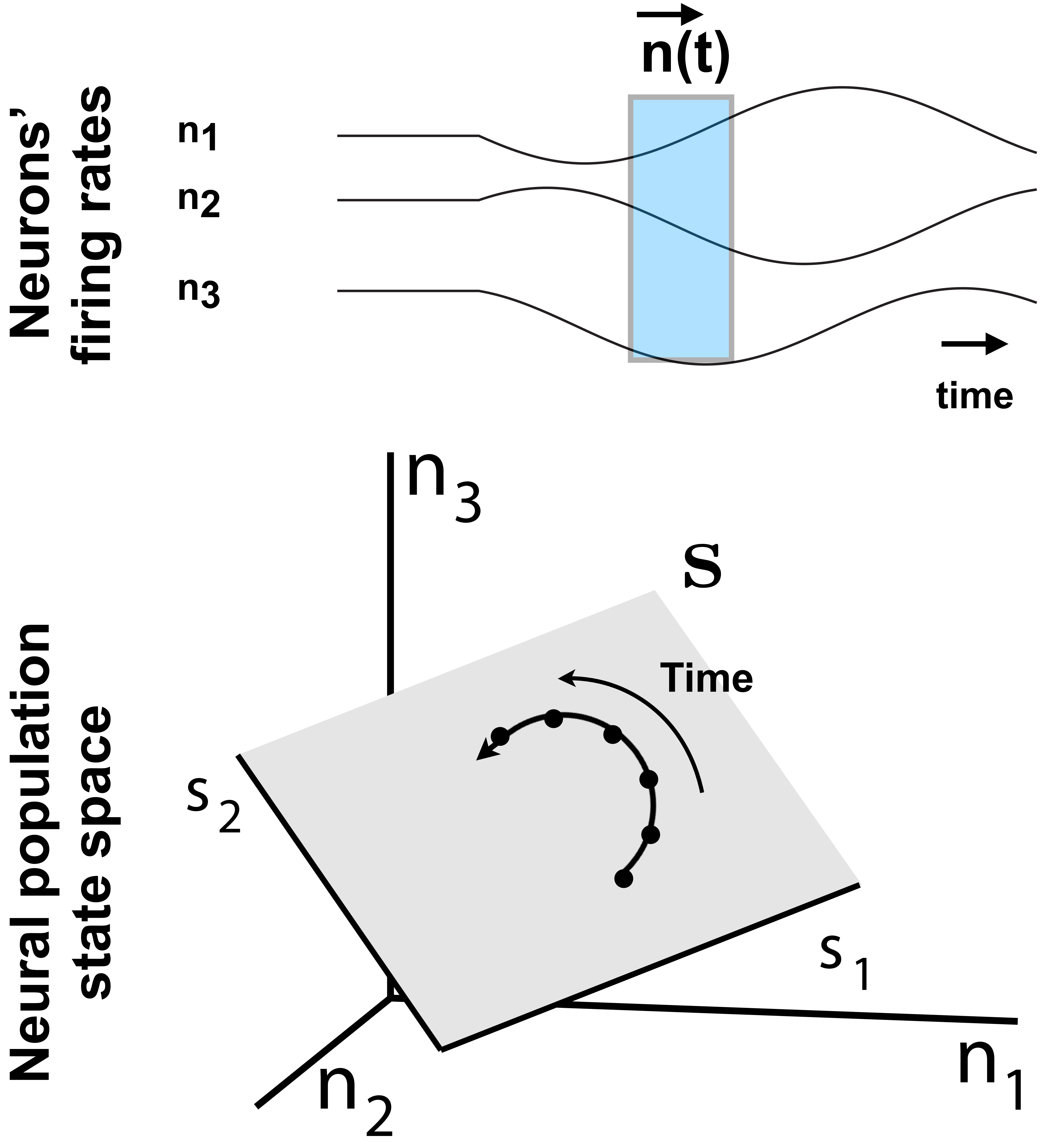
Adapted from Cunningham & Yu, *Nat Neuro* 2014

Changes in neurons' firing rates are coordinated

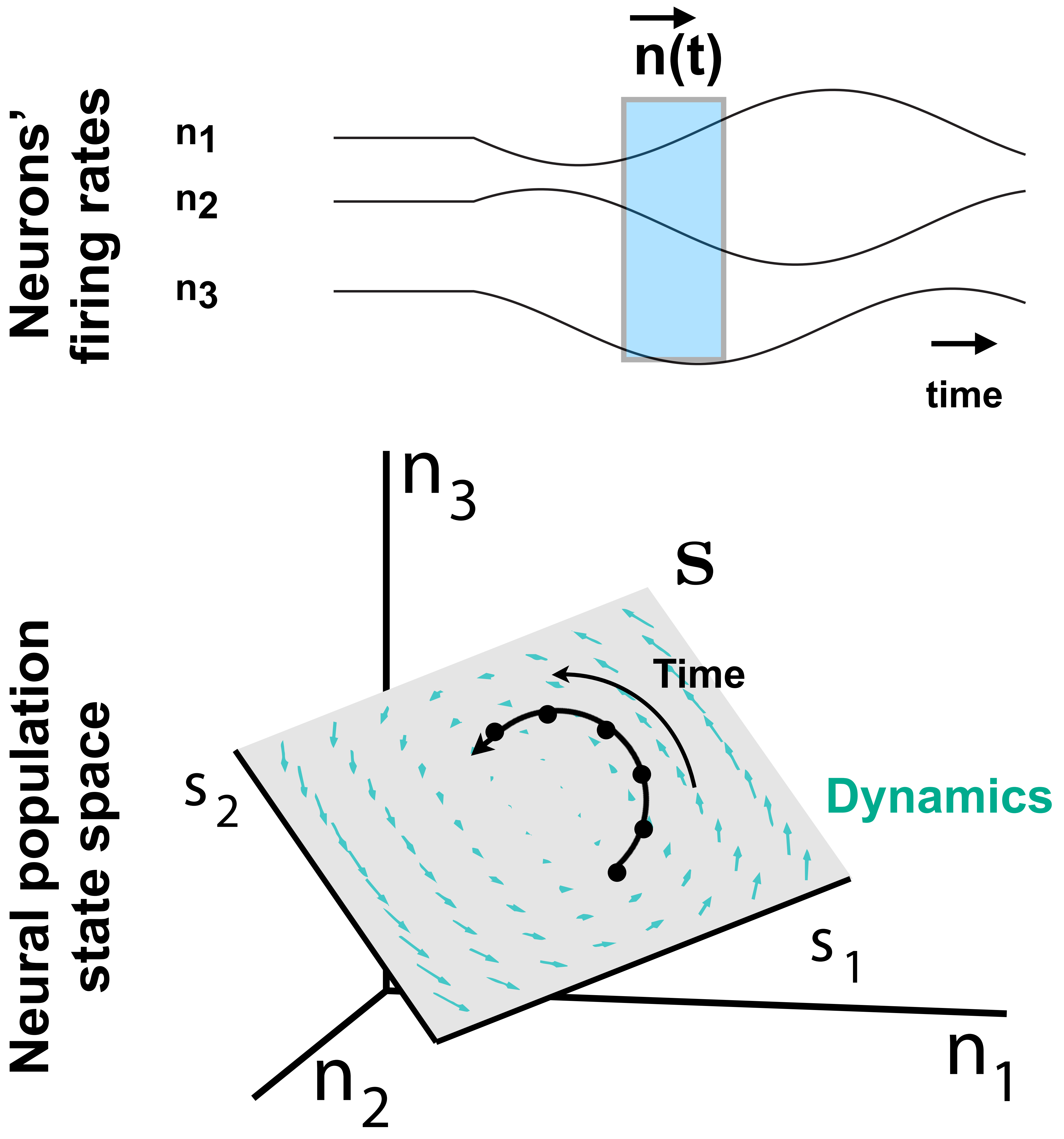


Adapted from Cunningham & Yu, *Nat Neuro* 2014

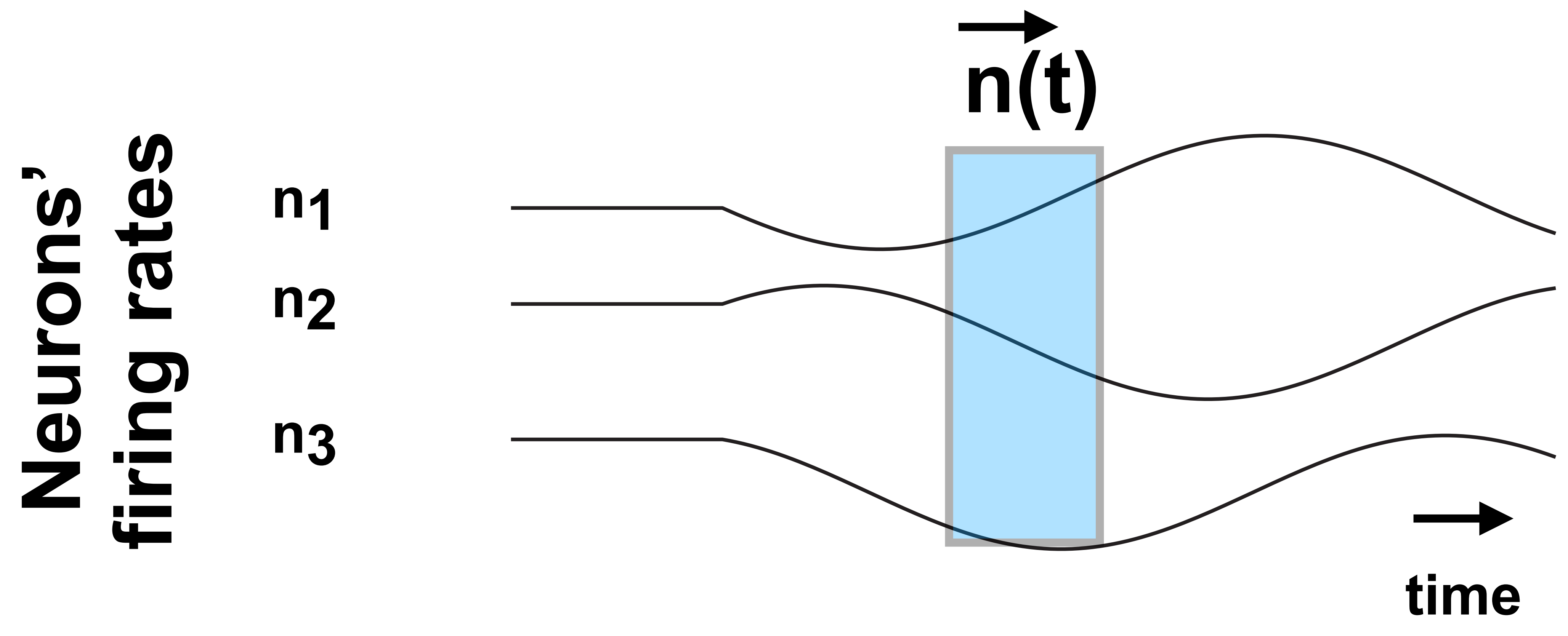
Neural population dynamics



Neural population dynamics

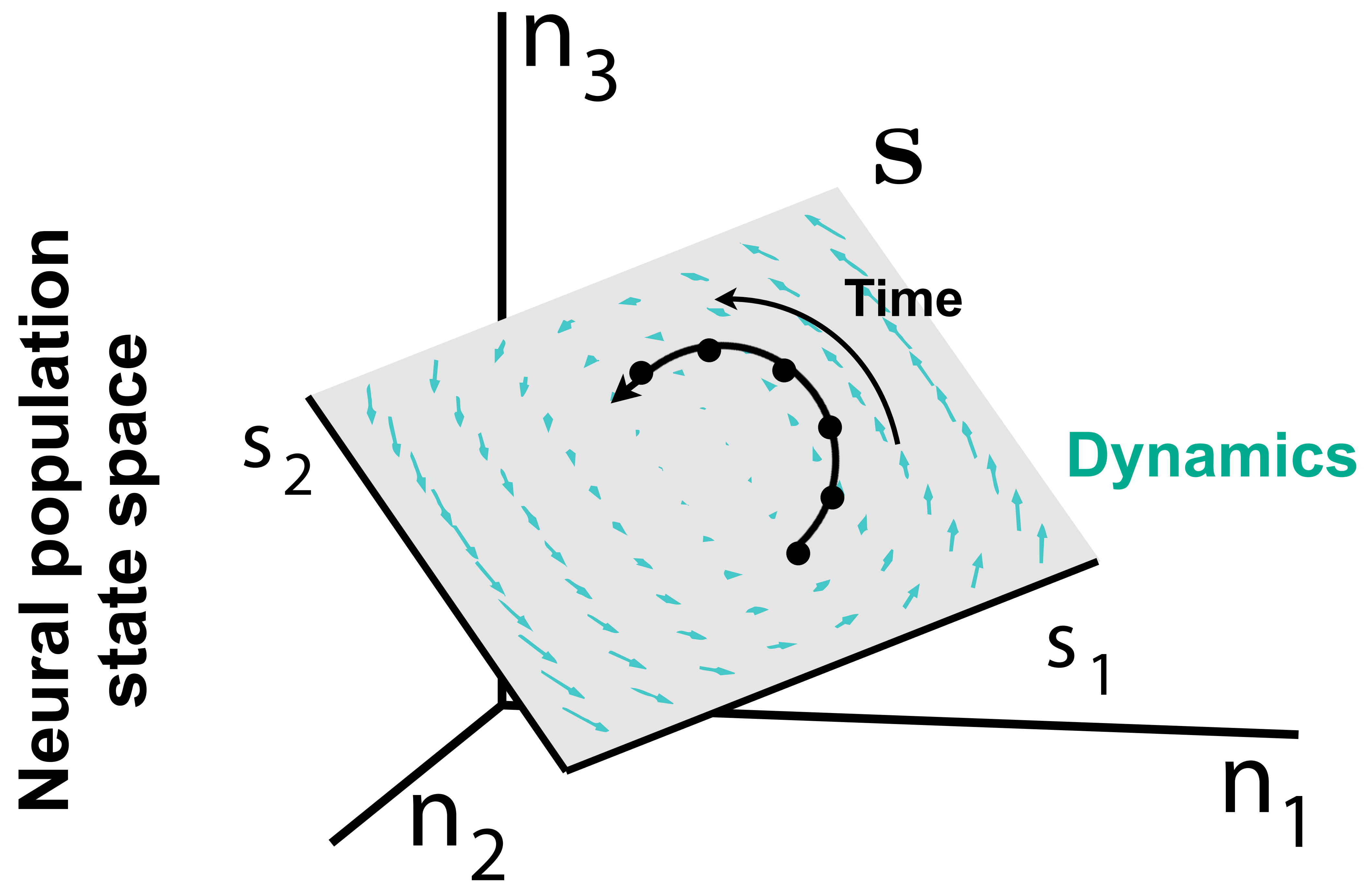


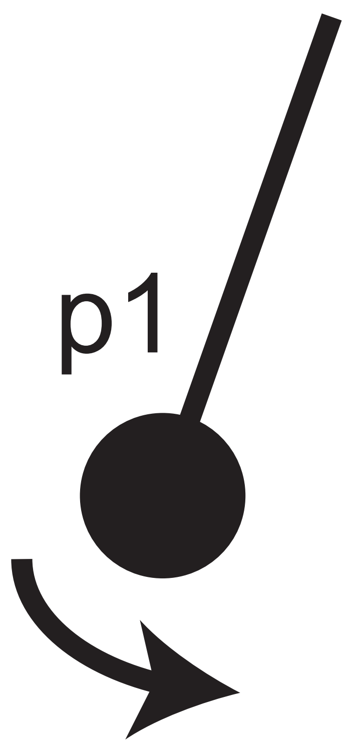
Neural population dynamics

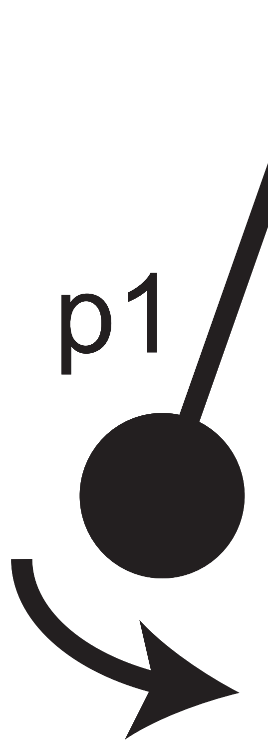


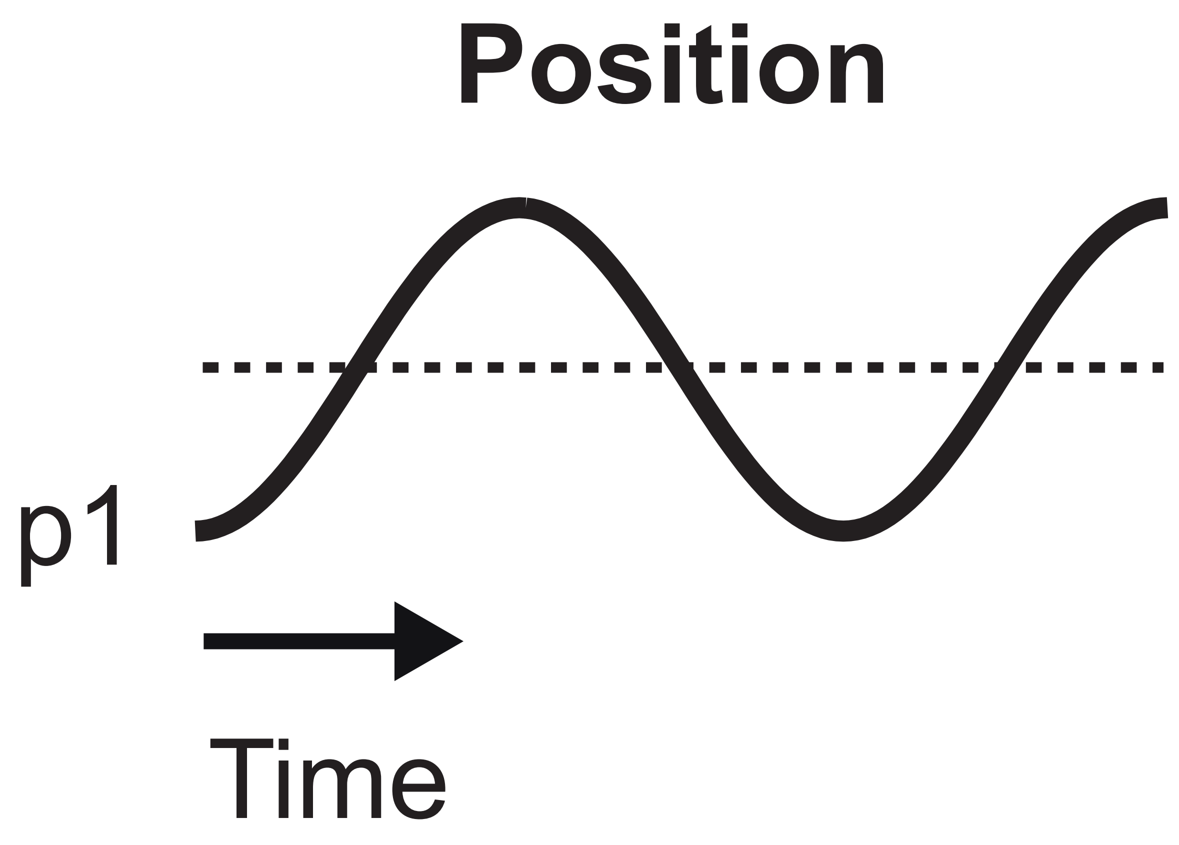
Predictable activity -
modeled by
autonomous
dynamics

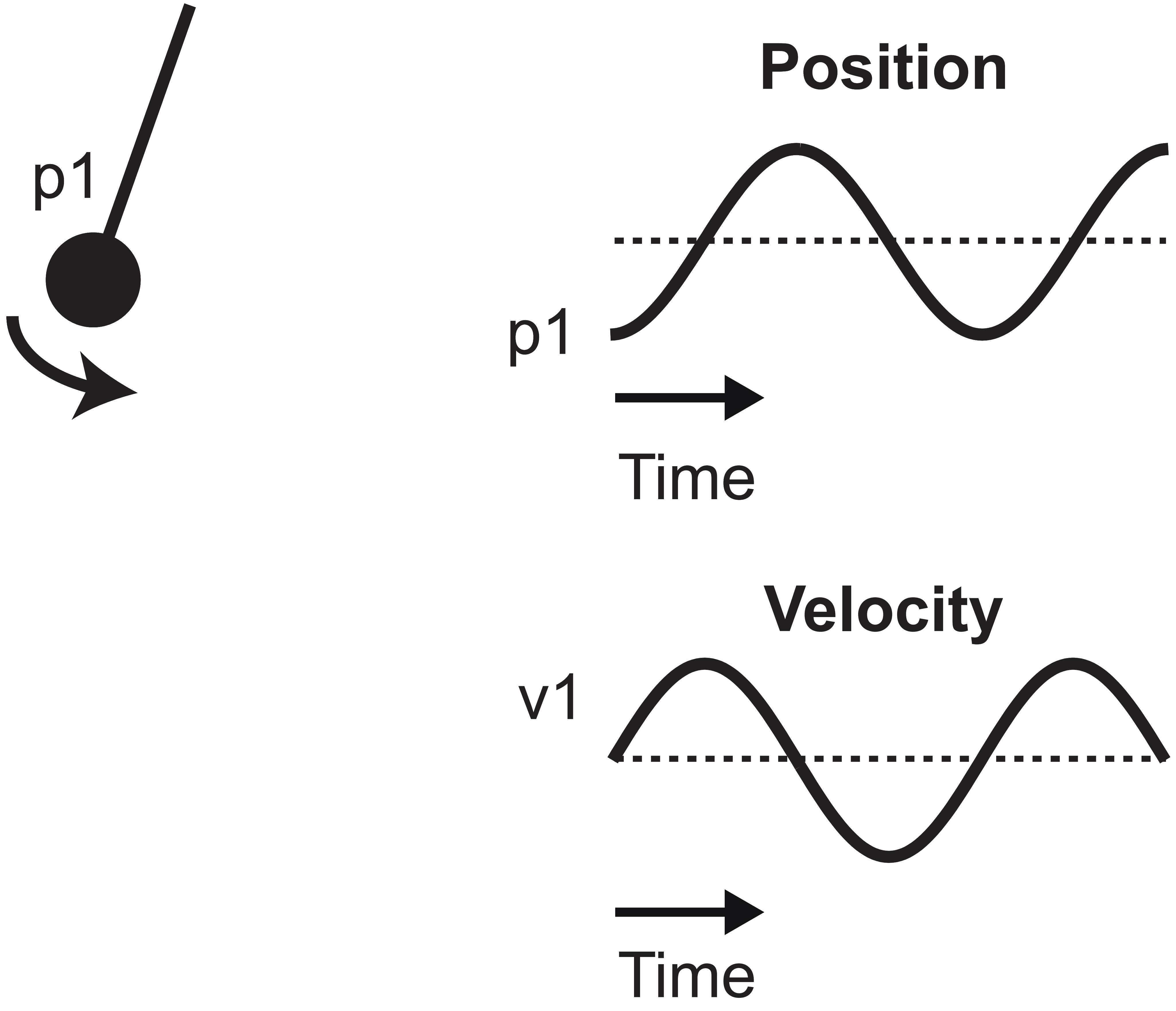
$$\frac{d\mathrm{s}}{dt} = f(\mathrm{s})$$

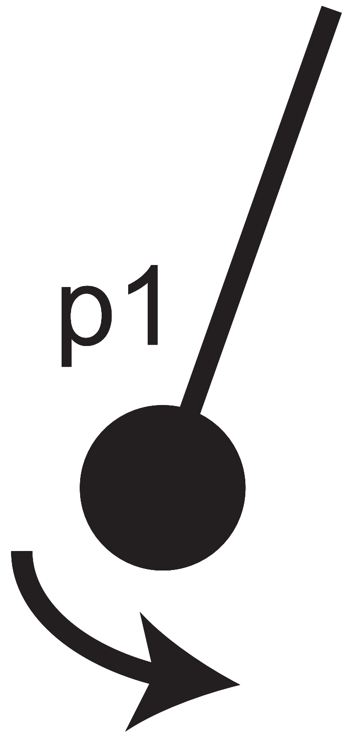


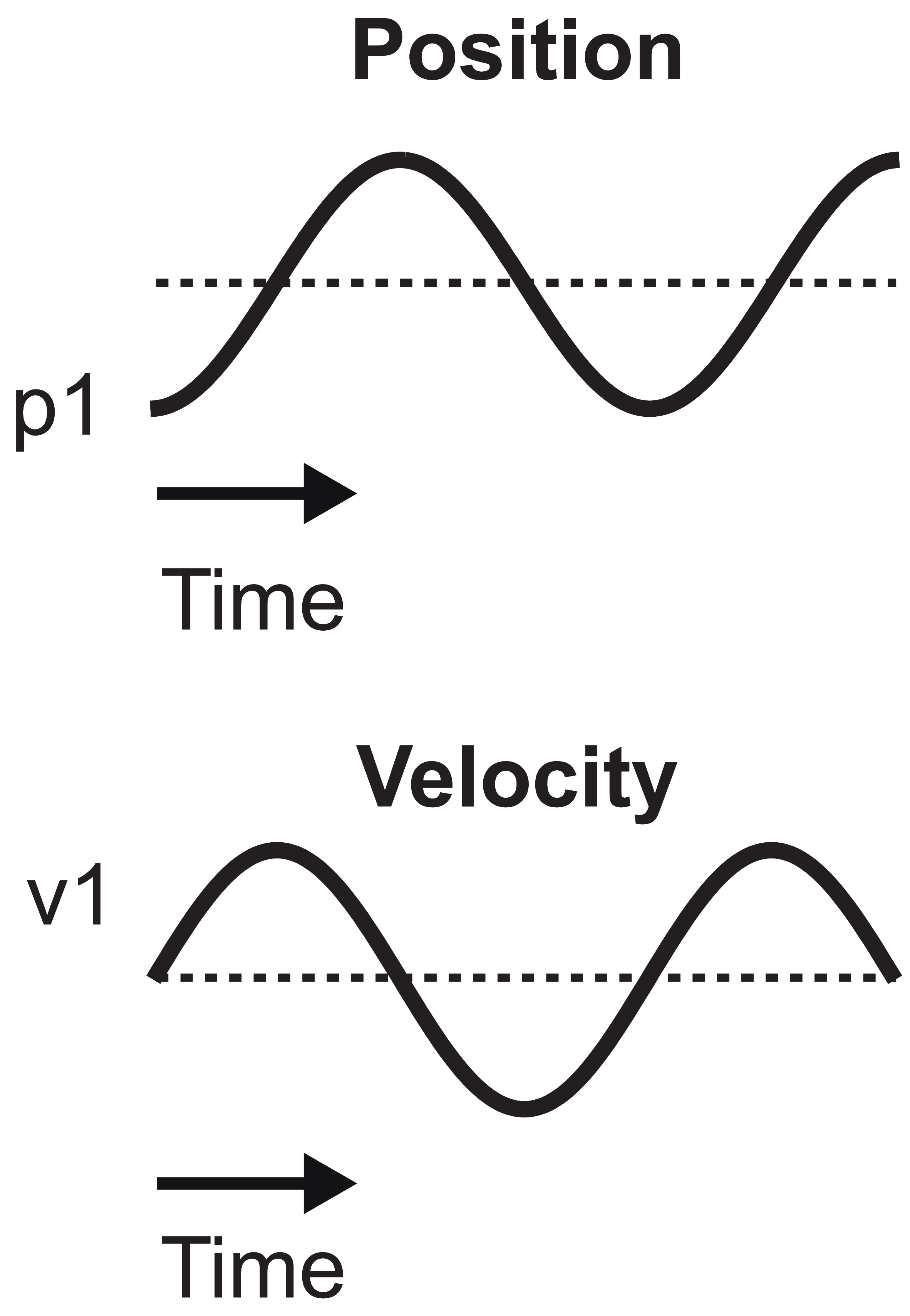

p1


p1

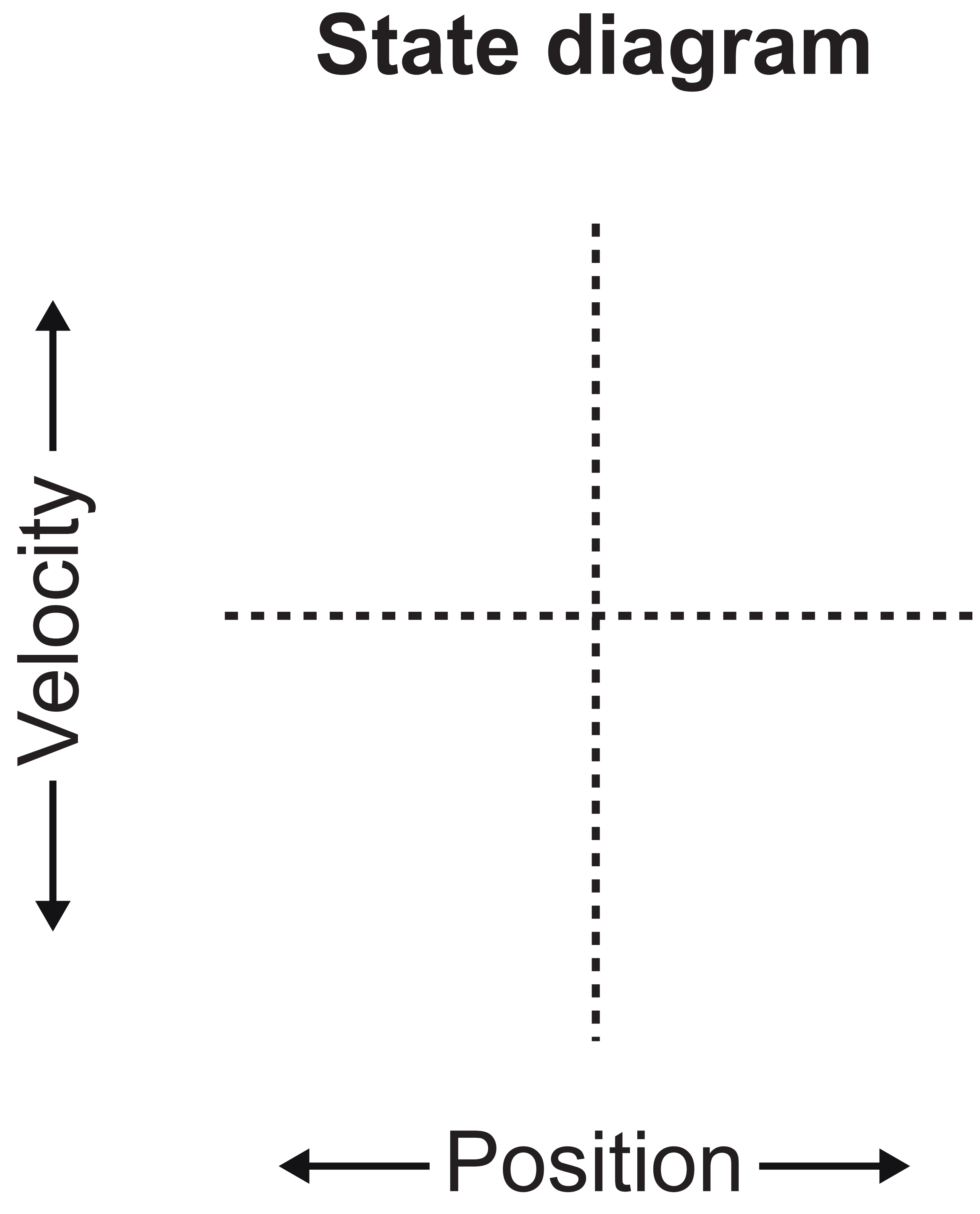

Position
p1
Time

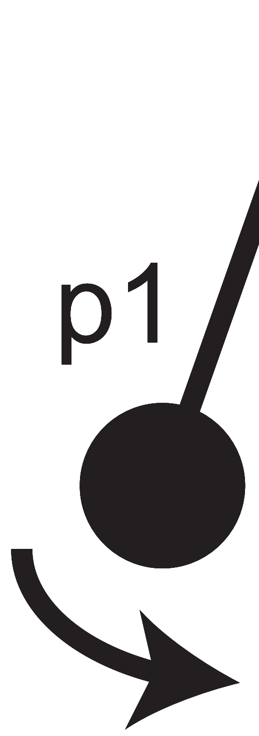

p1
Position
p1
Time
Velocity
v1
Time

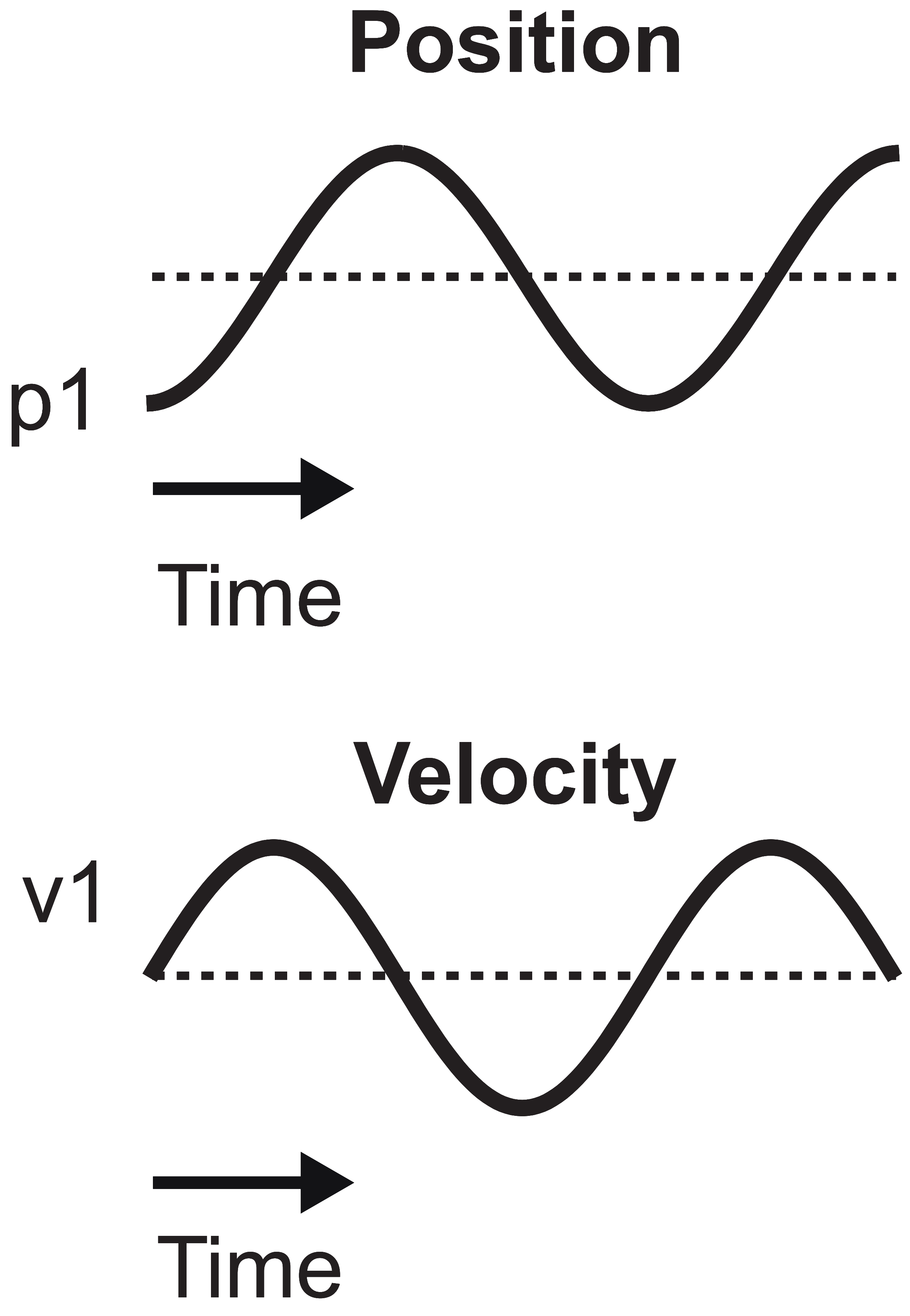




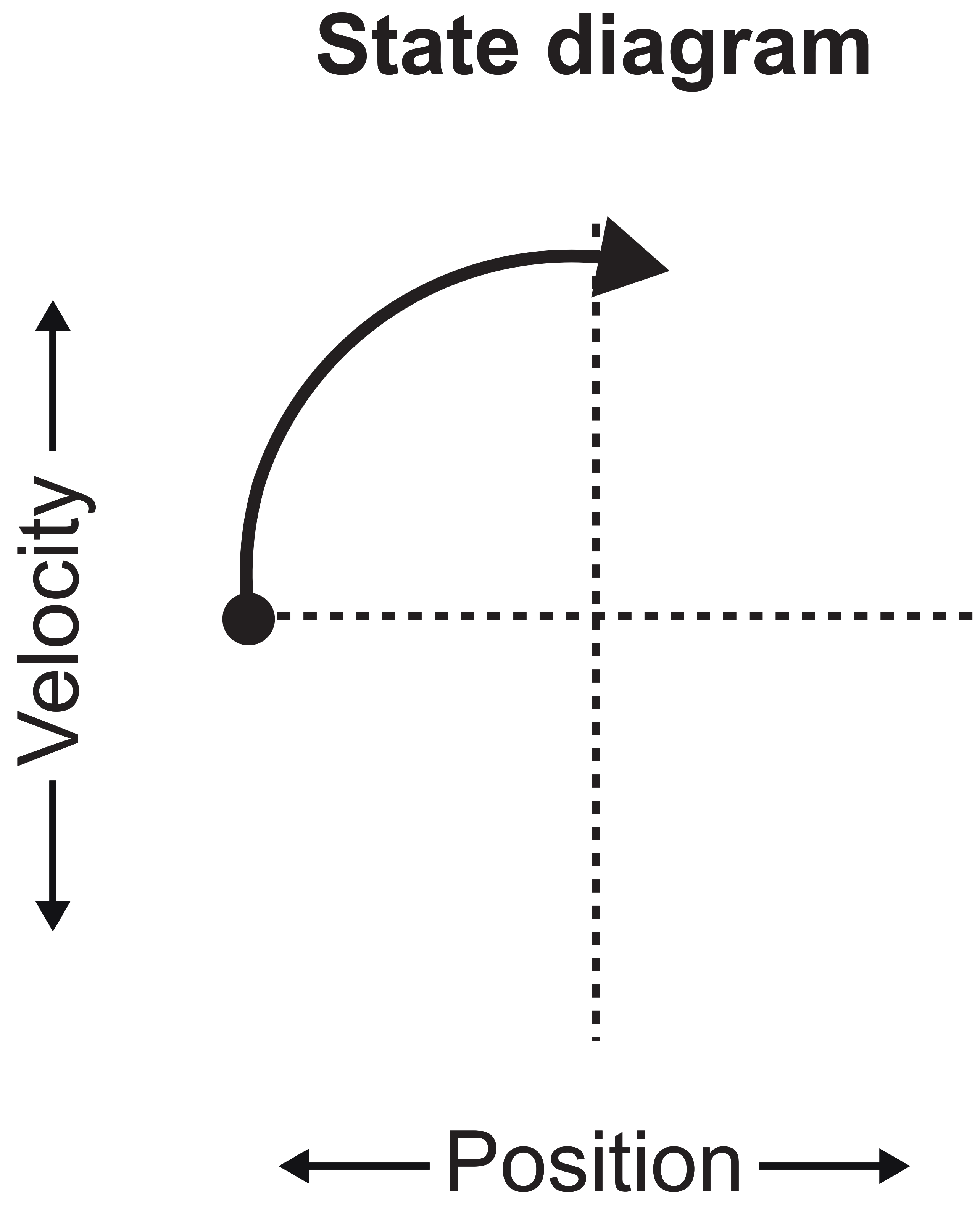
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

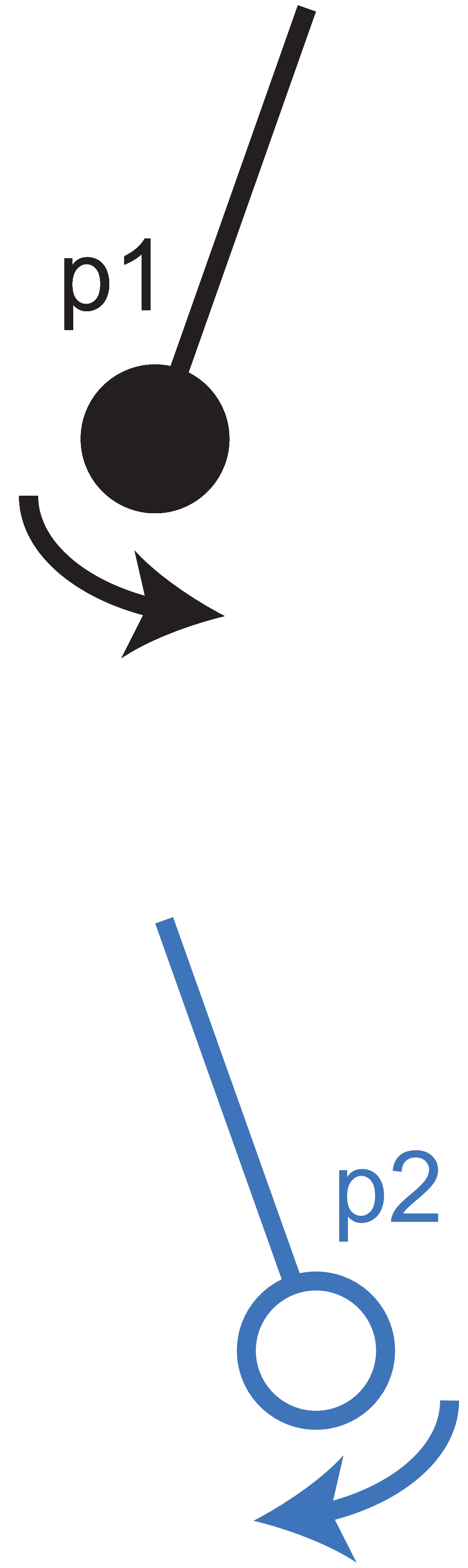


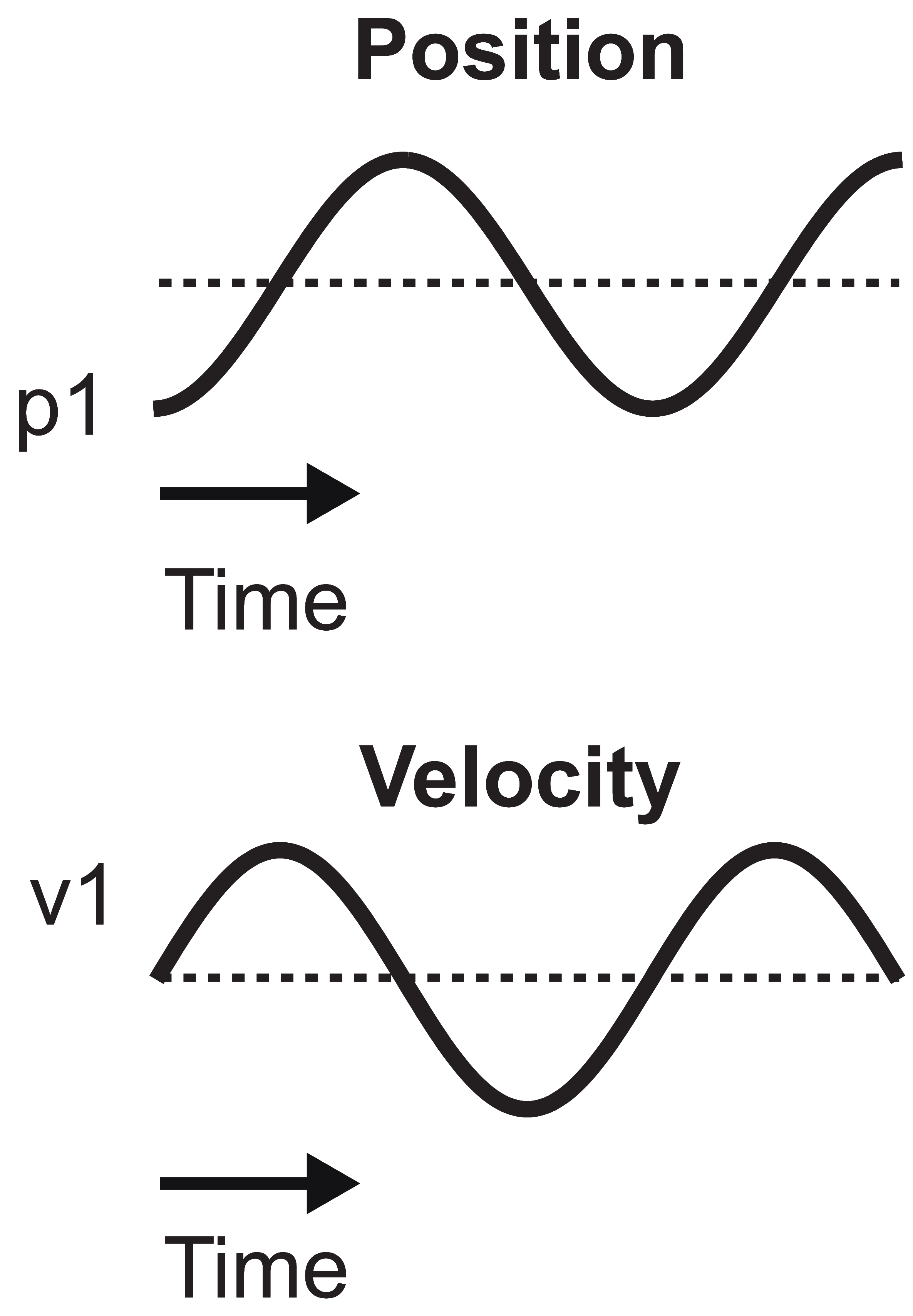




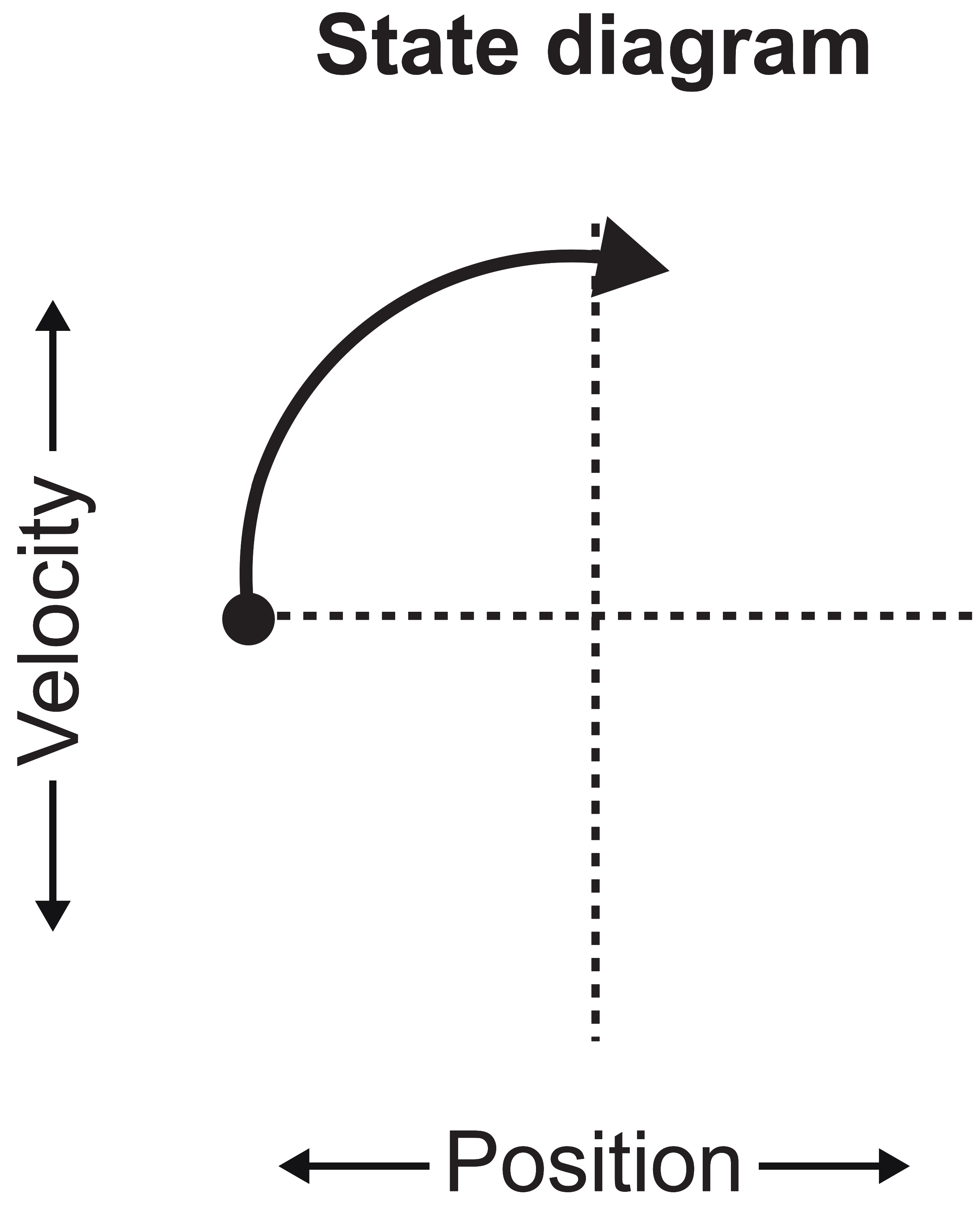
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

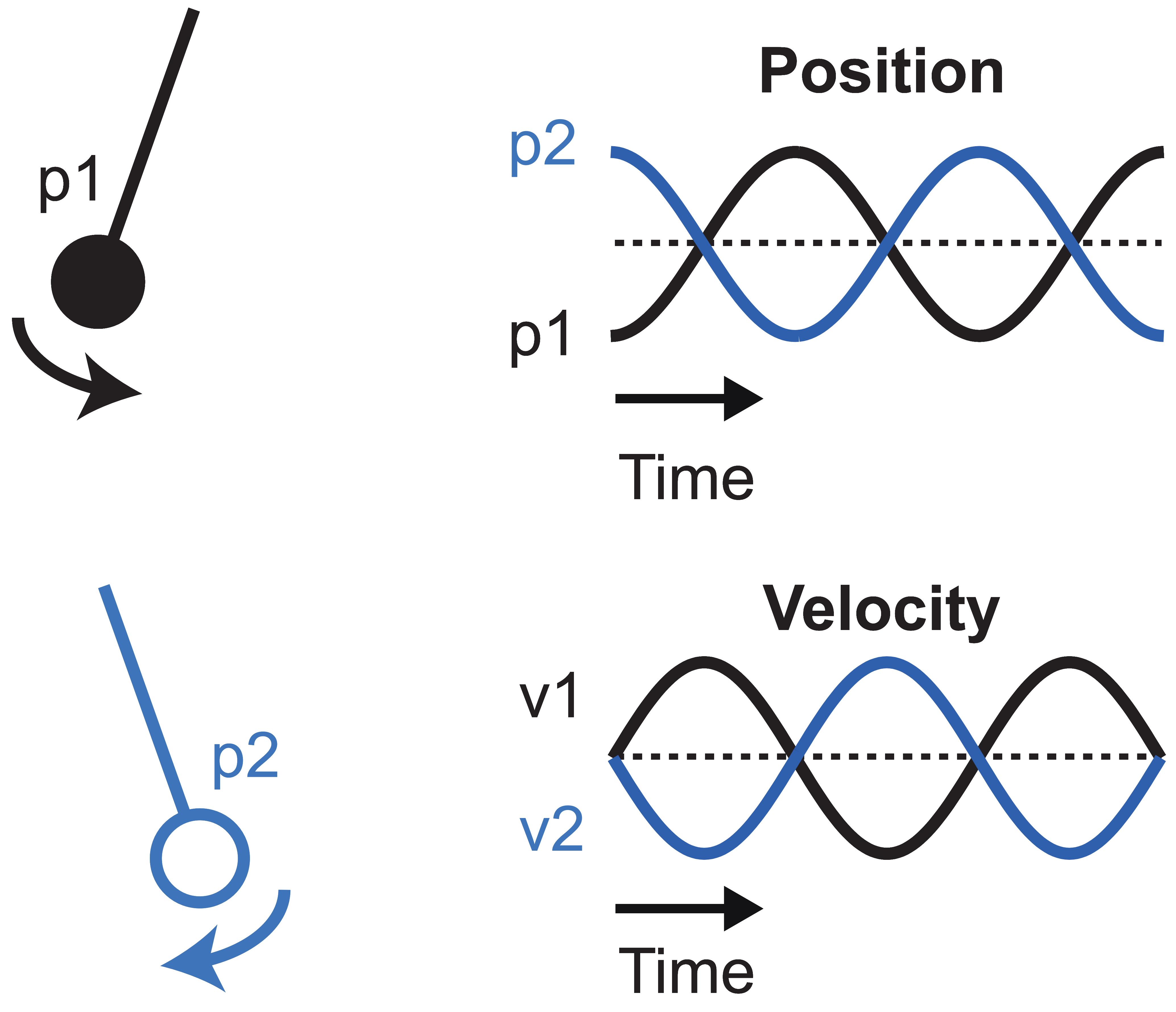




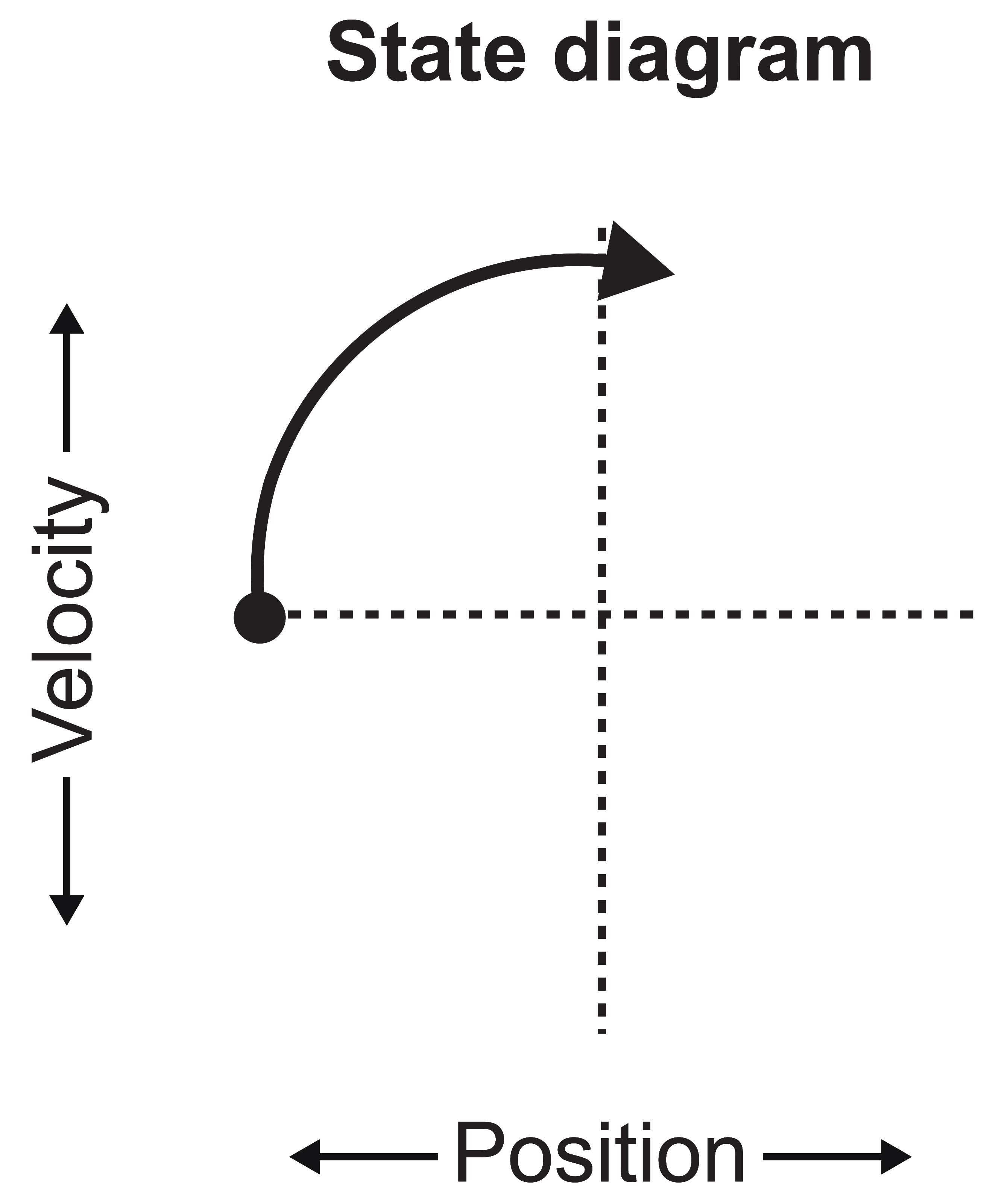


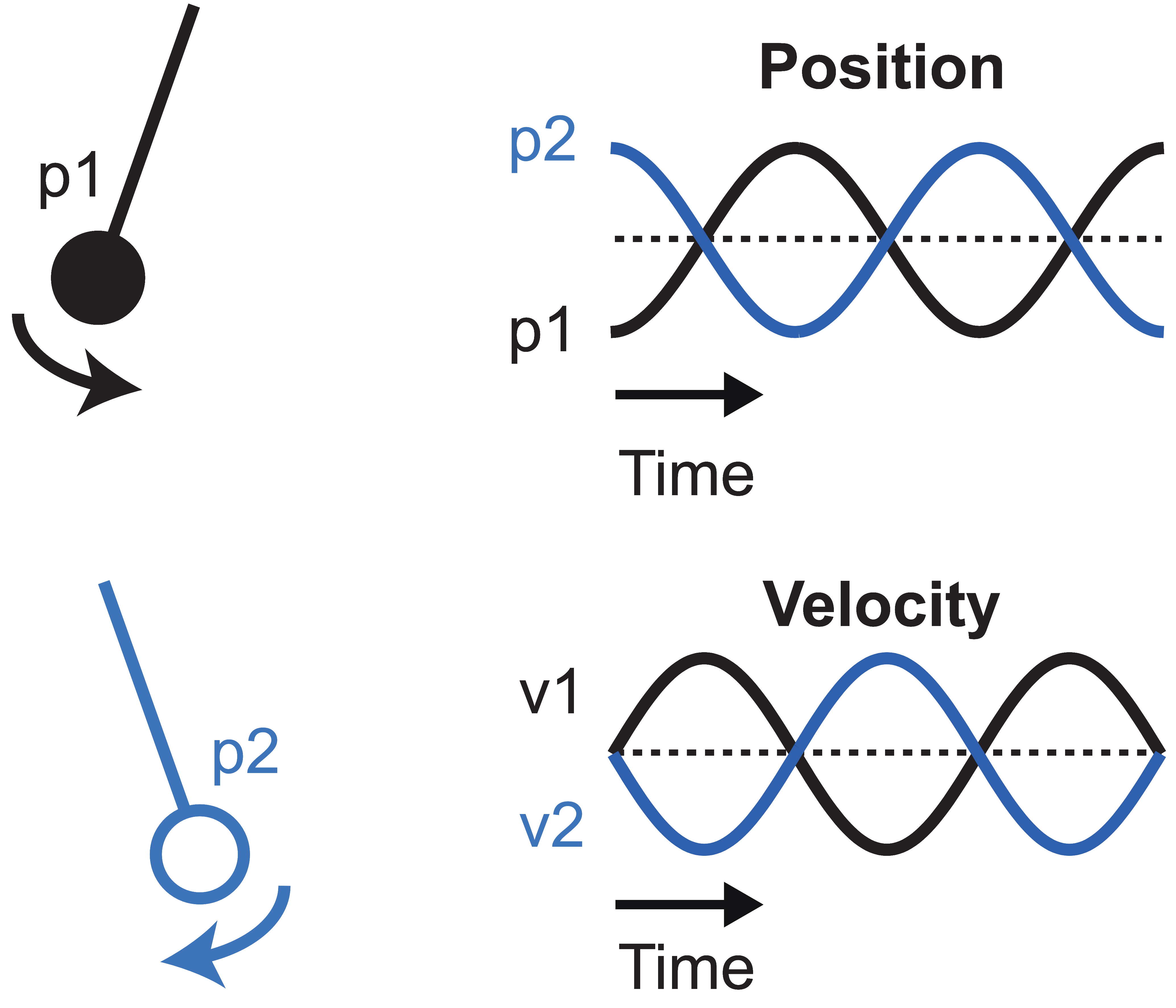
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$



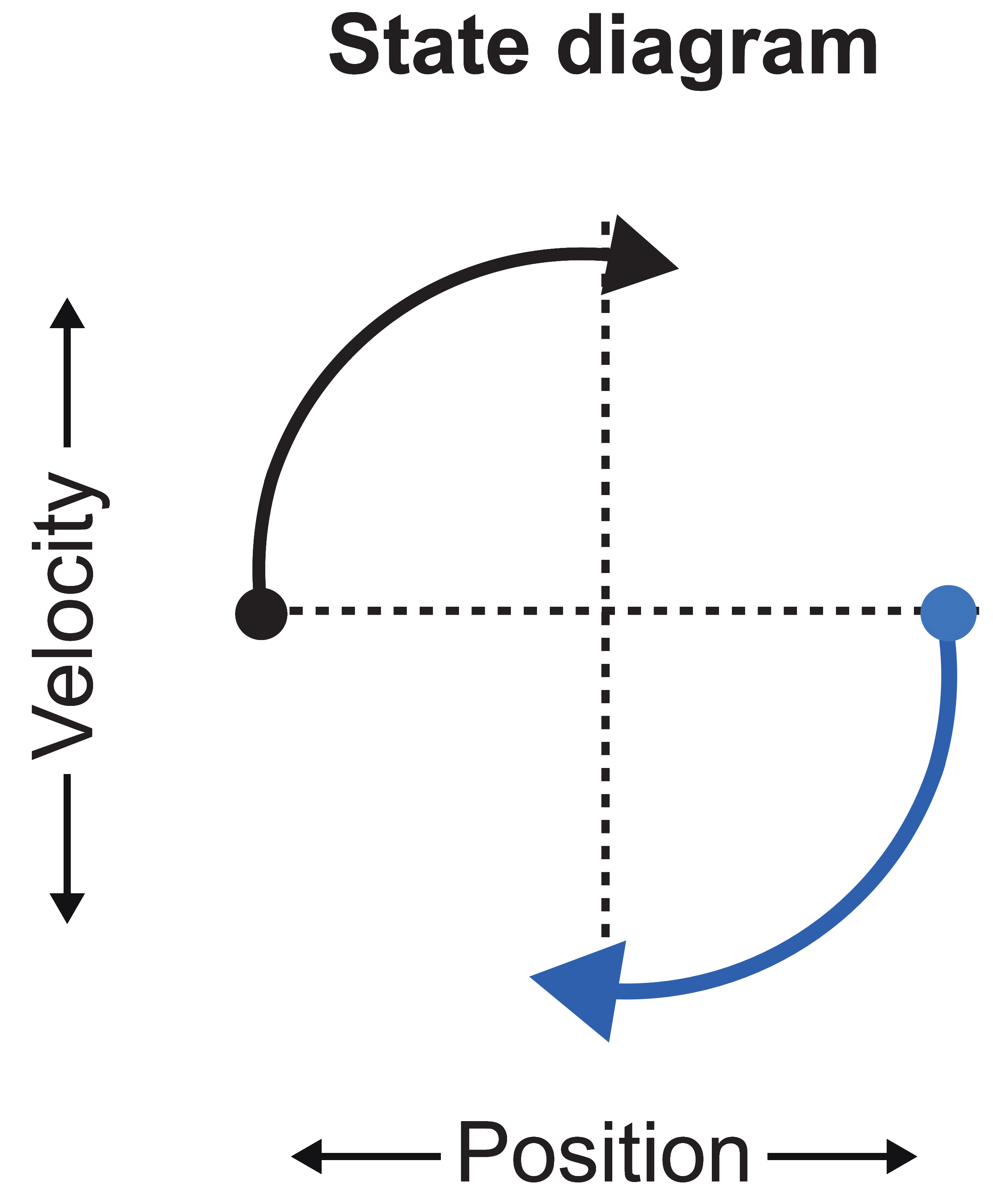


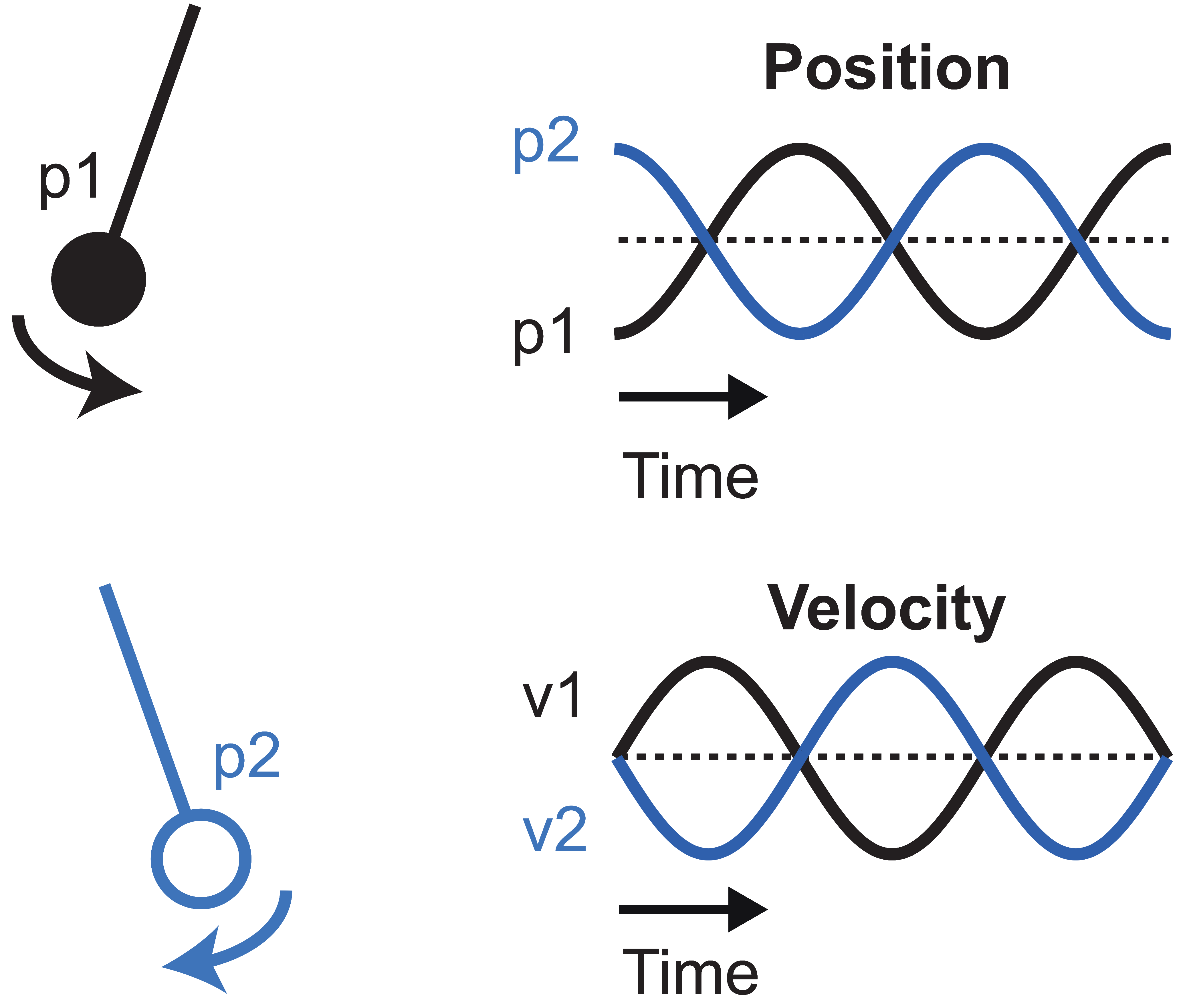
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$





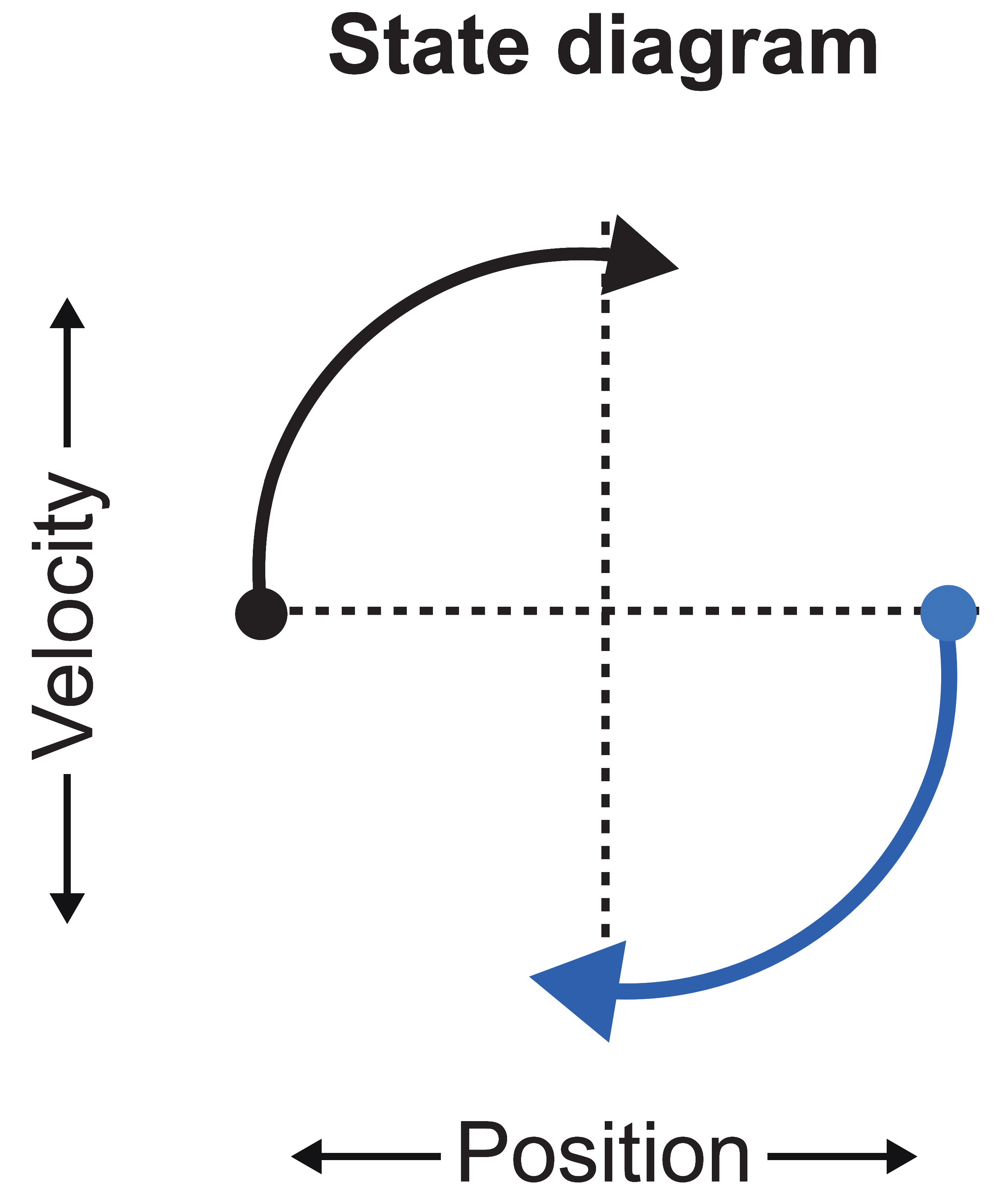
$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

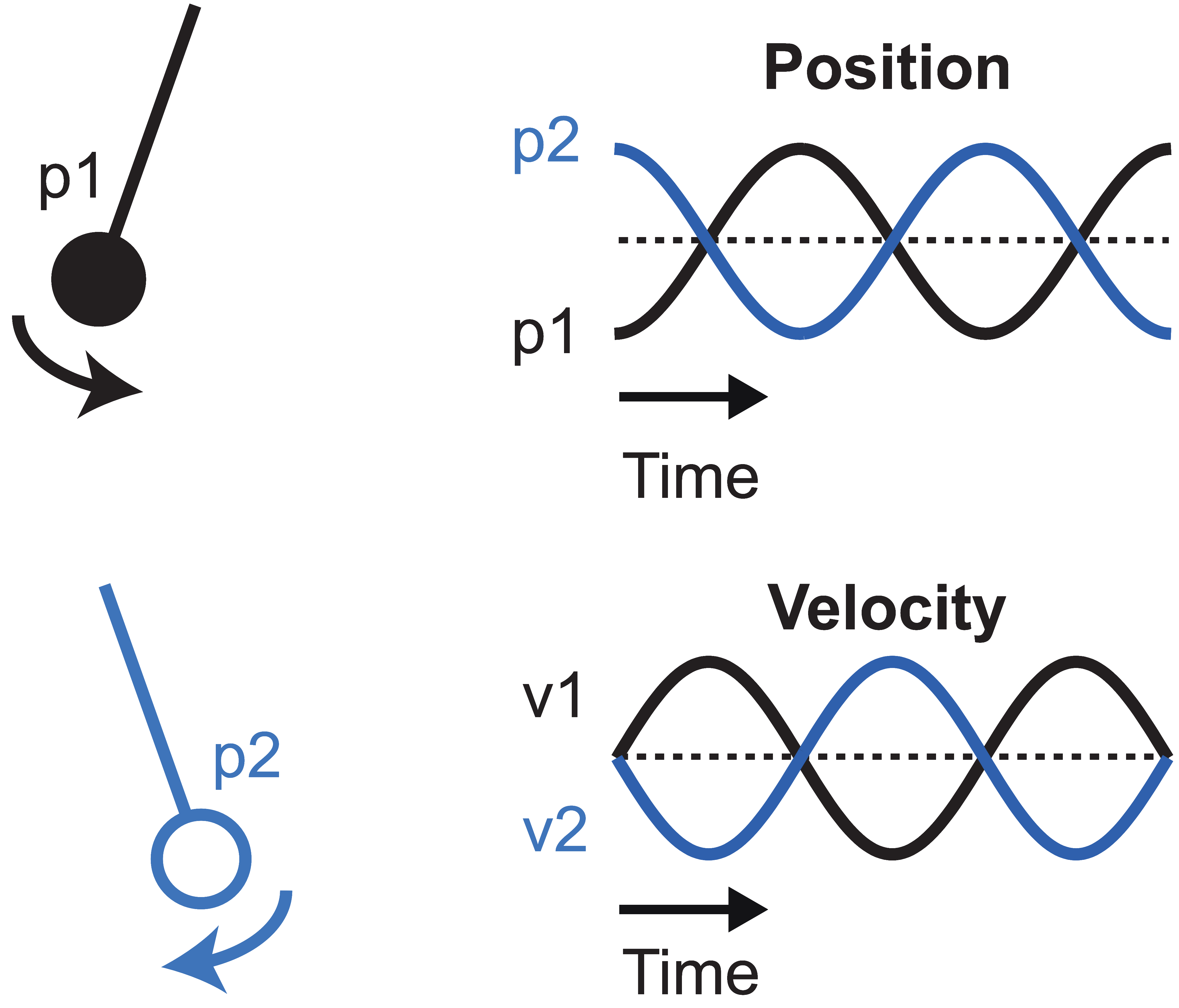




$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

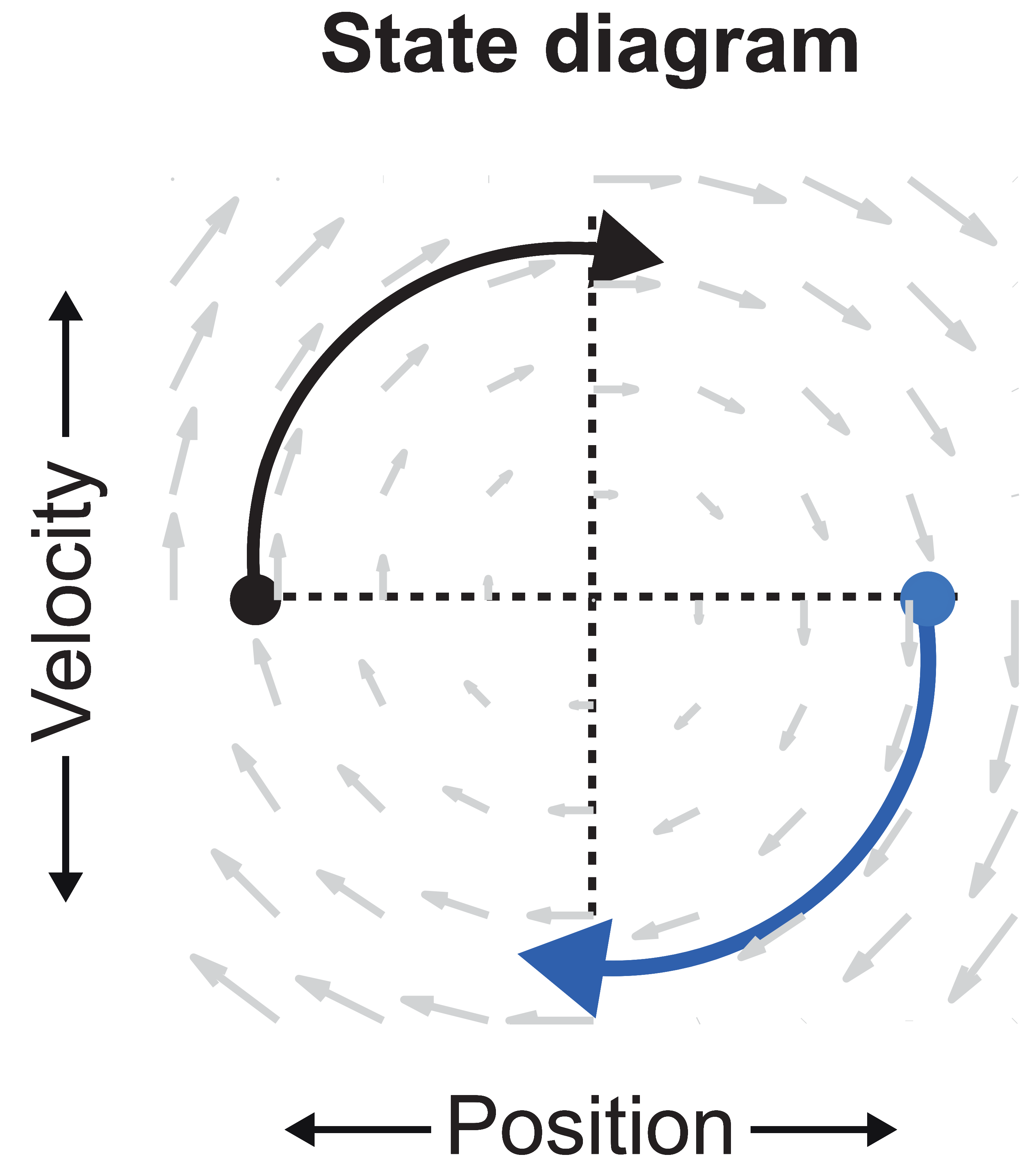
$$\dot{\mathbf{x}} = f(\mathbf{x})$$





$$\mathbf{x} = \begin{bmatrix} p \\ v \end{bmatrix}$$

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

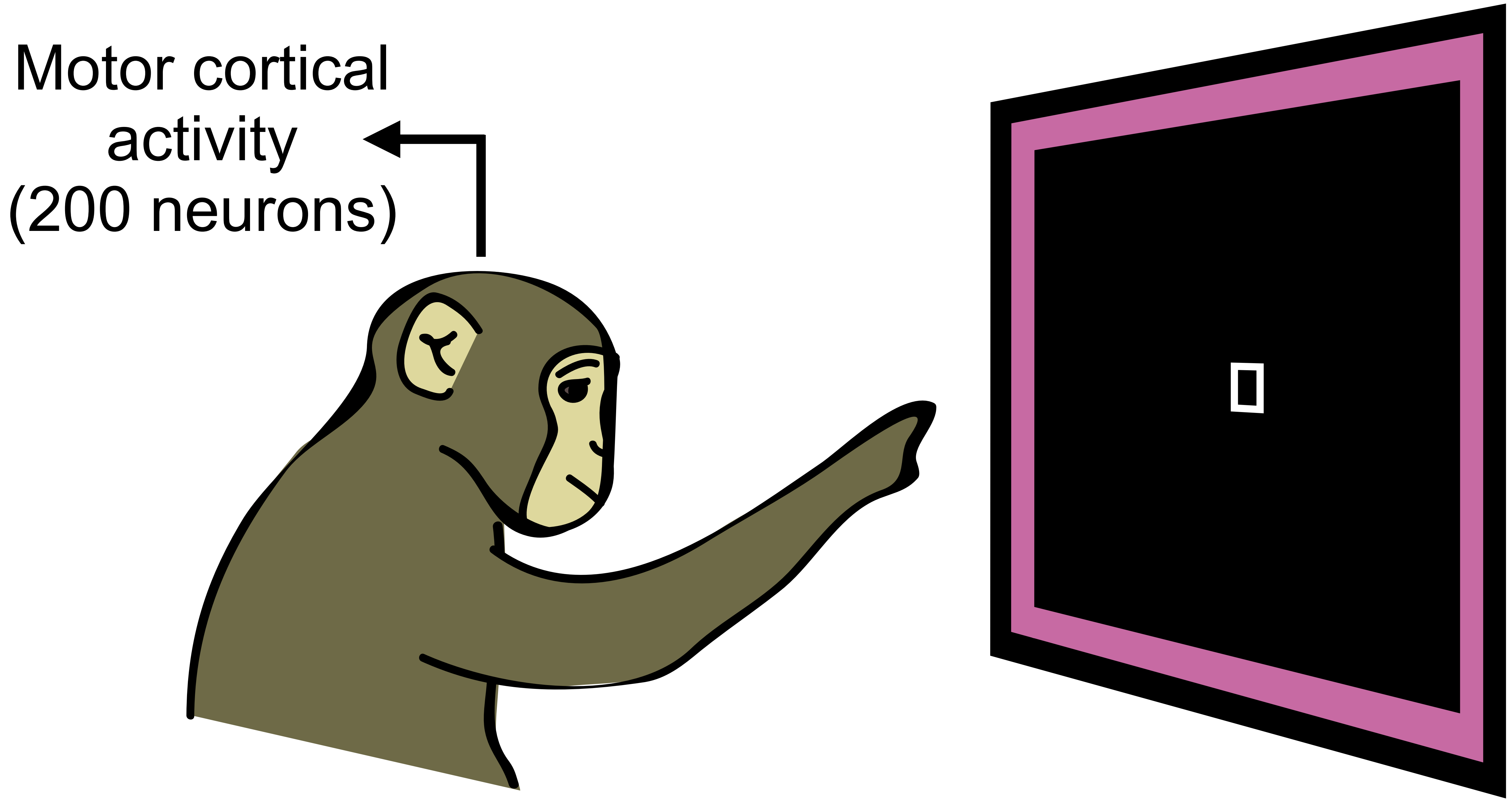


Predictable activity: delayed-reaching

“Maze” task

108 total conditions

- Motor cortex is set to an initial state during the preparatory phase
- Activity during movement execution is highly predictable based on initial state



Churchland*, Cunningham* … Shenoy, *Nature* 2012

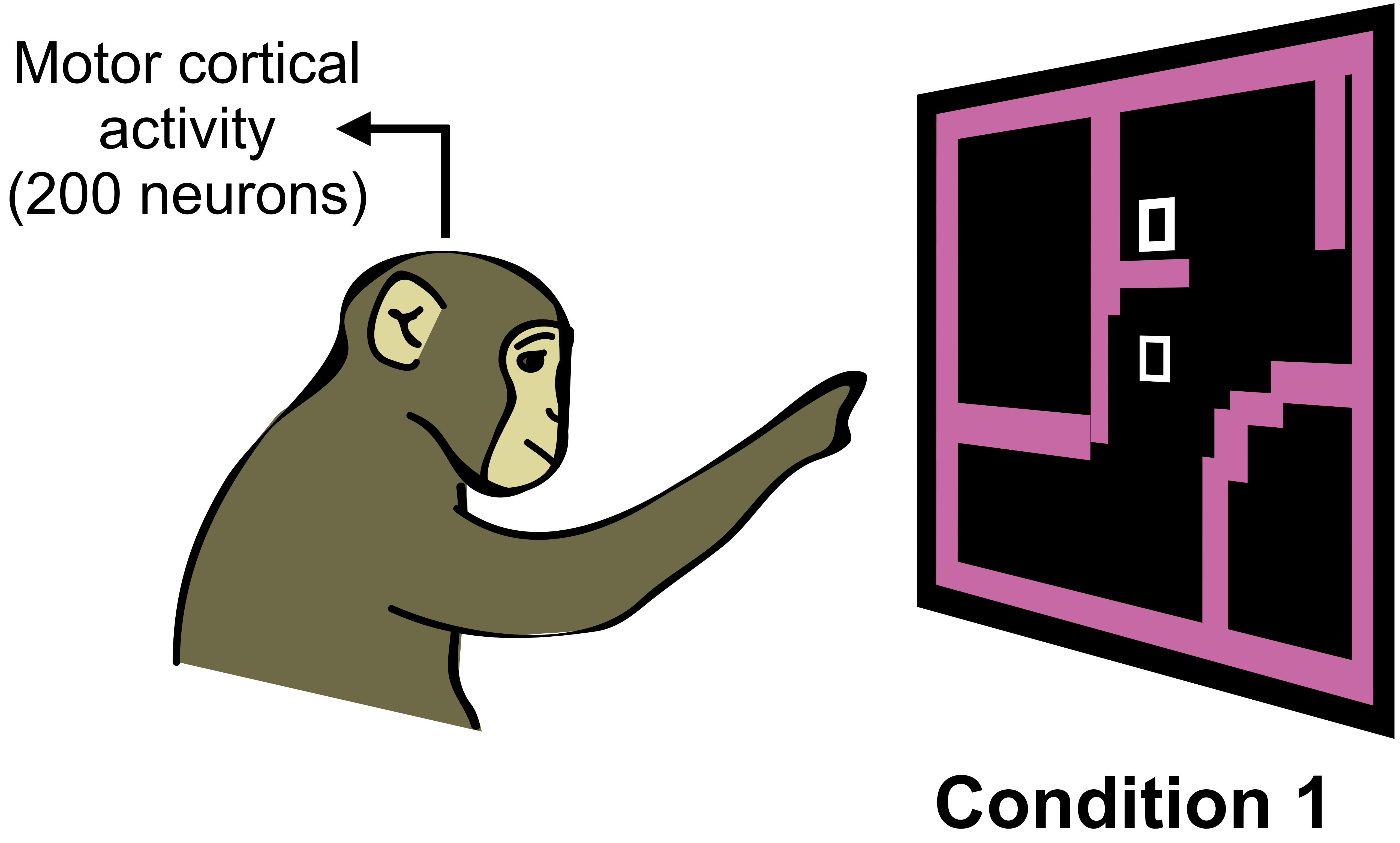
Shenoy, Sahani, Churchland, *Ann Rev Neuro* 2013

Predictable activity: delayed-reaching

"Maze" task

108 total conditions

- Motor cortex is set to an initial state during the preparatory phase
- Activity during movement execution is highly predictable based on initial state



Churchland*, Cunningham* … Shenoy, *Nature* 2012

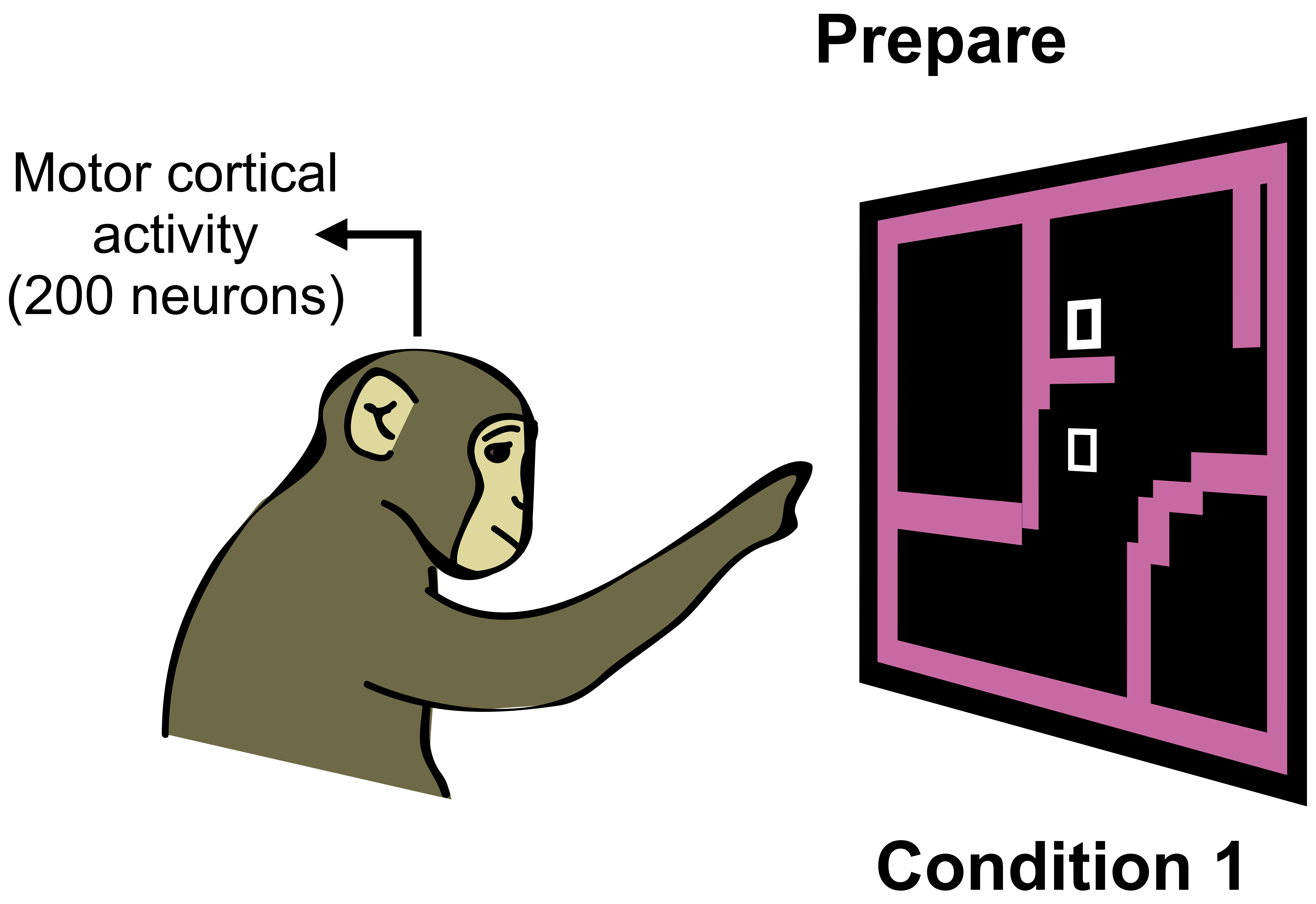
Shenoy, Sahani, Churchland, *Ann Rev Neuro* 2013

Predictable activity: delayed-reaching

"Maze" task

108 total conditions

- Motor cortex is set to an initial state during the preparatory phase
- Activity during movement execution is highly predictable based on initial state



Churchland*, Cunningham* … Shenoy, *Nature* 2012

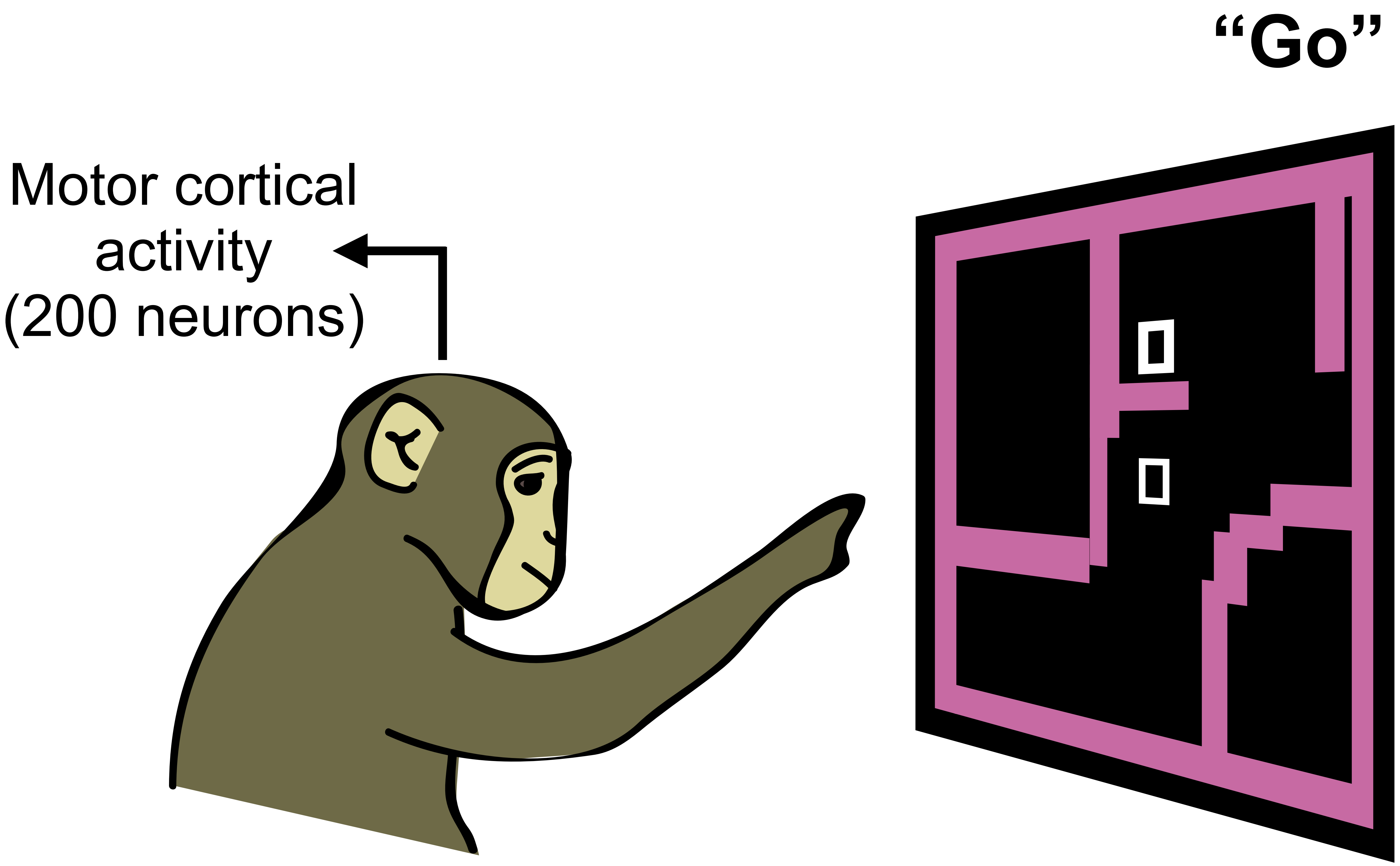
Shenoy, Sahani, Churchland, *Ann Rev Neuro* 2013

Predictable activity: delayed-reaching

"Maze" task

108 total conditions

- Motor cortex is set to an initial state during the preparatory phase
- Activity during movement execution is highly predictable based on initial state



Condition 1

Churchland*, Cunningham* … Shenoy, *Nature* 2012

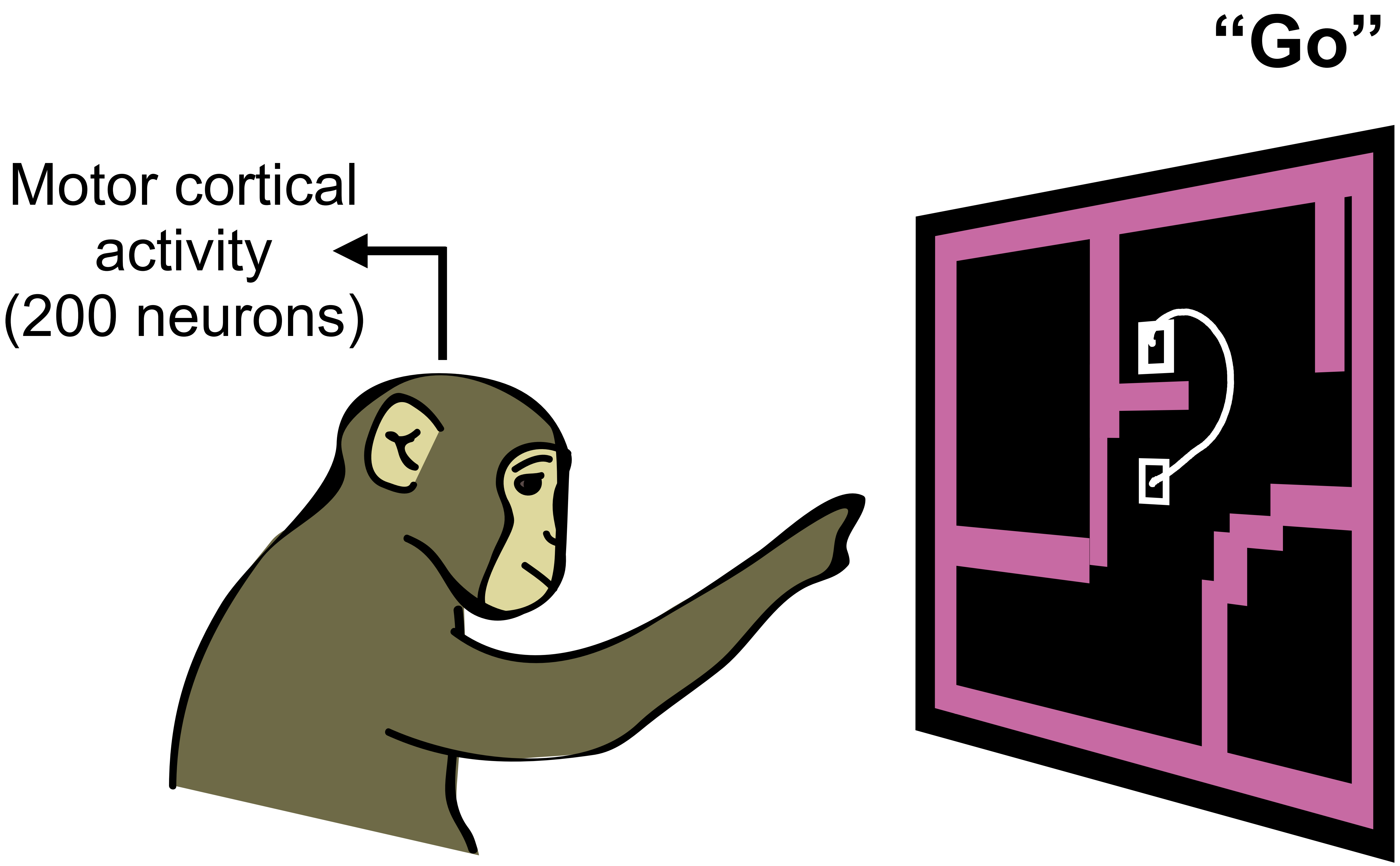
Shenoy, Sahani, Churchland, *Ann Rev Neuro* 2013

Predictable activity: delayed-reaching

“Maze” task

108 total conditions

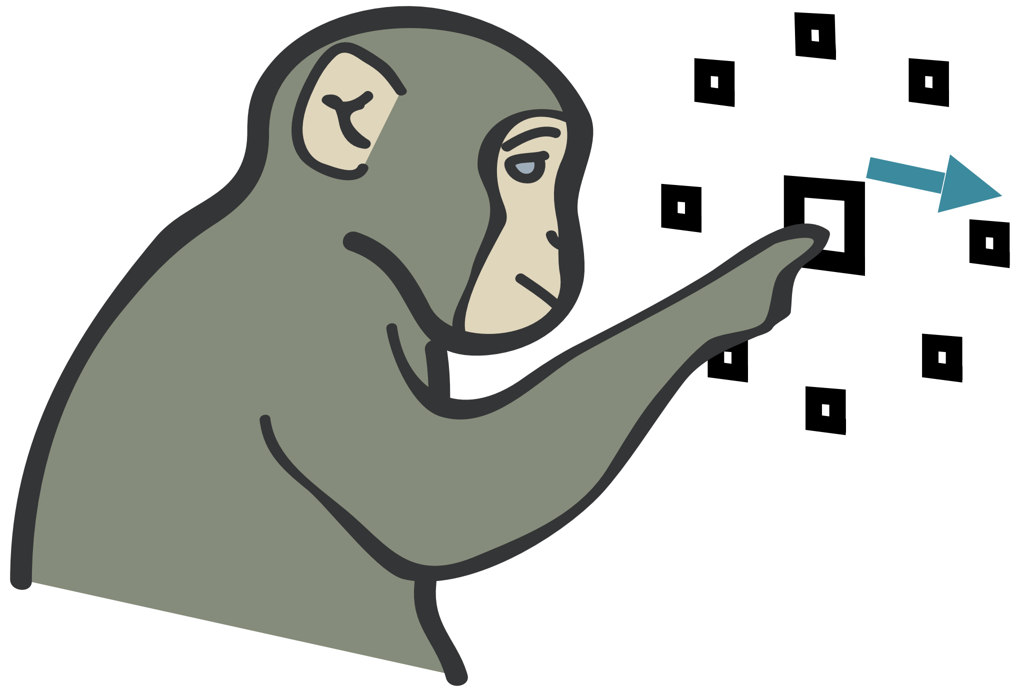
- Motor cortex is set to an initial state during the preparatory phase
- Activity during movement execution is highly predictable based on initial state



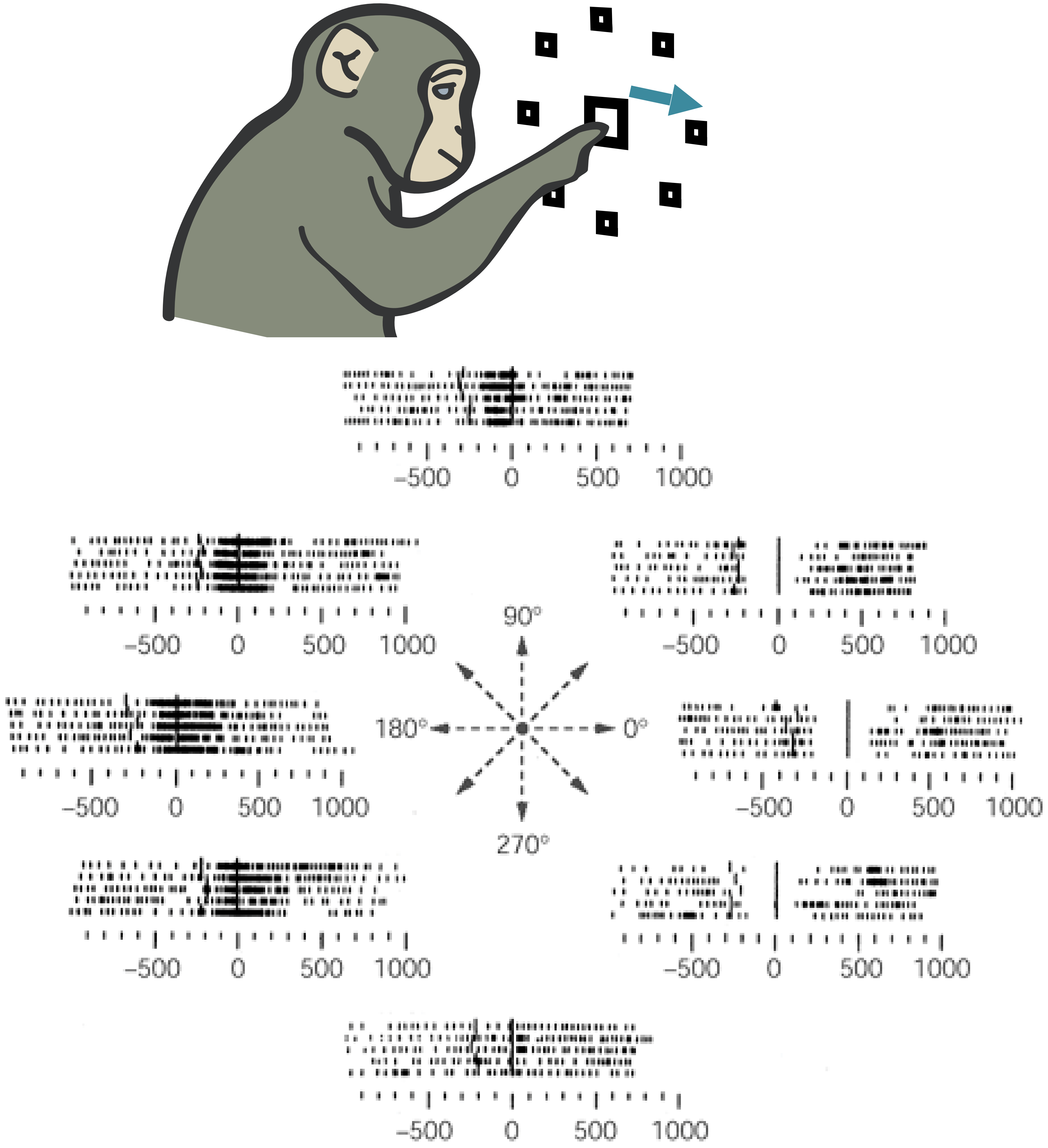
Condition 1

Churchland*, Cunningham* … Shenoy, *Nature* 2012

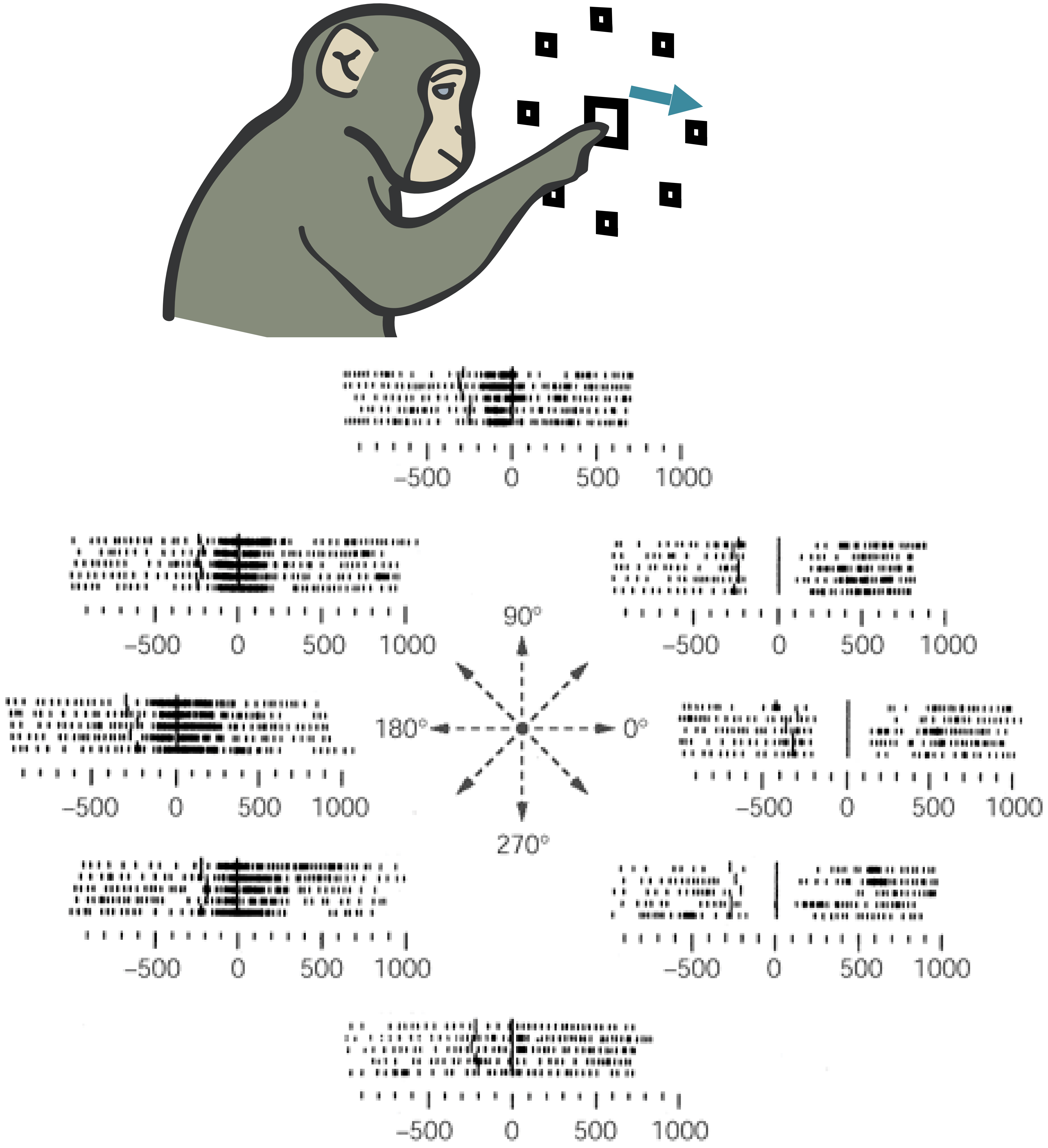
Shenoy, Sahani, Churchland, *Ann Rev Neuro* 2013

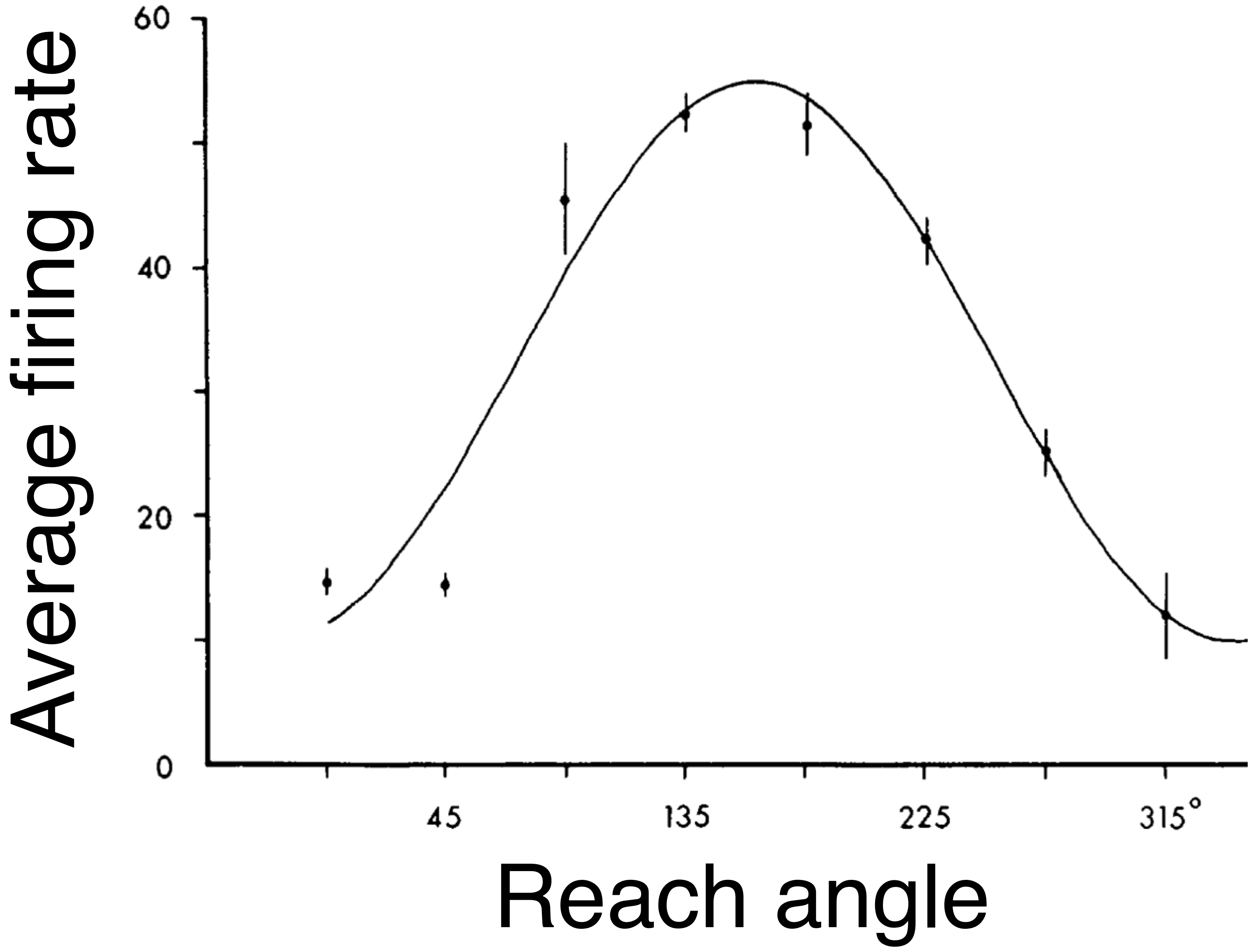


Georgopoulous et al. (1982)
Schwartz et al. (1988)

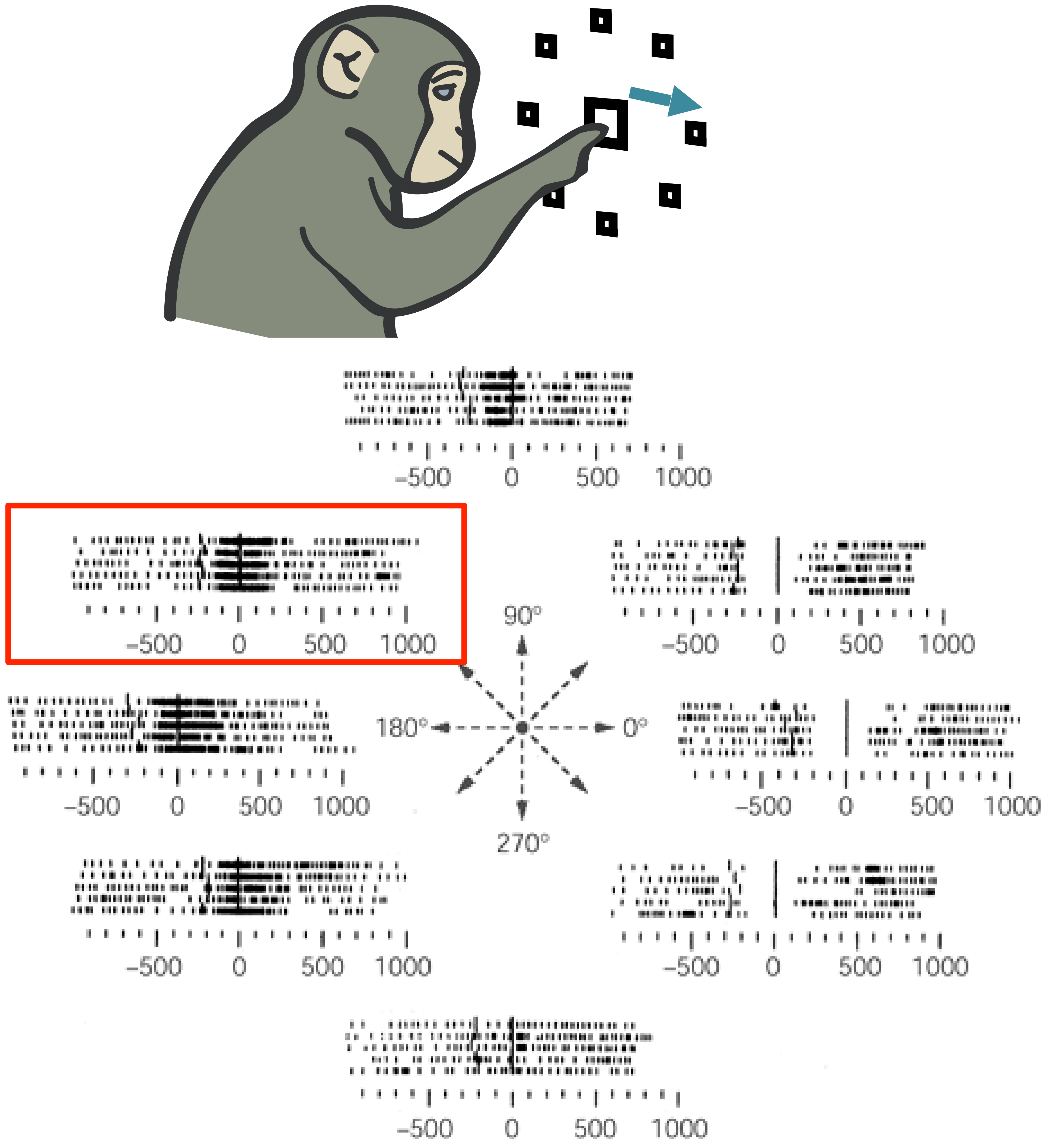


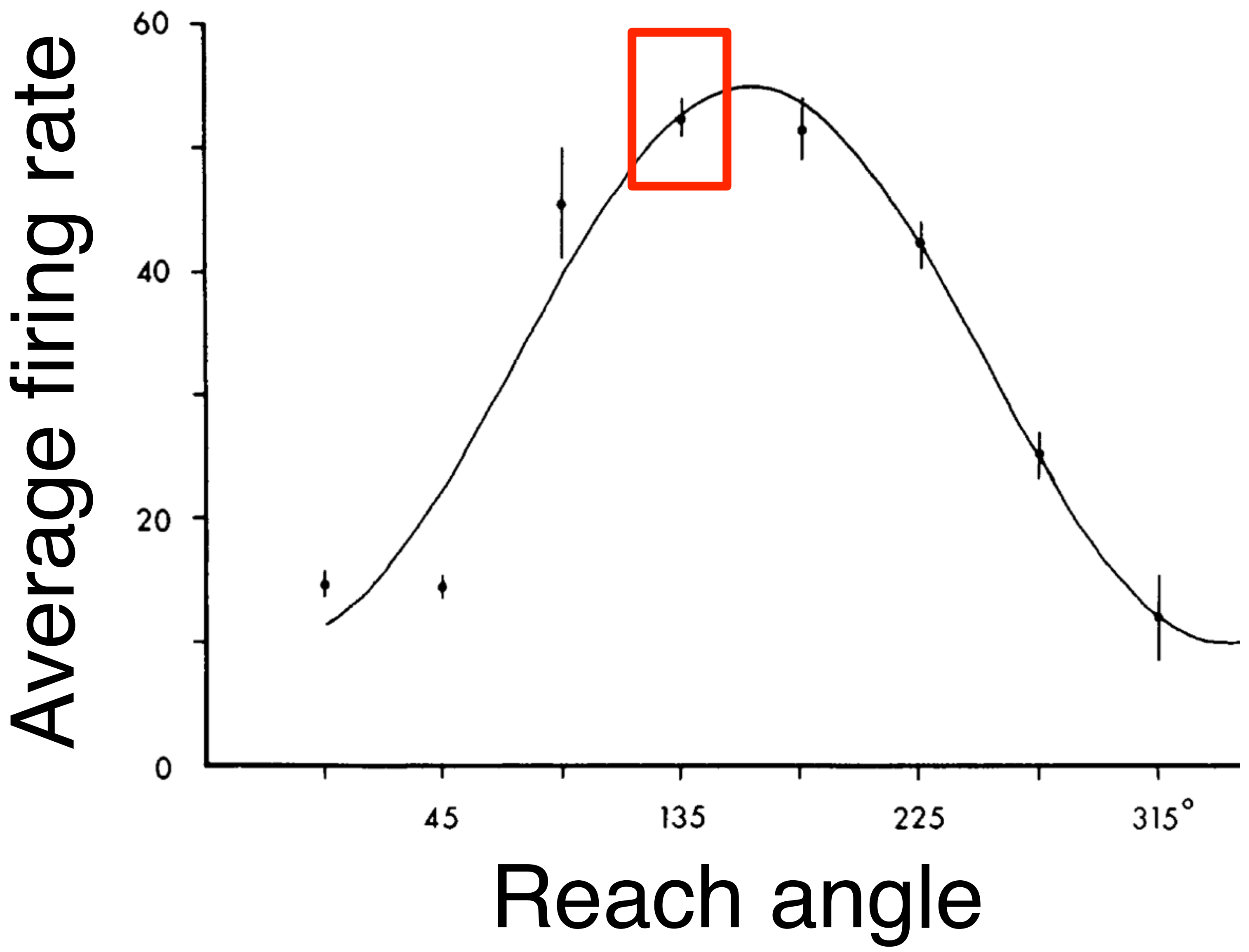
Georgopoulous et al. (1982)
Schwartz et al. (1988)



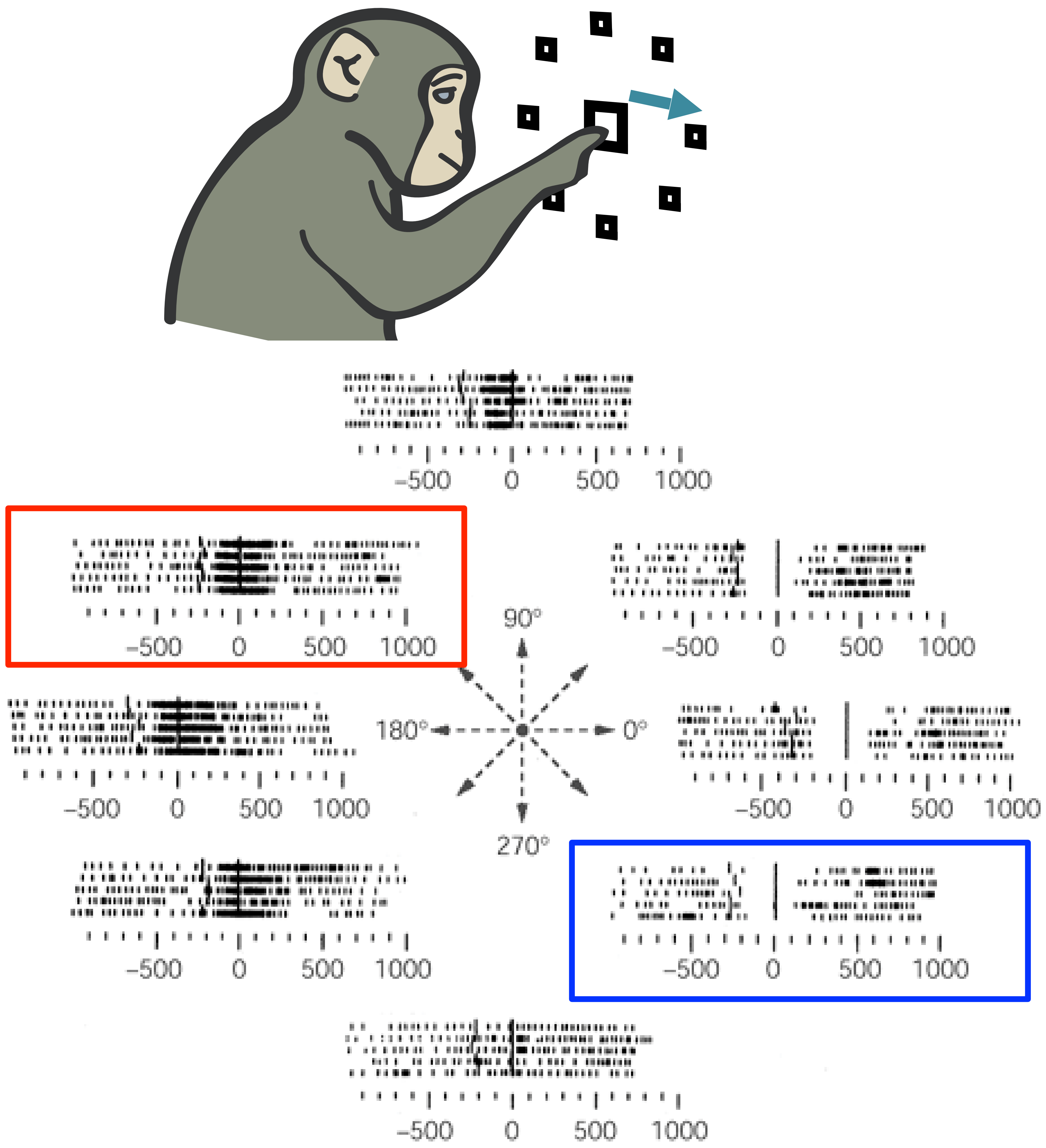


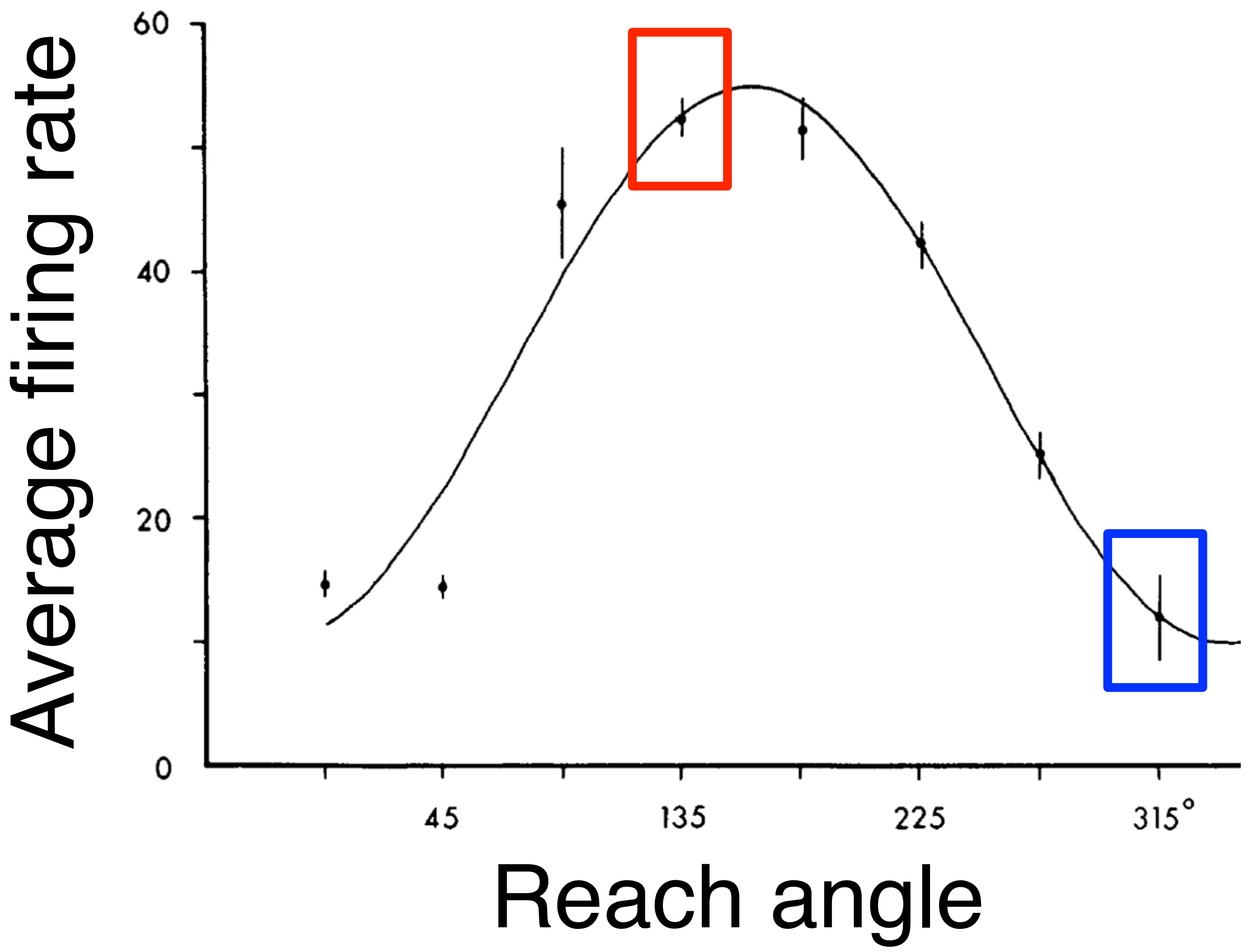
Georgopoulous et al. (1982)
Schwartz et al. (1988)



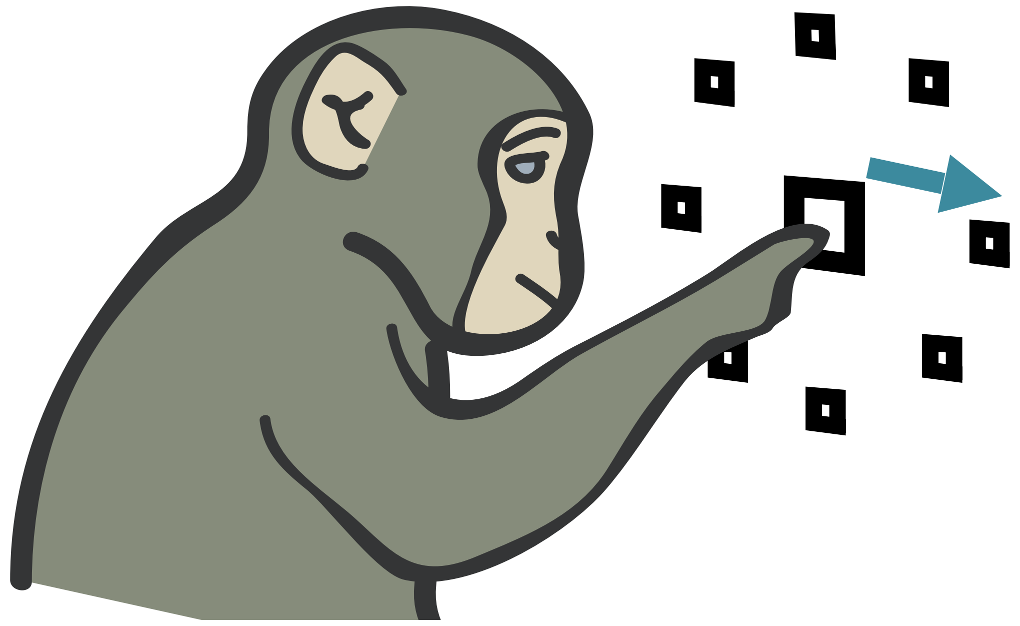


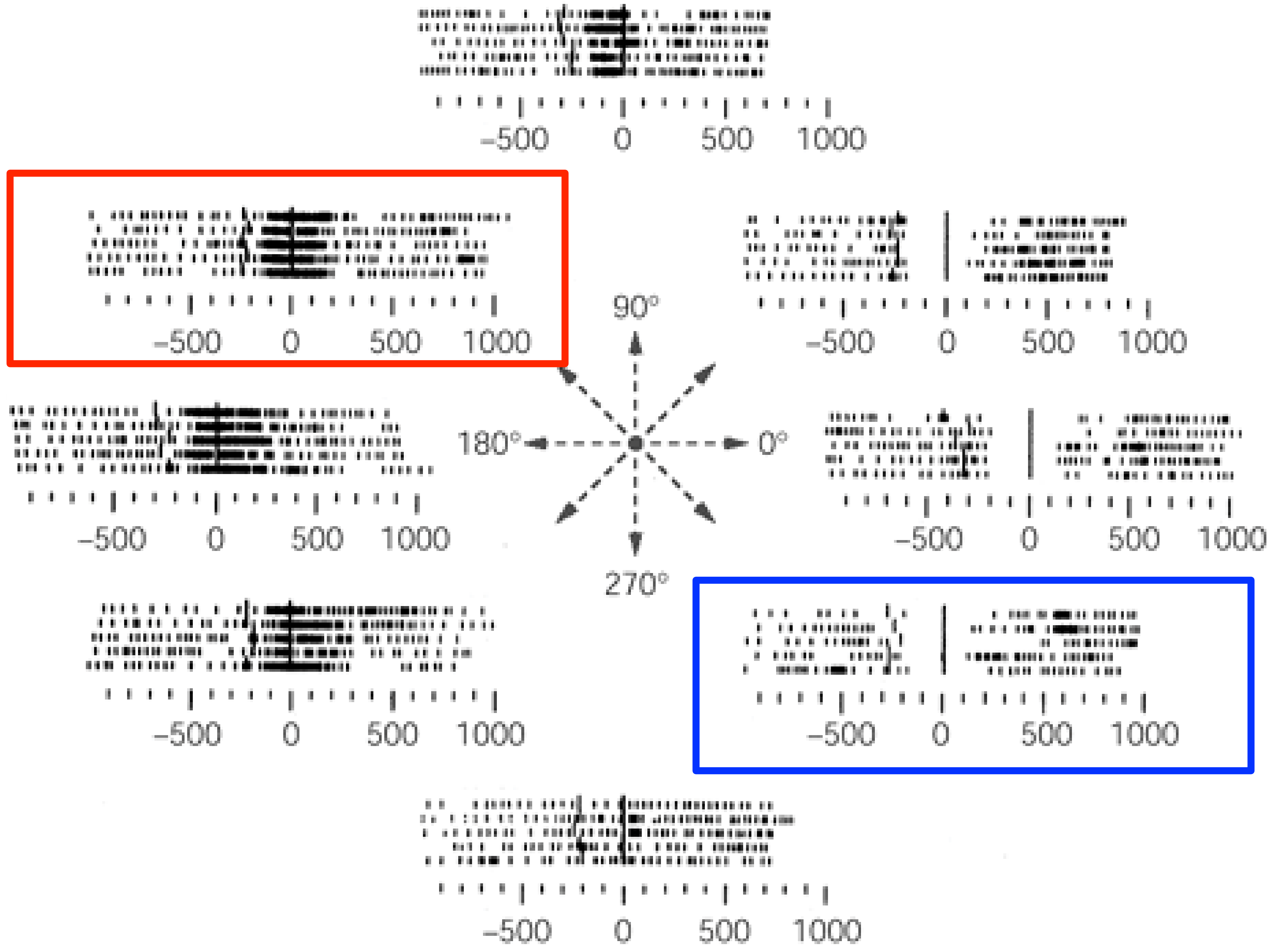
Georgopoulous et al. (1982)
Schwartz et al. (1988)

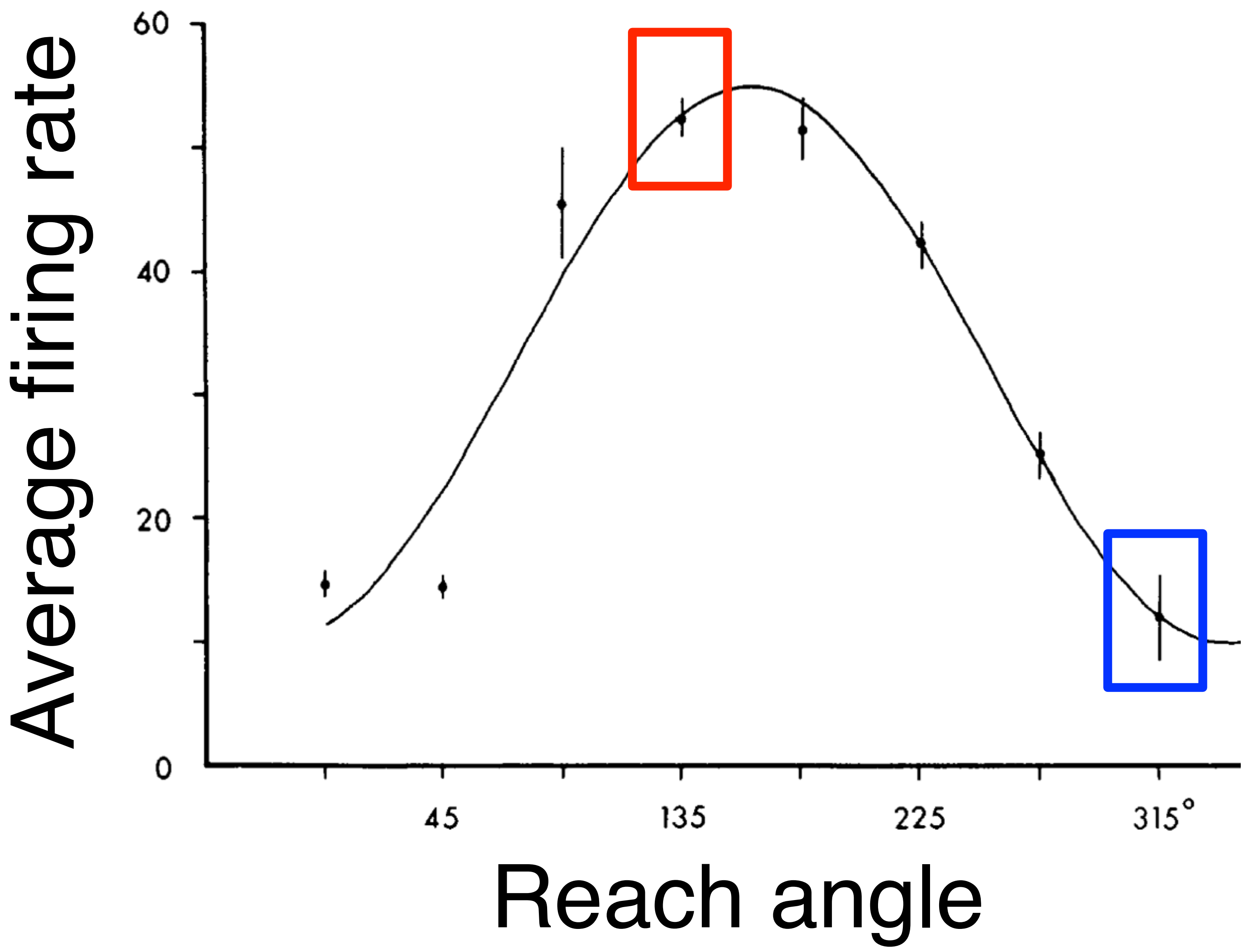




Georgopoulous et al. (1982)
Schwartz et al. (1988)







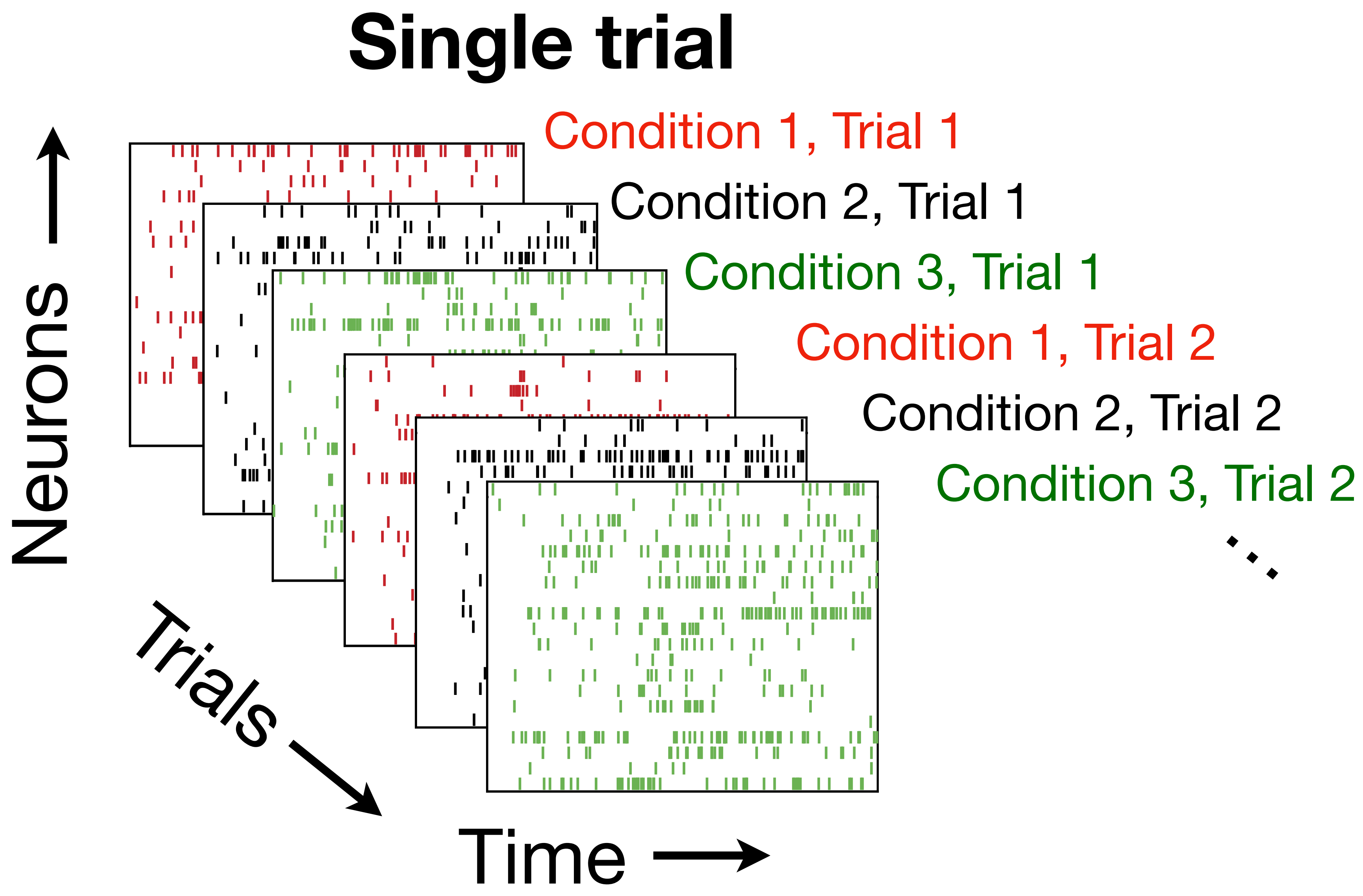
$$\mathbf{z} = f(\mathbf{k})$$

Neurons' firing rate

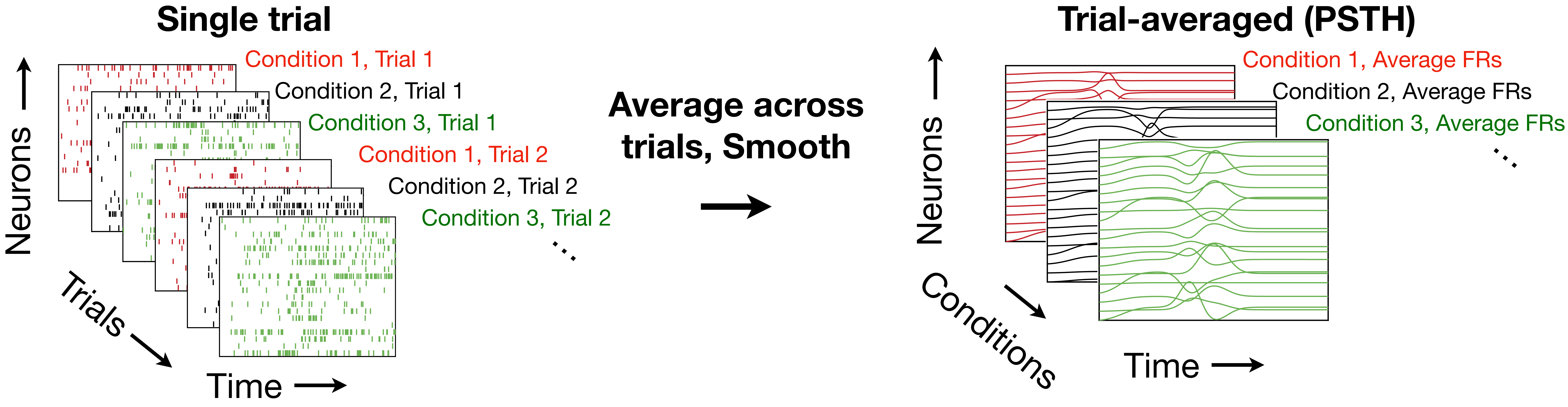
Kinematics
or kinetics

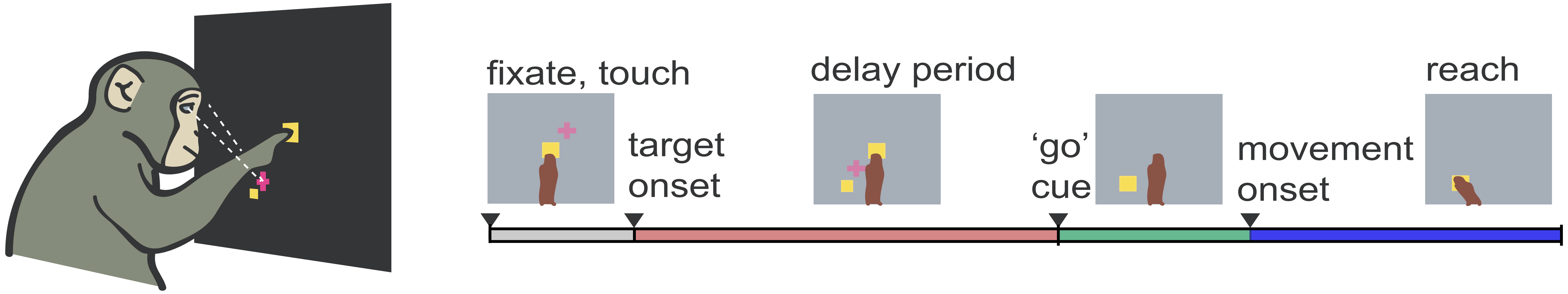
Georgopoulous et al. (1982)
Schwartz et al. (1988)

Visualizing the data

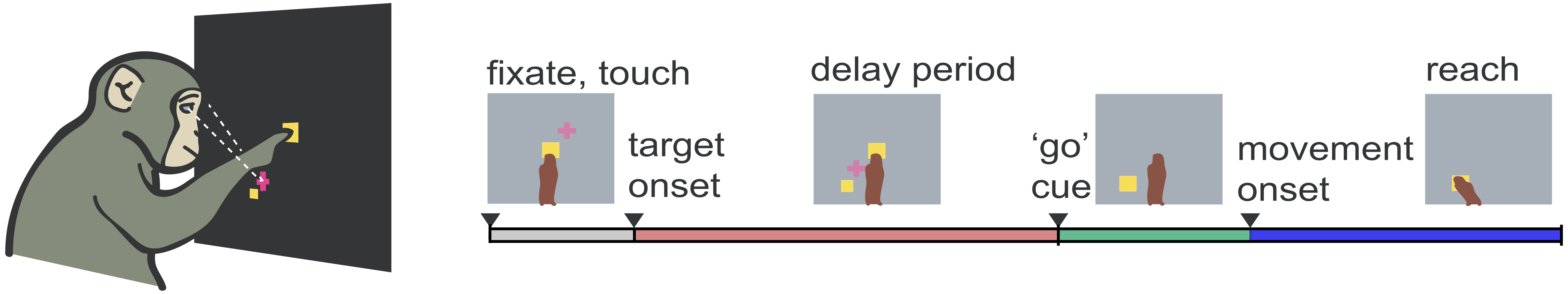


Visualizing the data

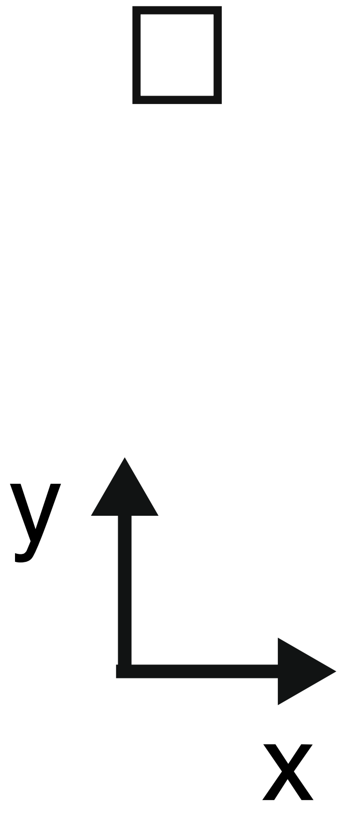




Churchland*, Cunningham* … Shenoy, *Nature* (2012)

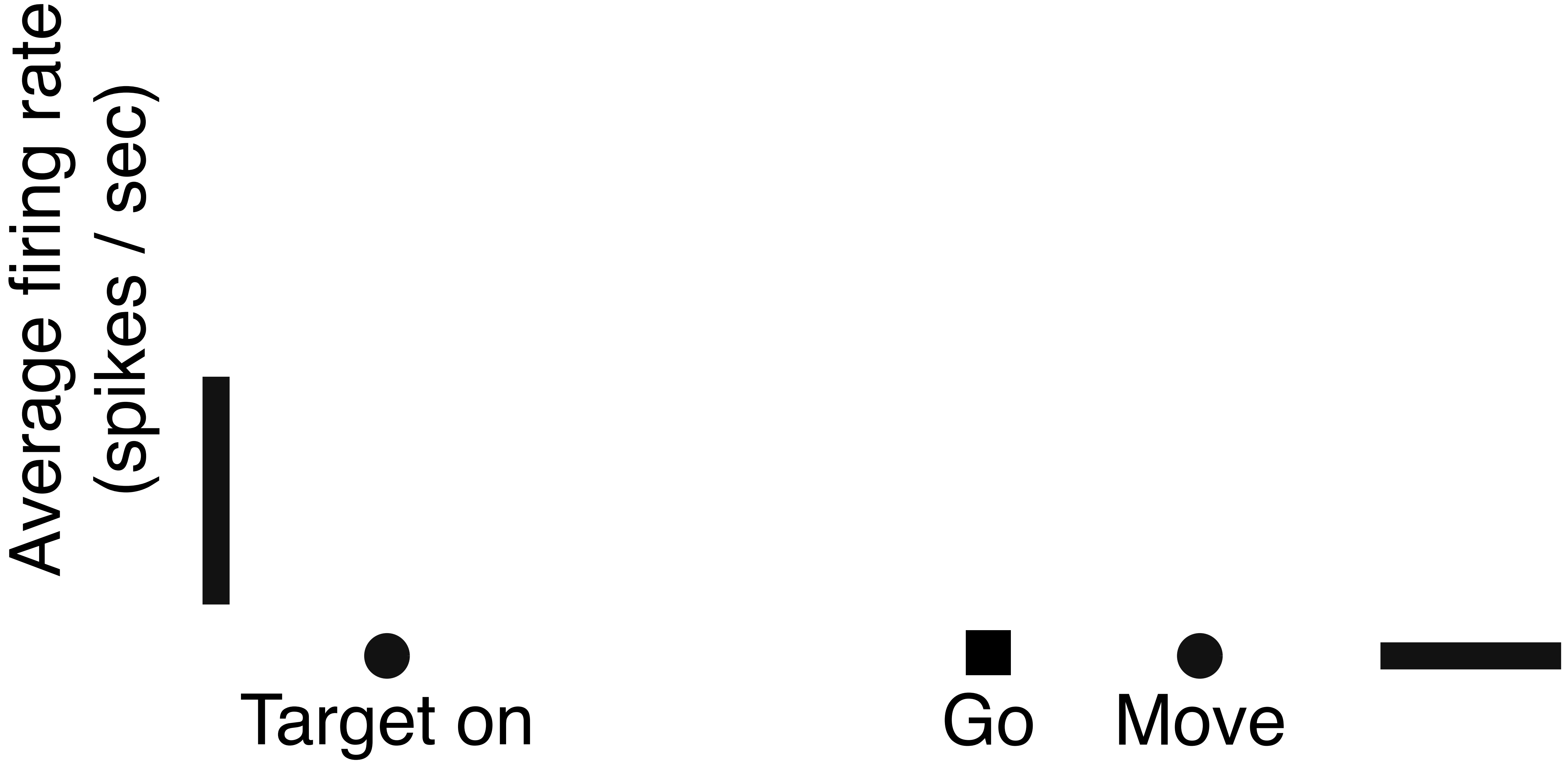


Reach trajectories

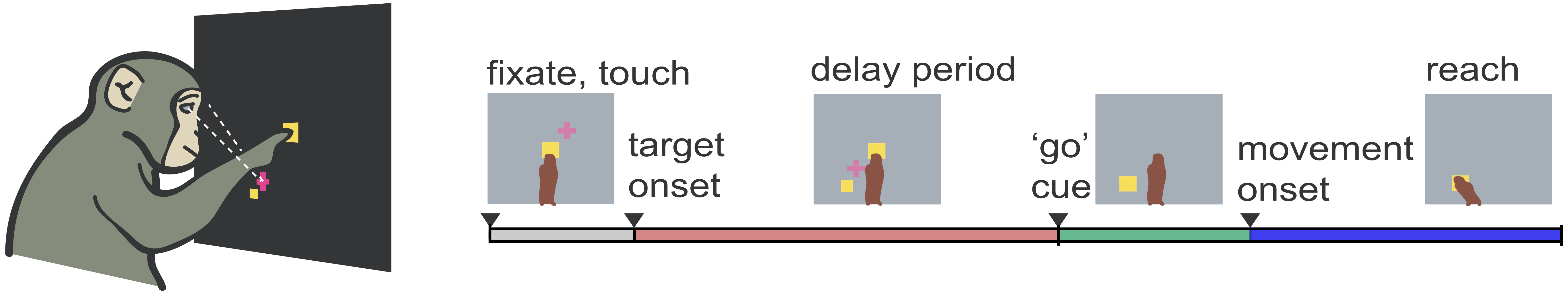


Single neuron firing rate (PSTH)

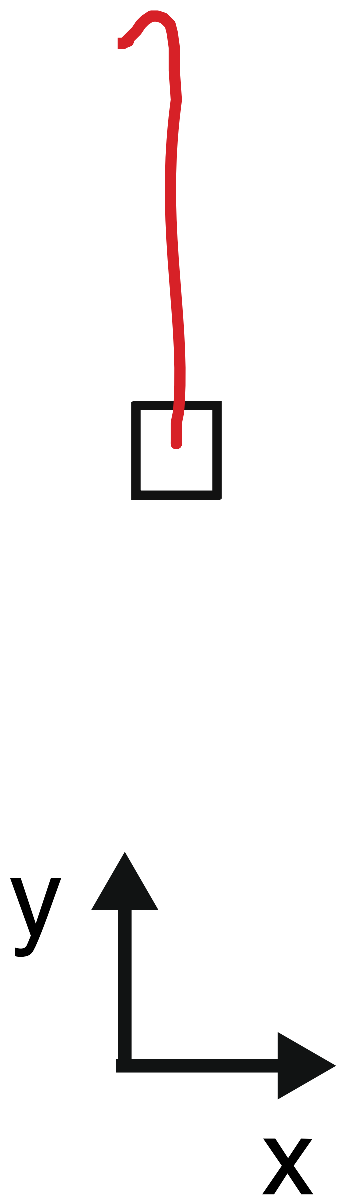
Cell 12
Monkey B



Churchland*, Cunningham* … Shenoy, *Nature* (2012)

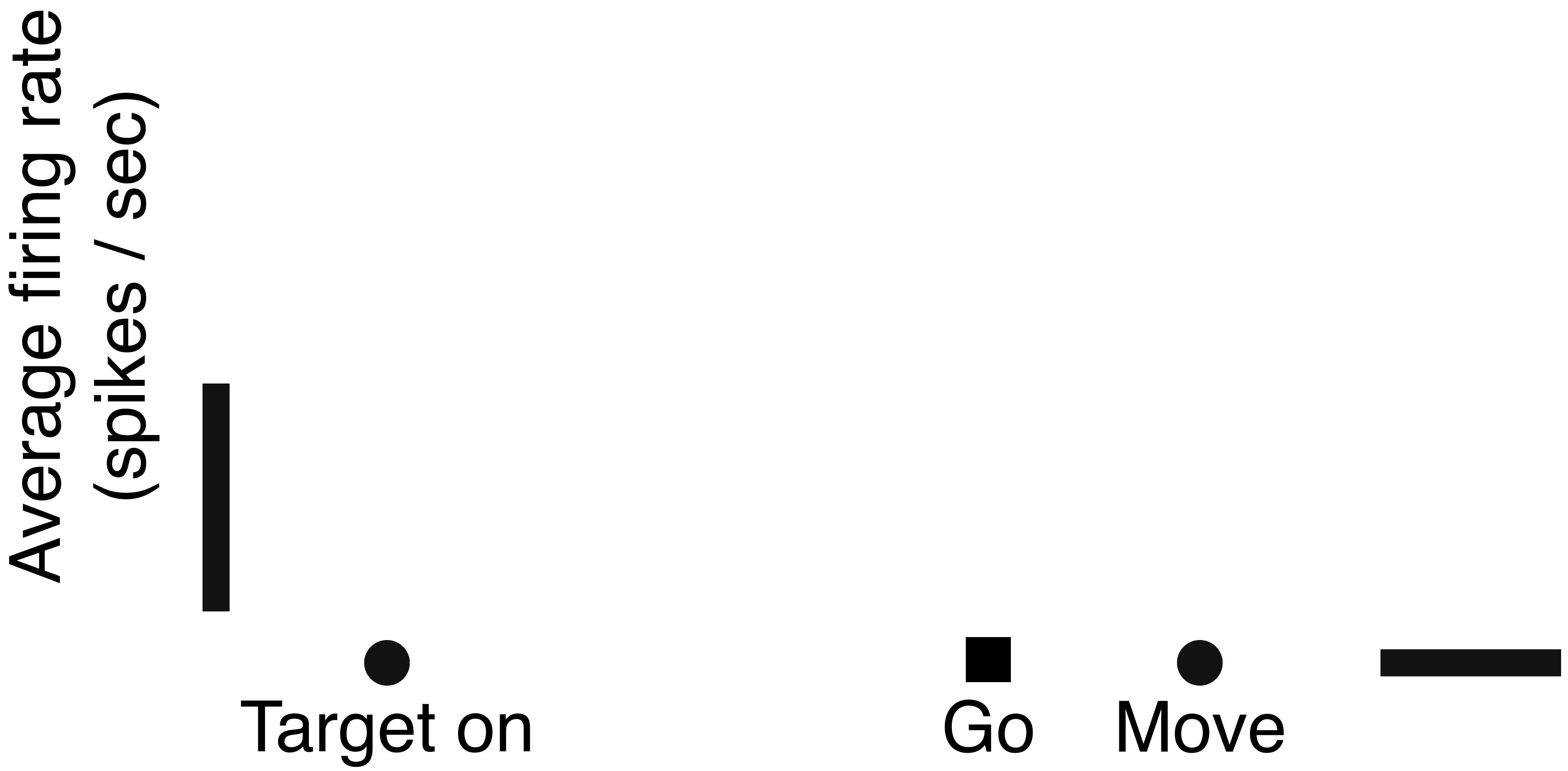


Reach trajectories

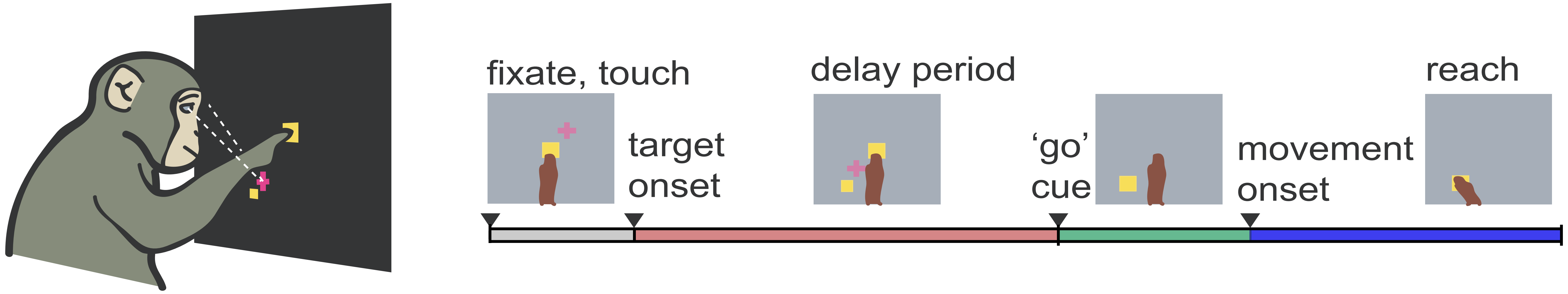


Single neuron firing rate (PSTH)

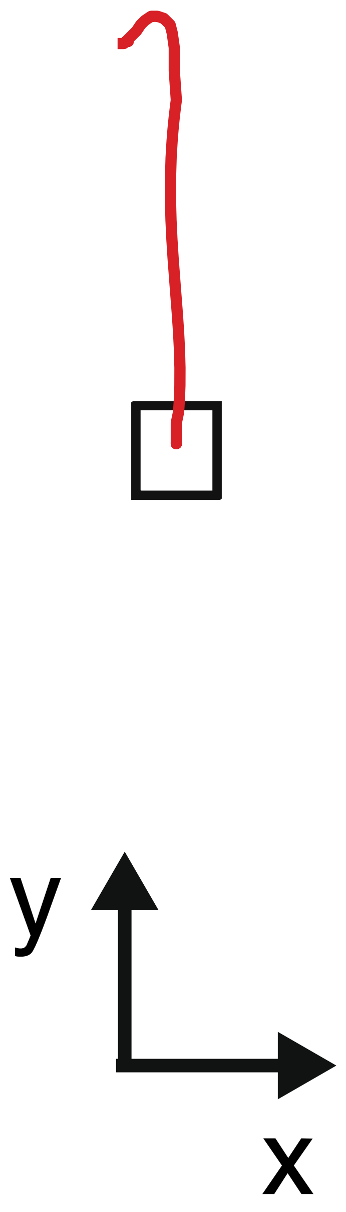
Cell 12
Monkey B



Churchland*, Cunningham* … Shenoy, *Nature* (2012)

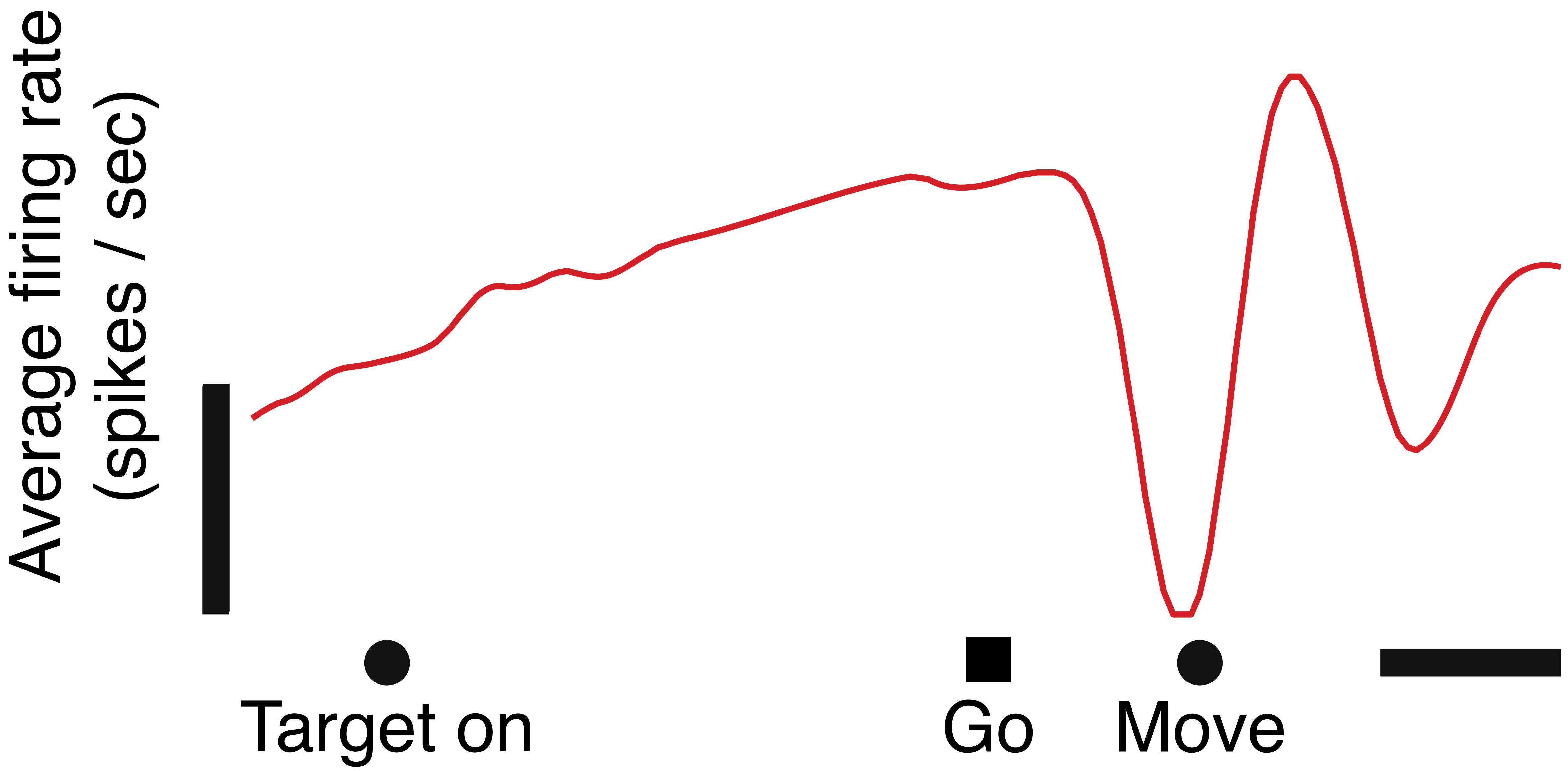


Reach trajectories

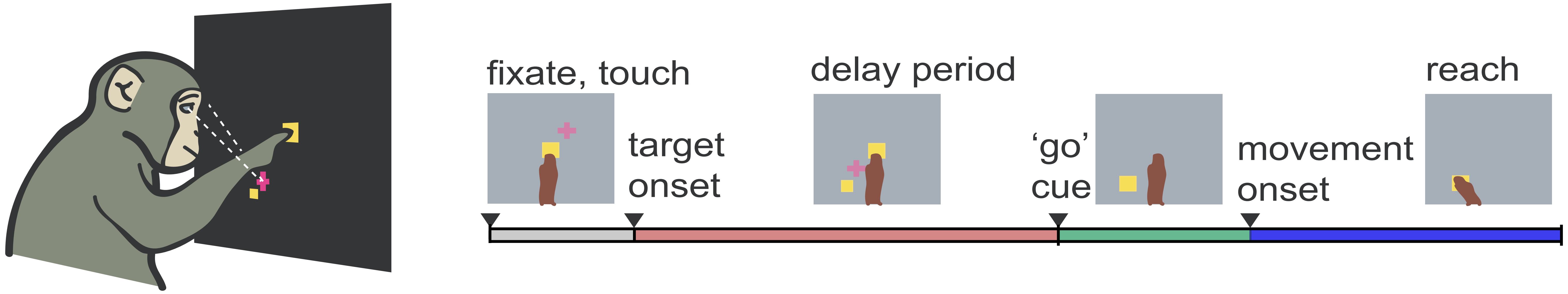


Single neuron firing rate (PSTH)

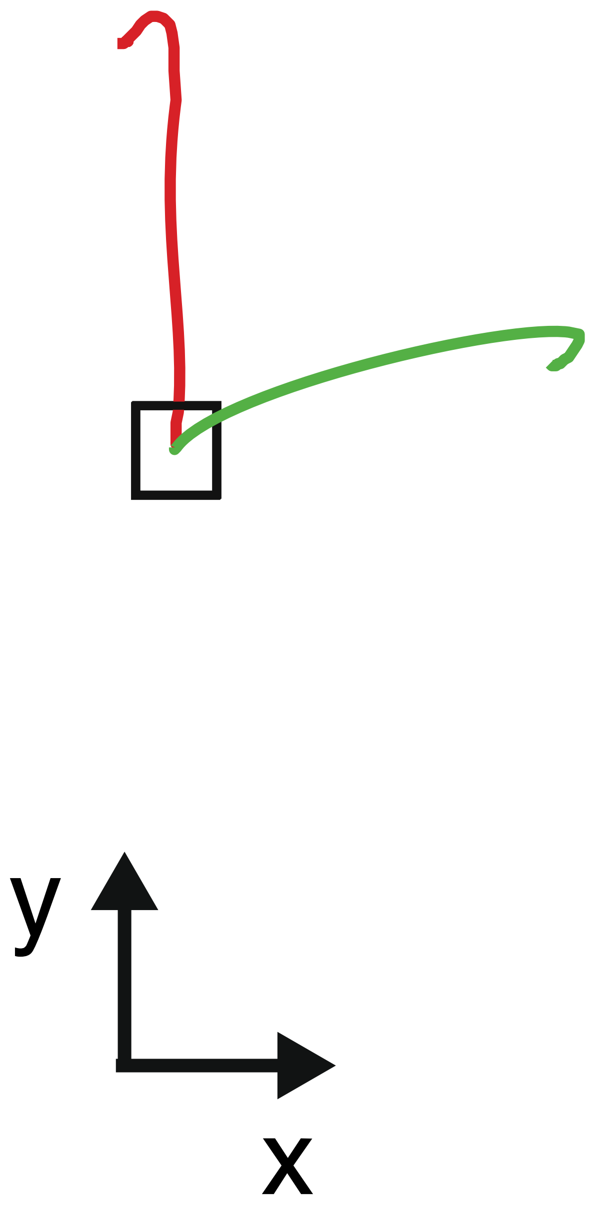
Cell 12
Monkey B



Churchland*, Cunningham* … Shenoy, *Nature* (2012)

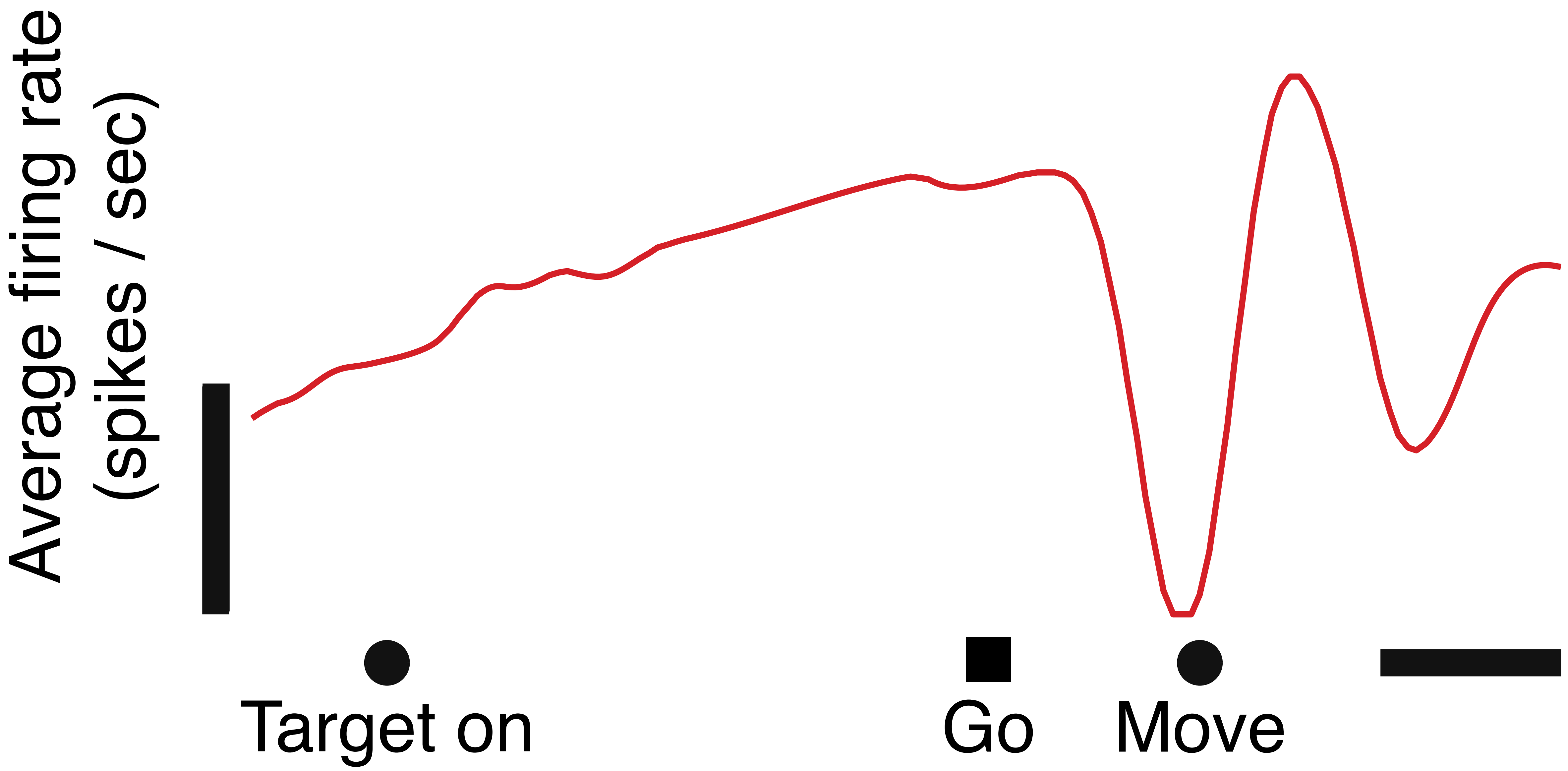


Reach trajectories

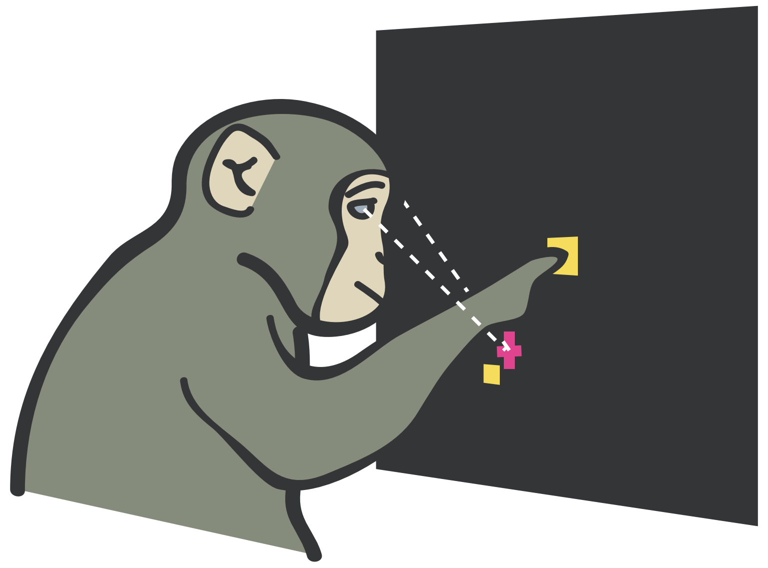


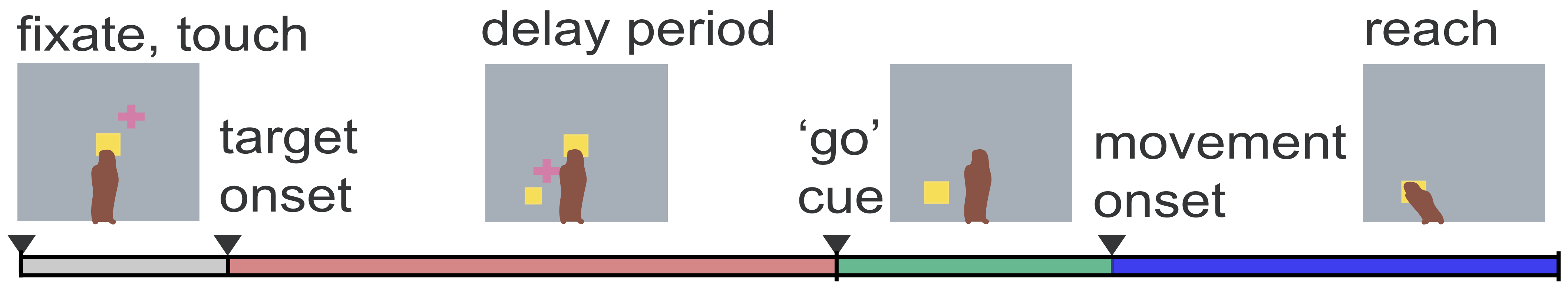
Single neuron firing rate (PSTH)

Cell 12
Monkey B

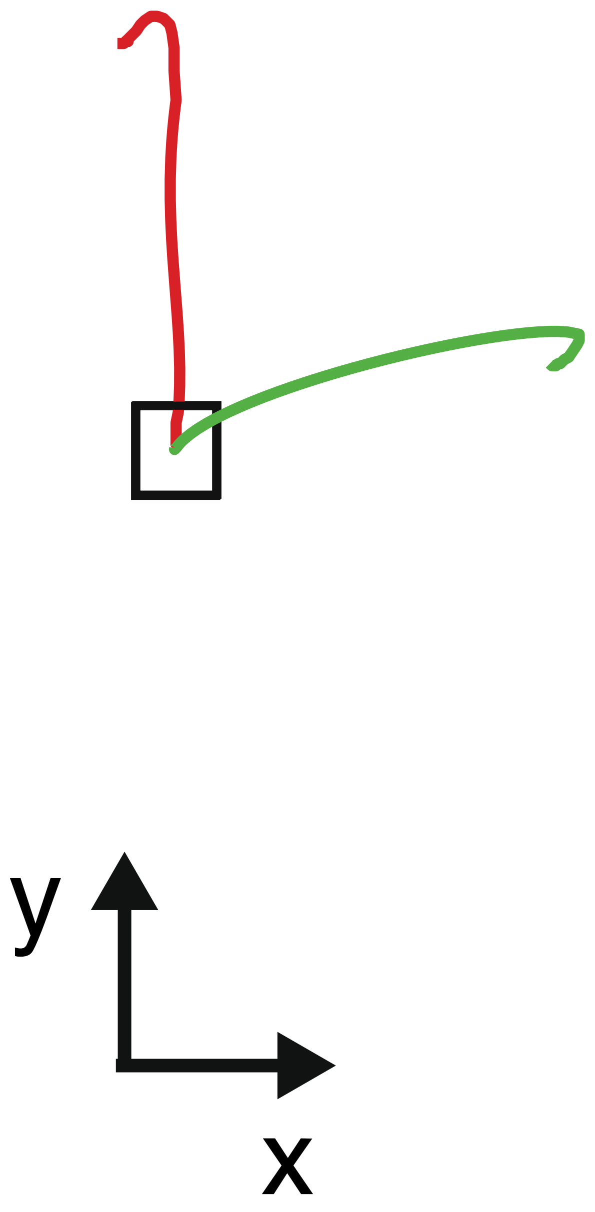


Churchland*, Cunningham* … Shenoy, *Nature* (2012)

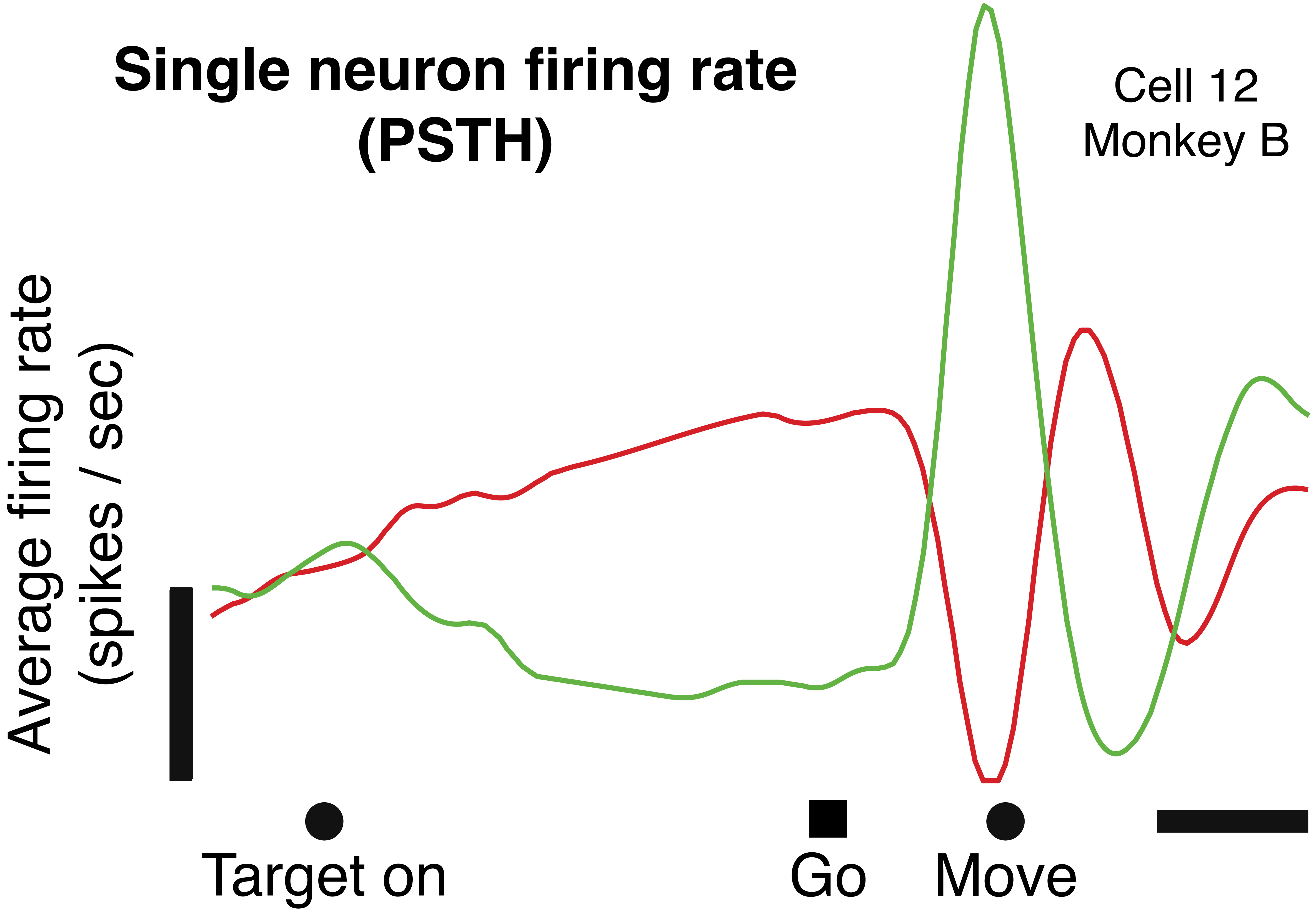




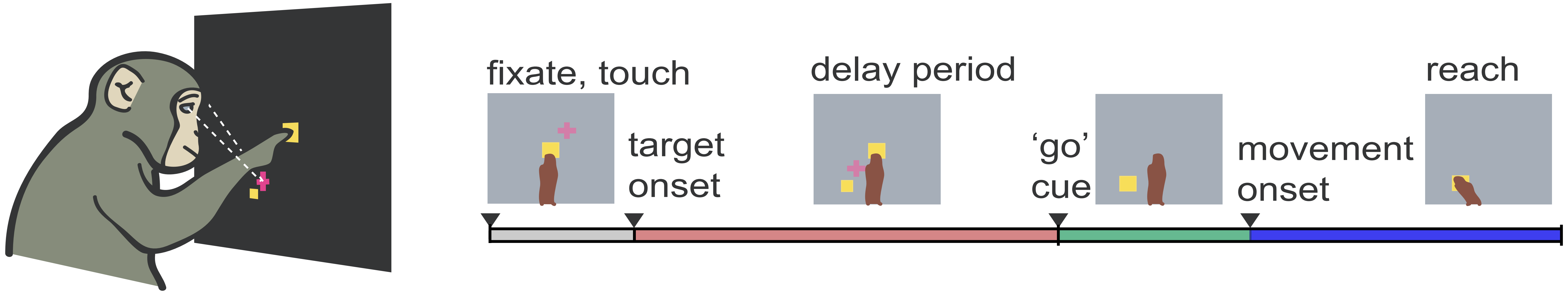
Reach trajectories



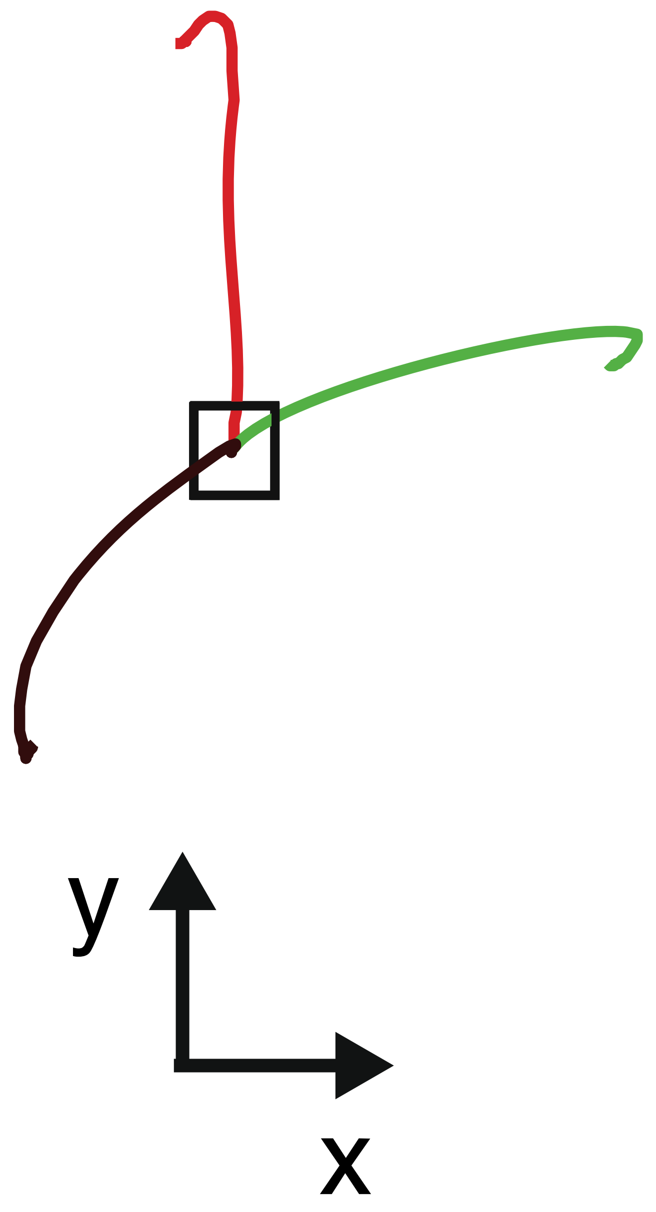
Single neuron firing rate (PSTH)



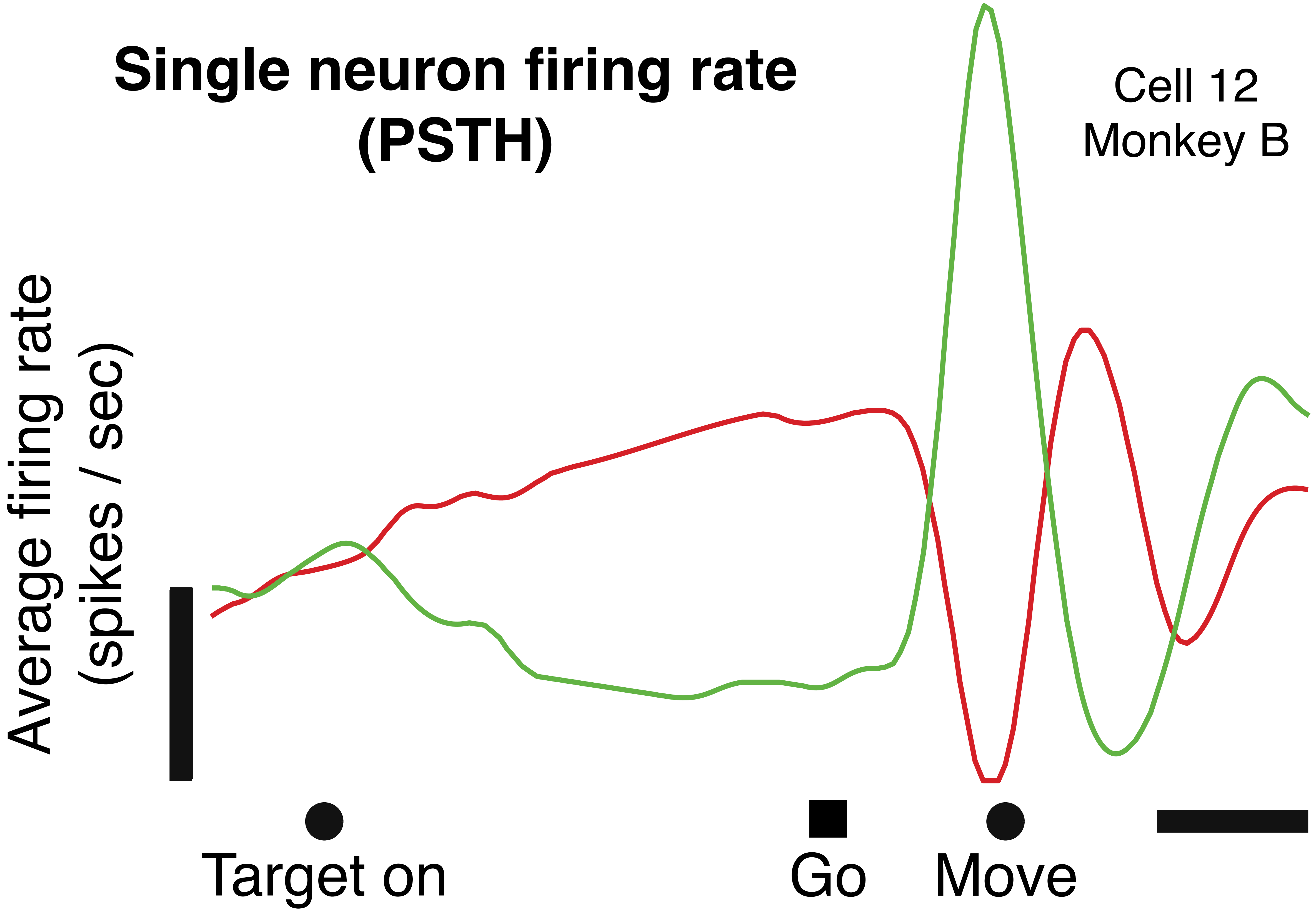
Churchland*, Cunningham* … Shenoy, *Nature* (2012)



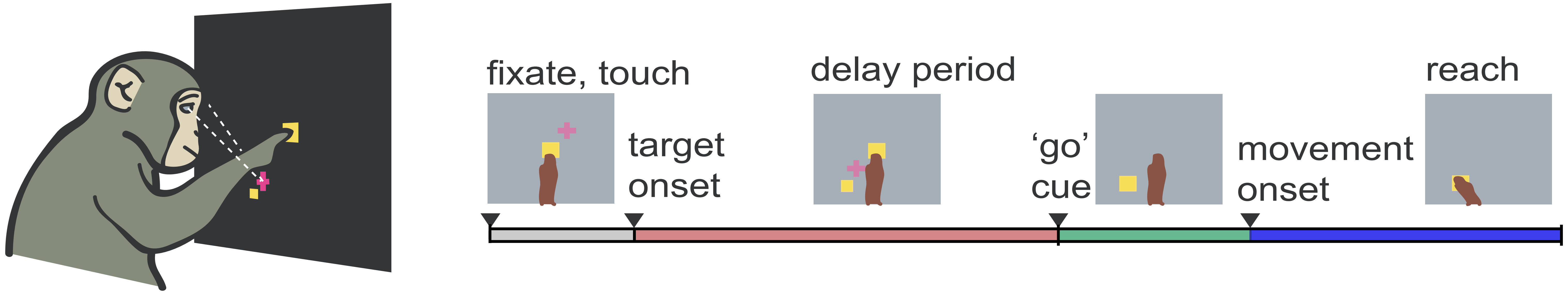
Reach trajectories



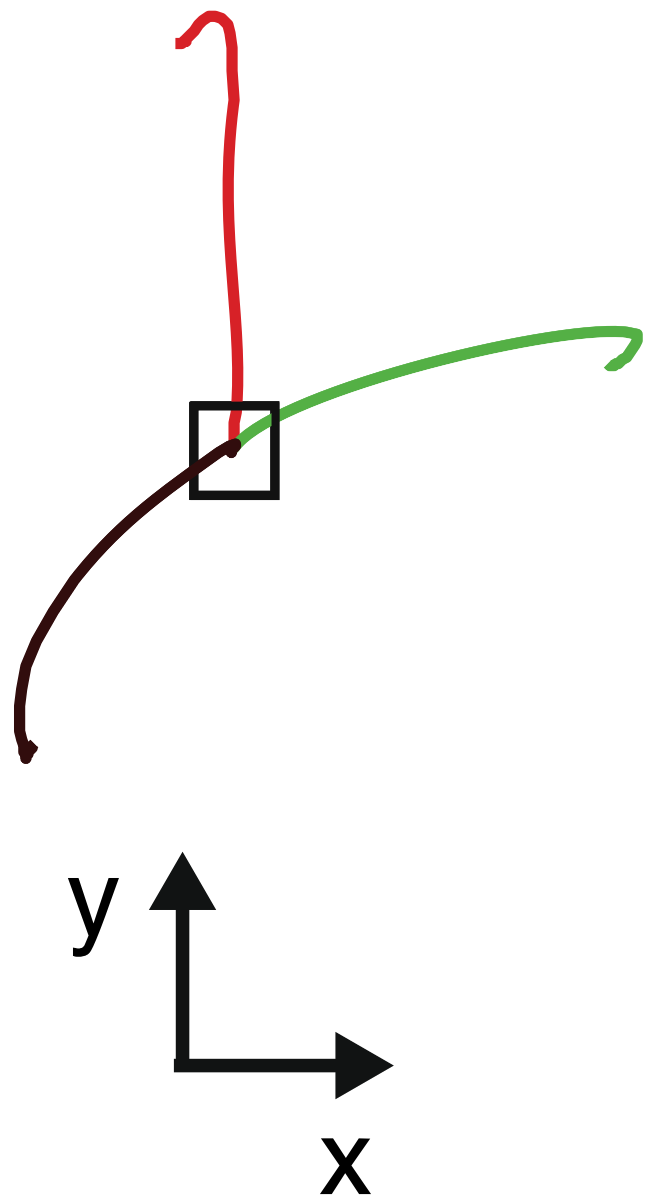
Single neuron firing rate (PSTH)



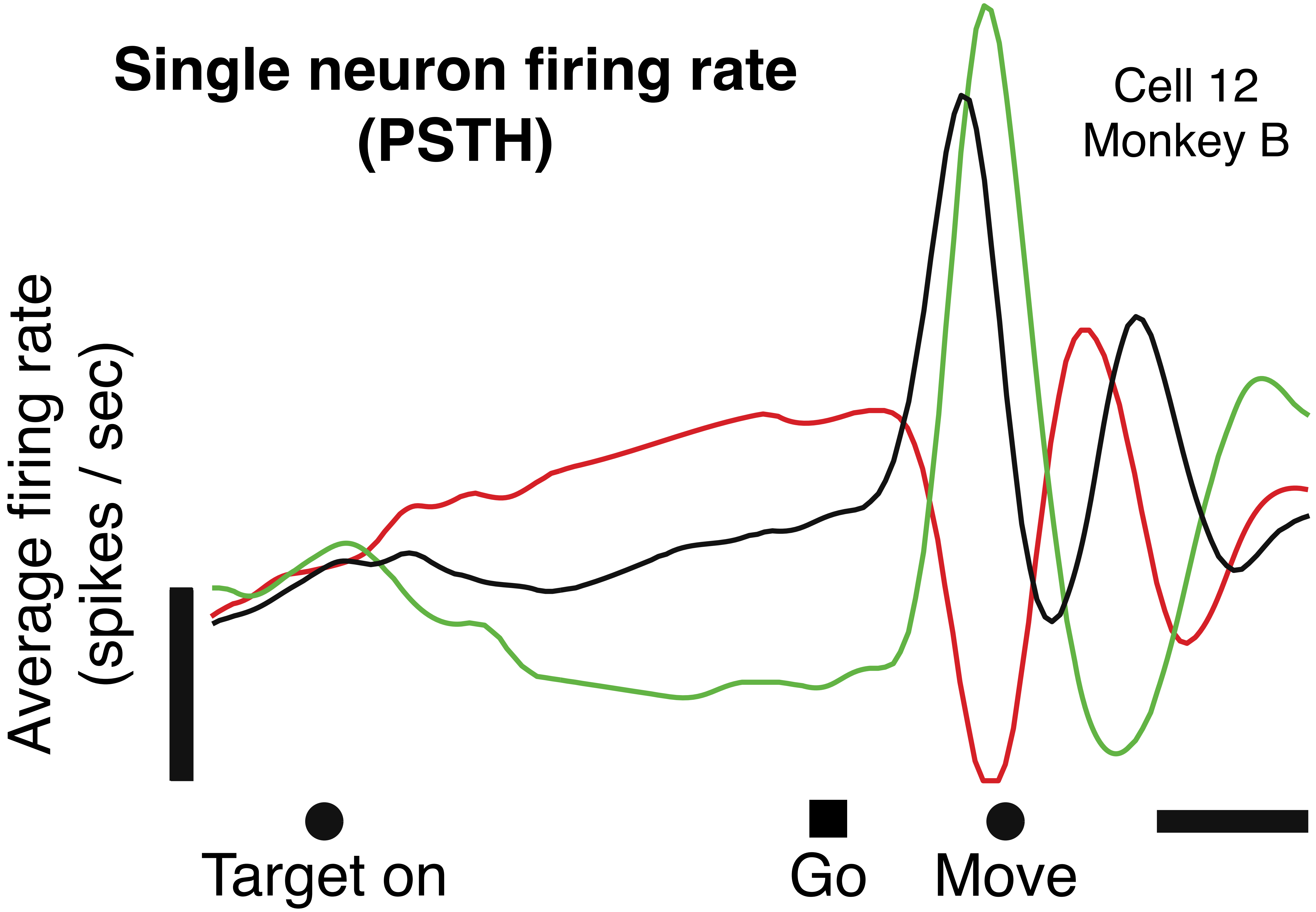
Churchland*, Cunningham* … Shenoy, *Nature* (2012)



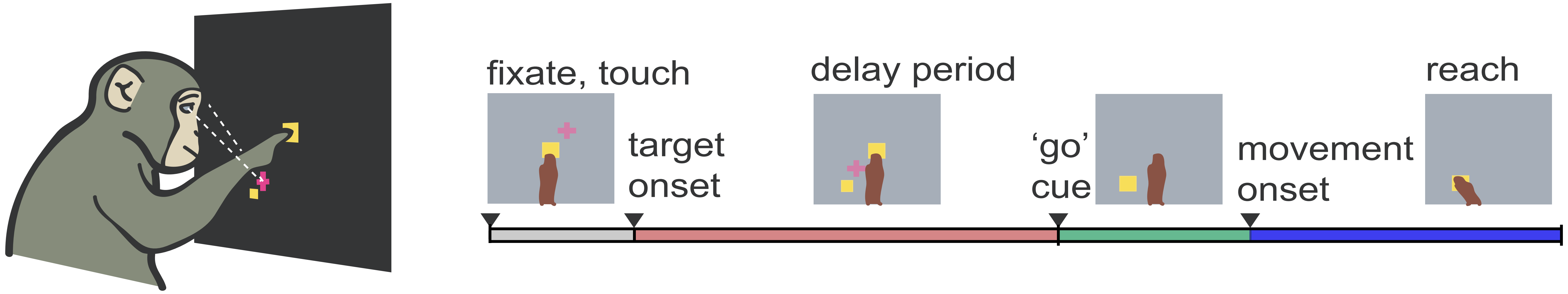
Reach trajectories



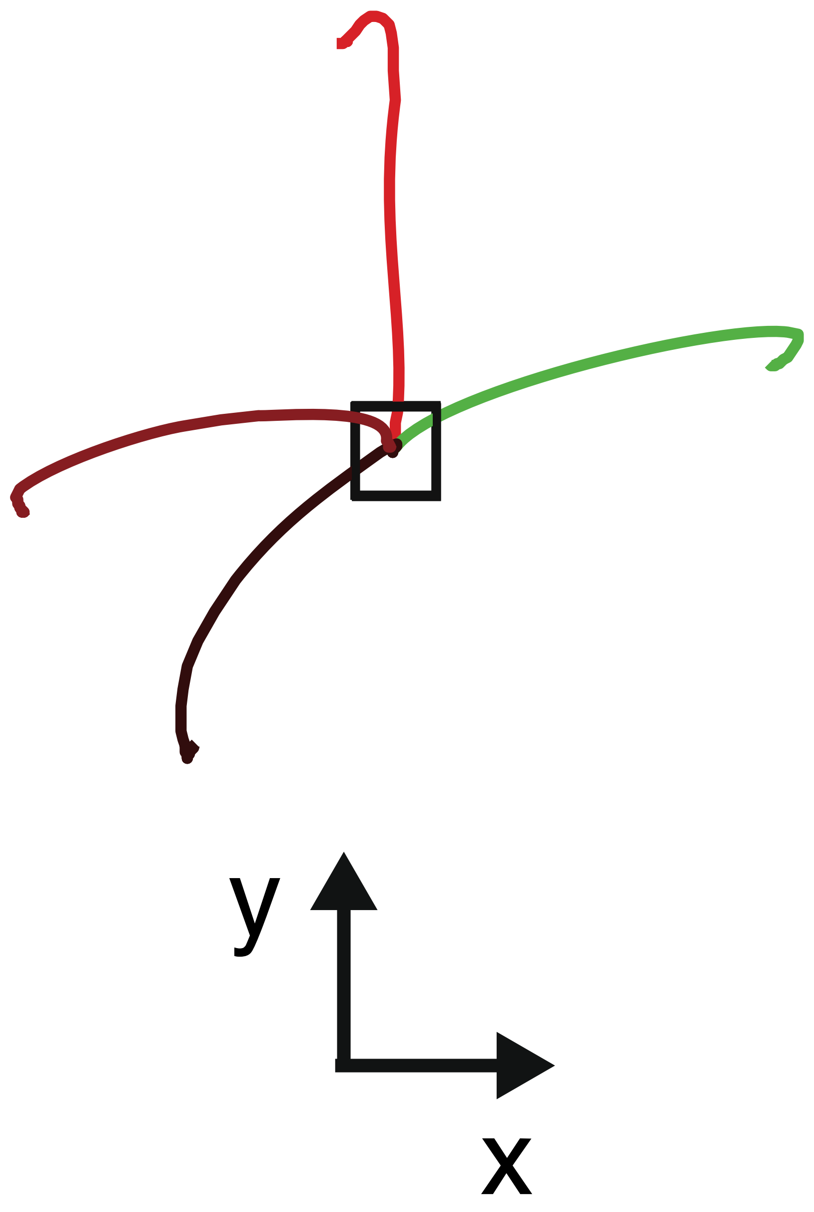
Single neuron firing rate (PSTH)



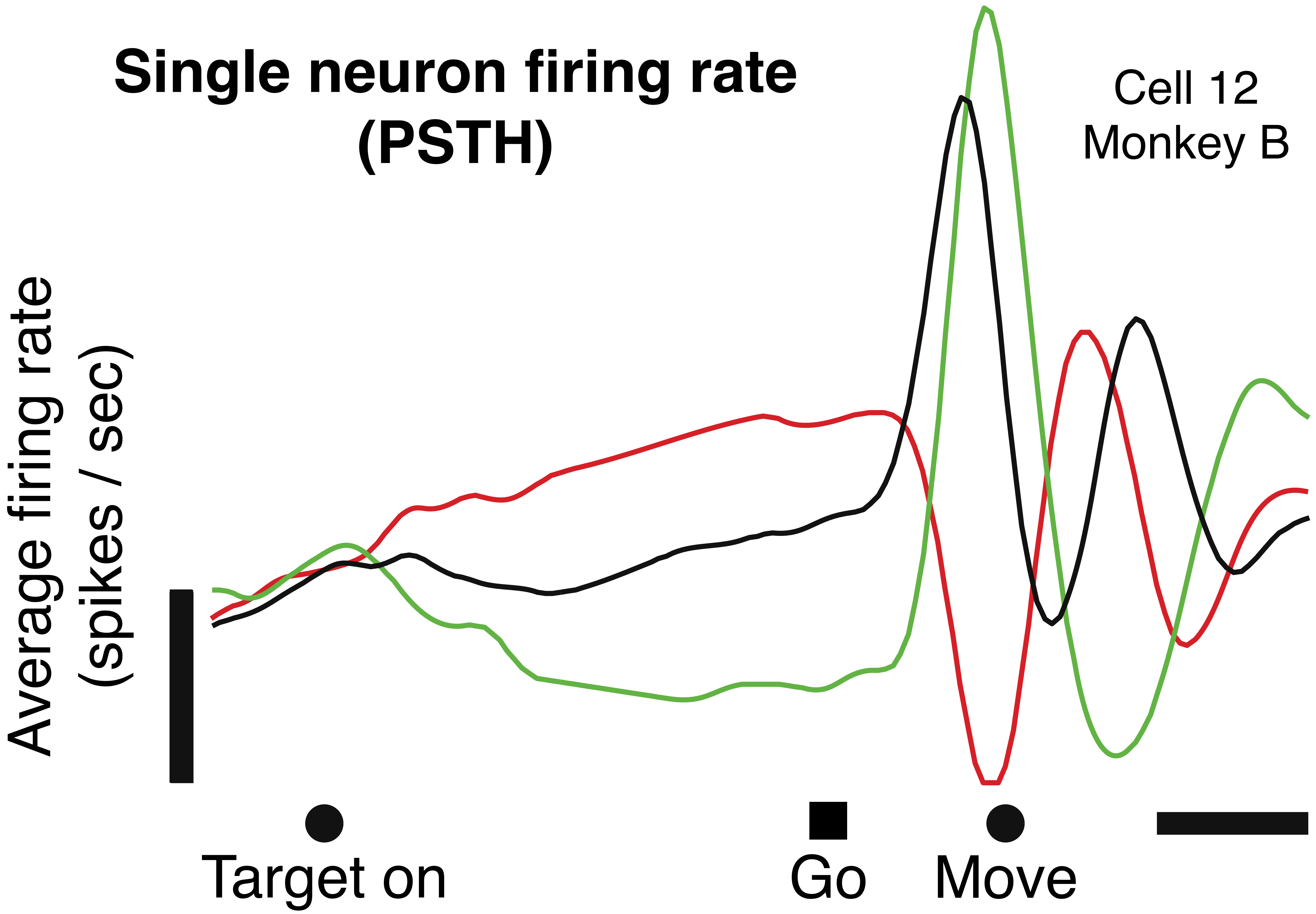
Churchland*, Cunningham* … Shenoy, *Nature* (2012)



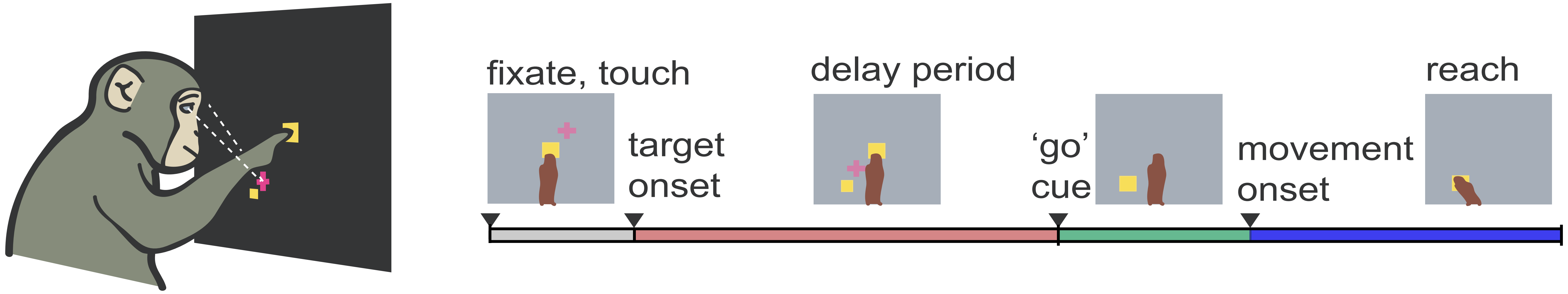
Reach trajectories

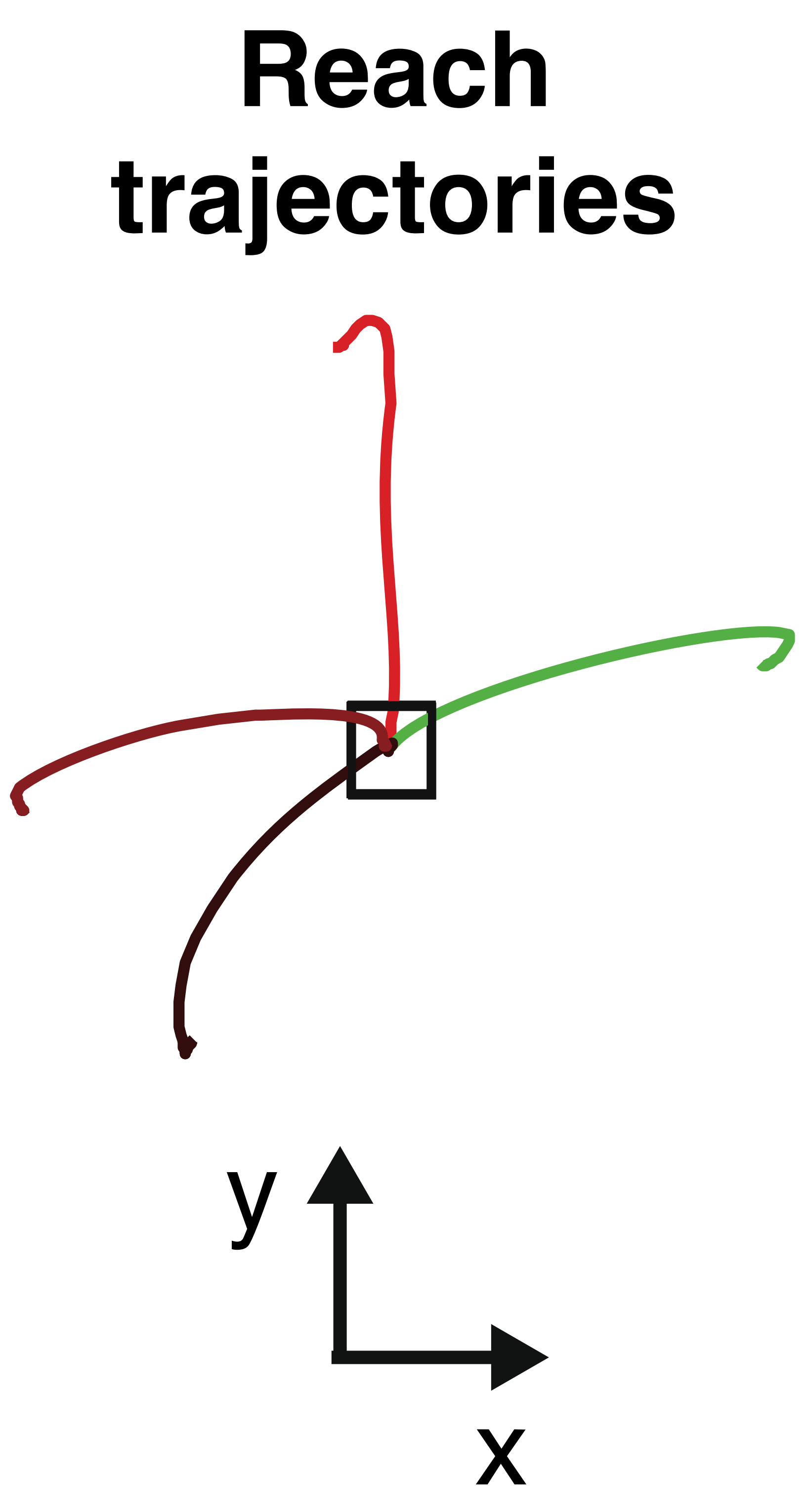


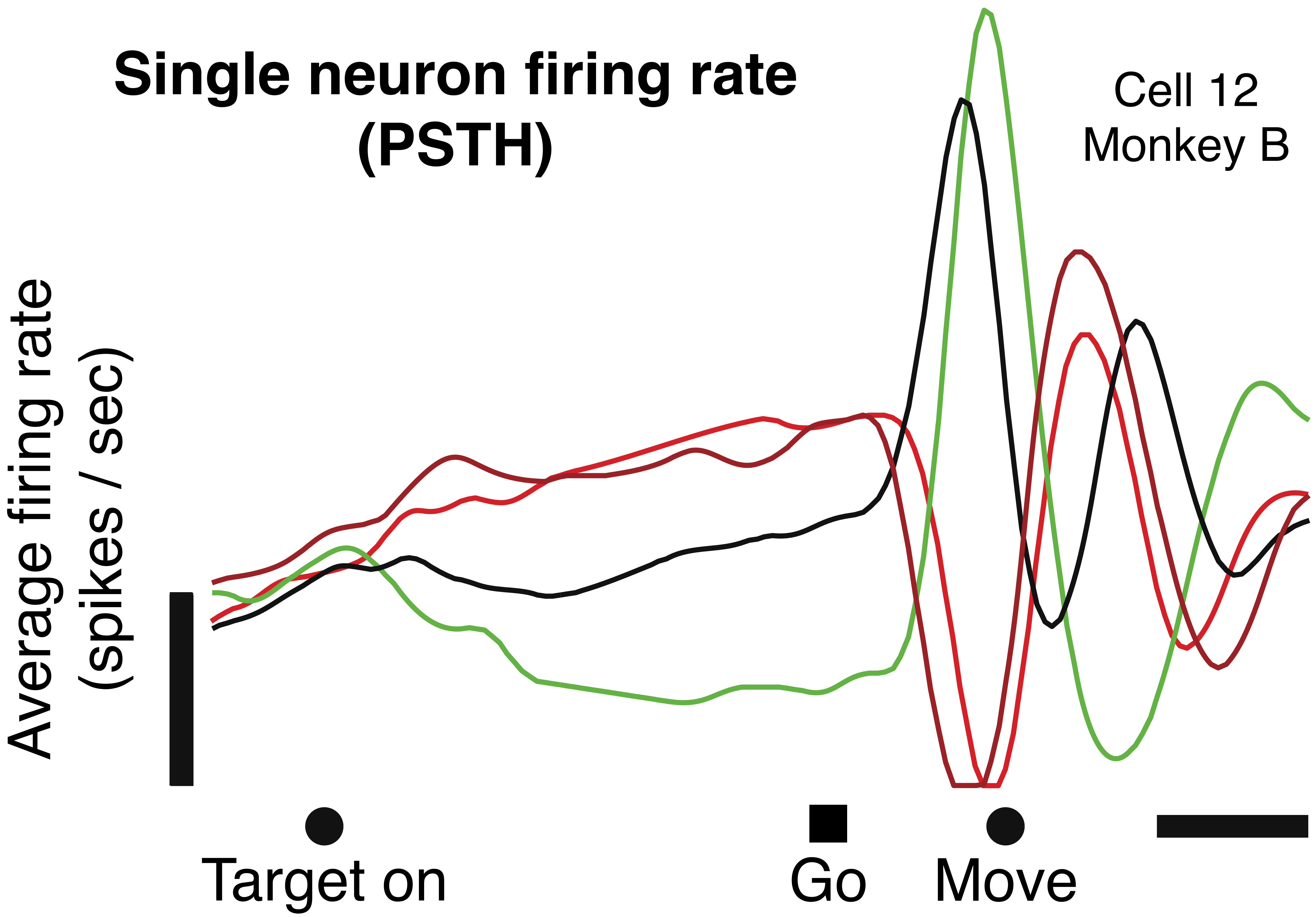
Single neuron firing rate (PSTH)



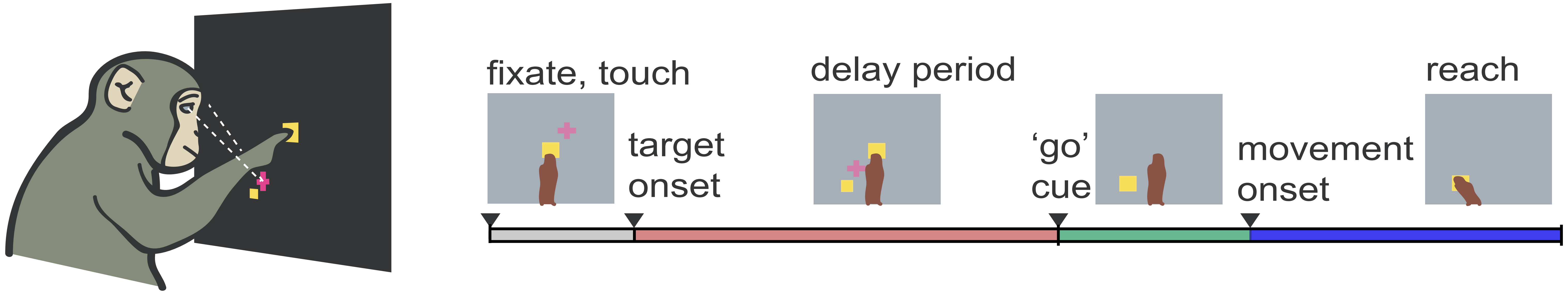
Churchland*, Cunningham* … Shenoy, *Nature* (2012)

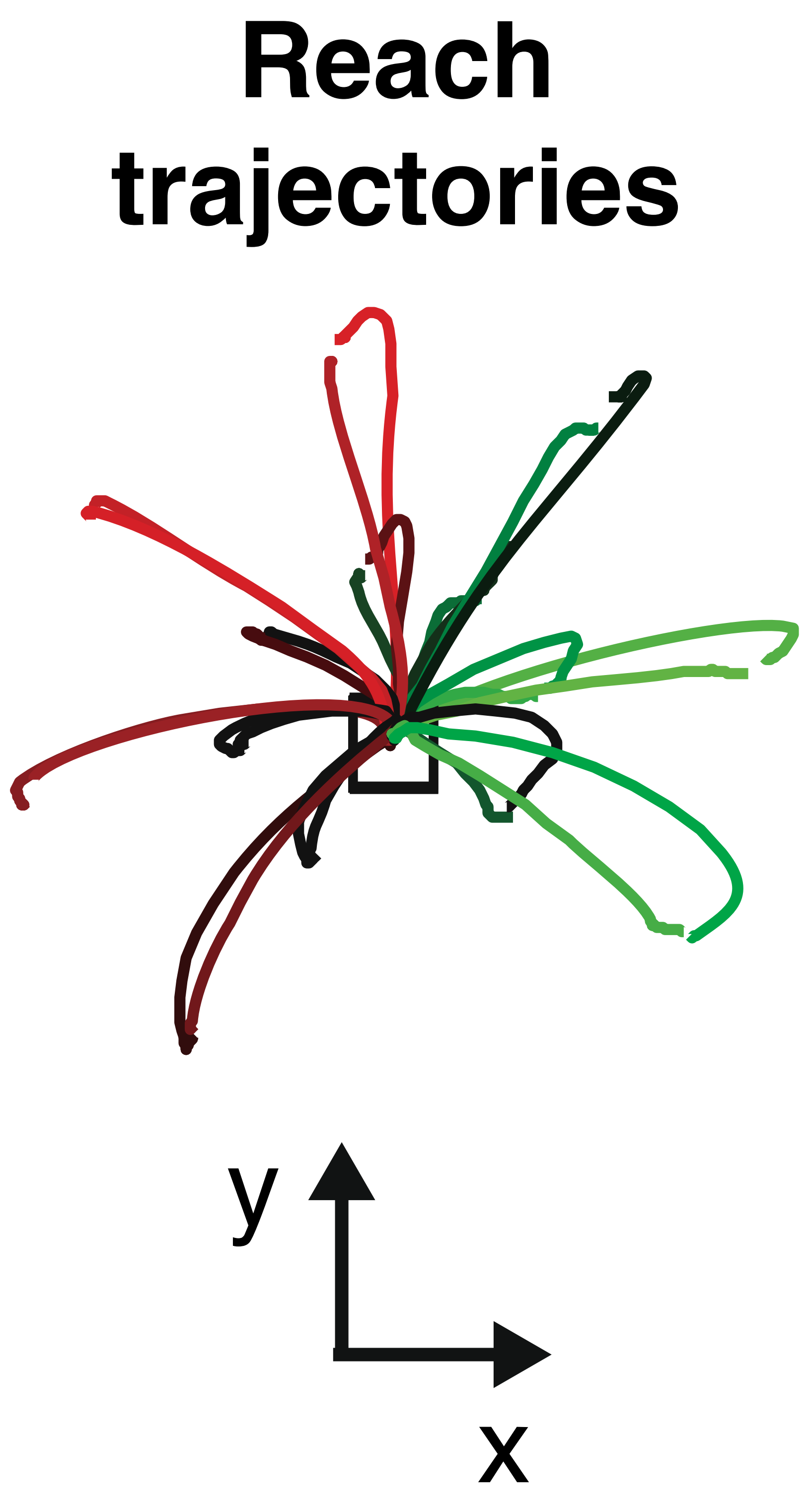


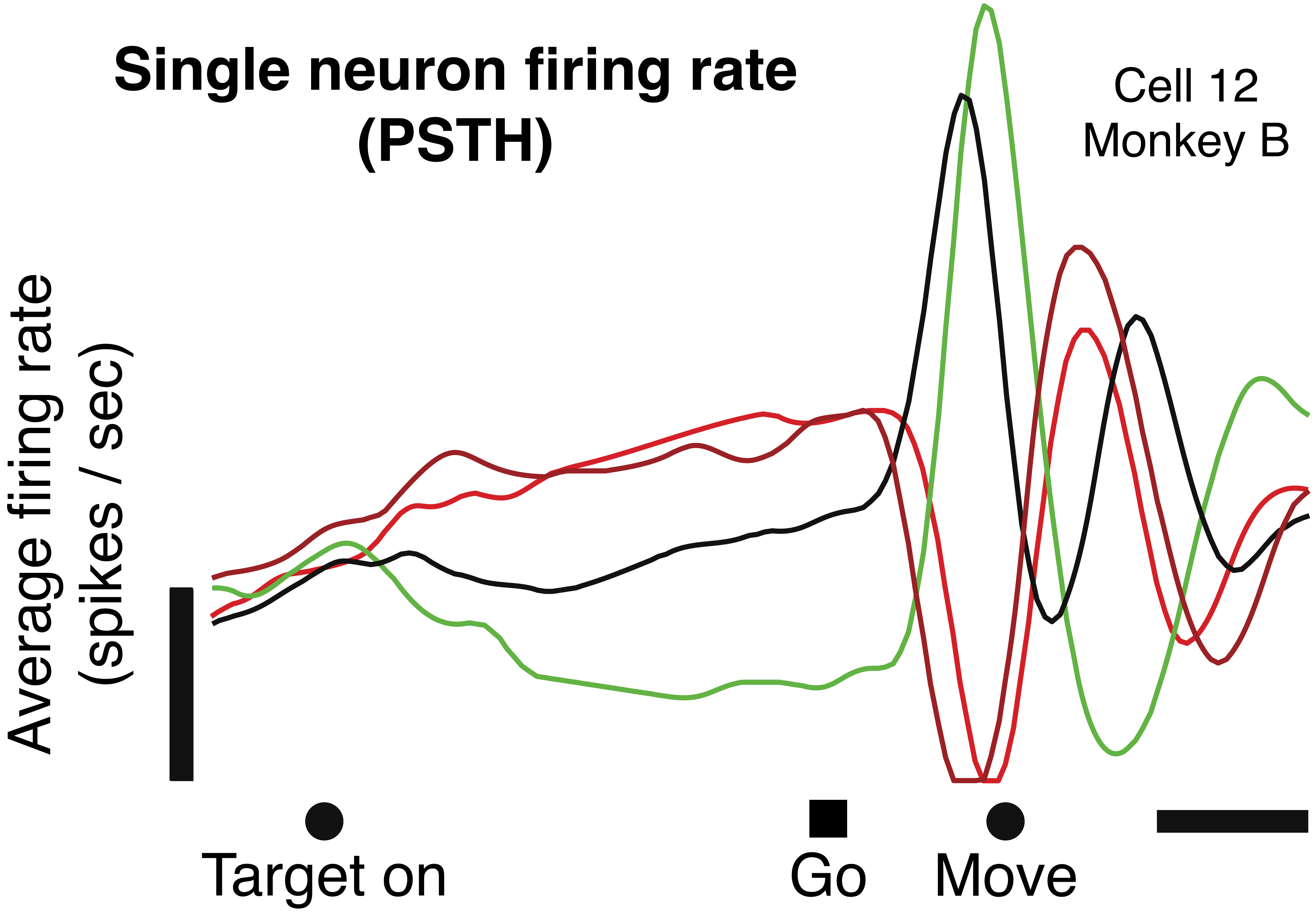




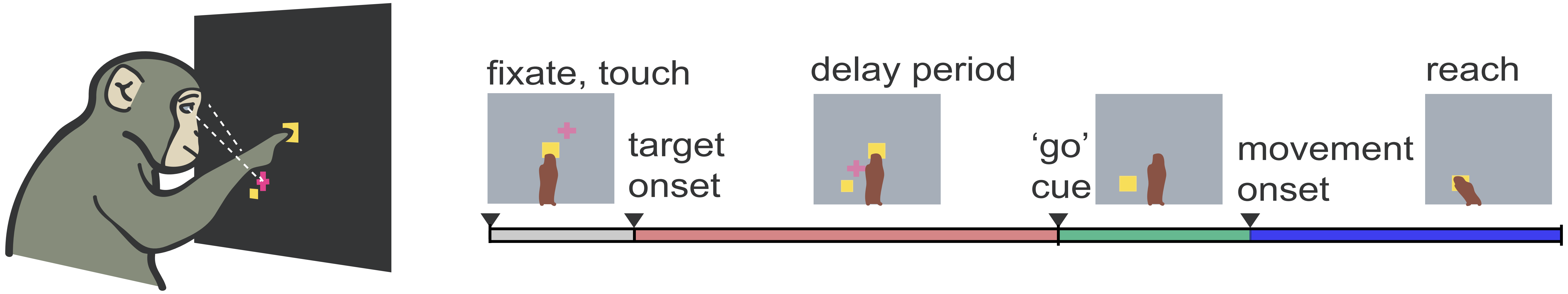
Churchland*, Cunningham* … Shenoy, *Nature* (2012)



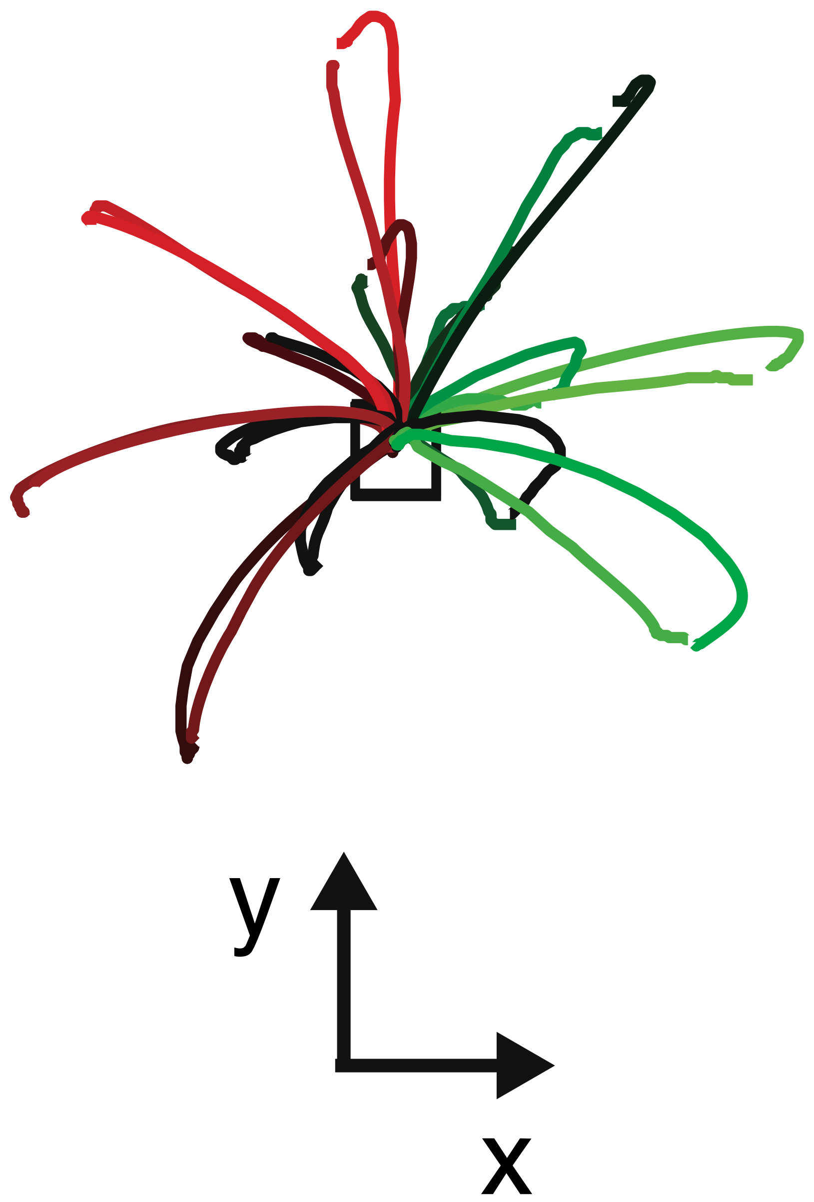




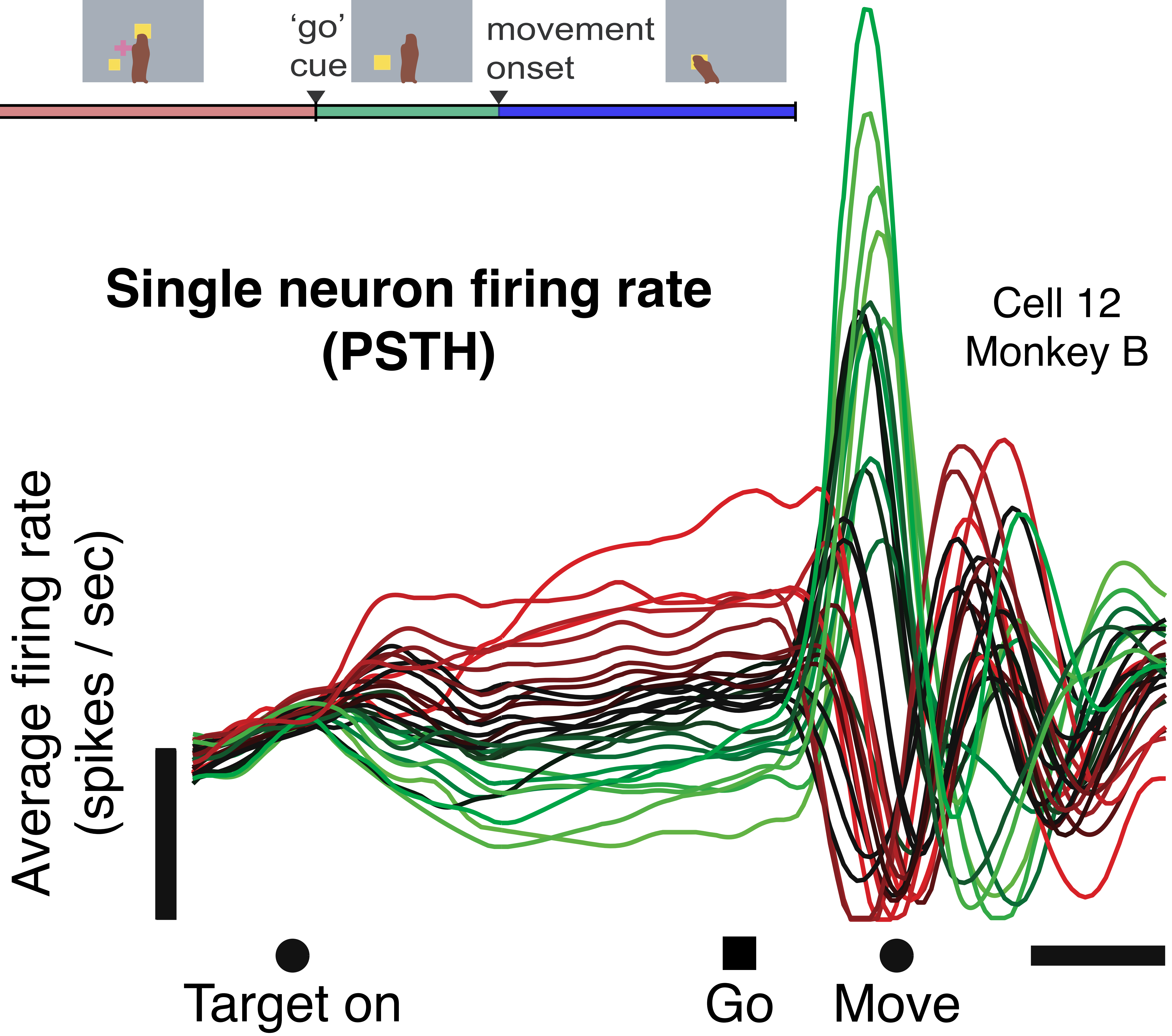
Churchland*, Cunningham* … Shenoy, *Nature* (2012)



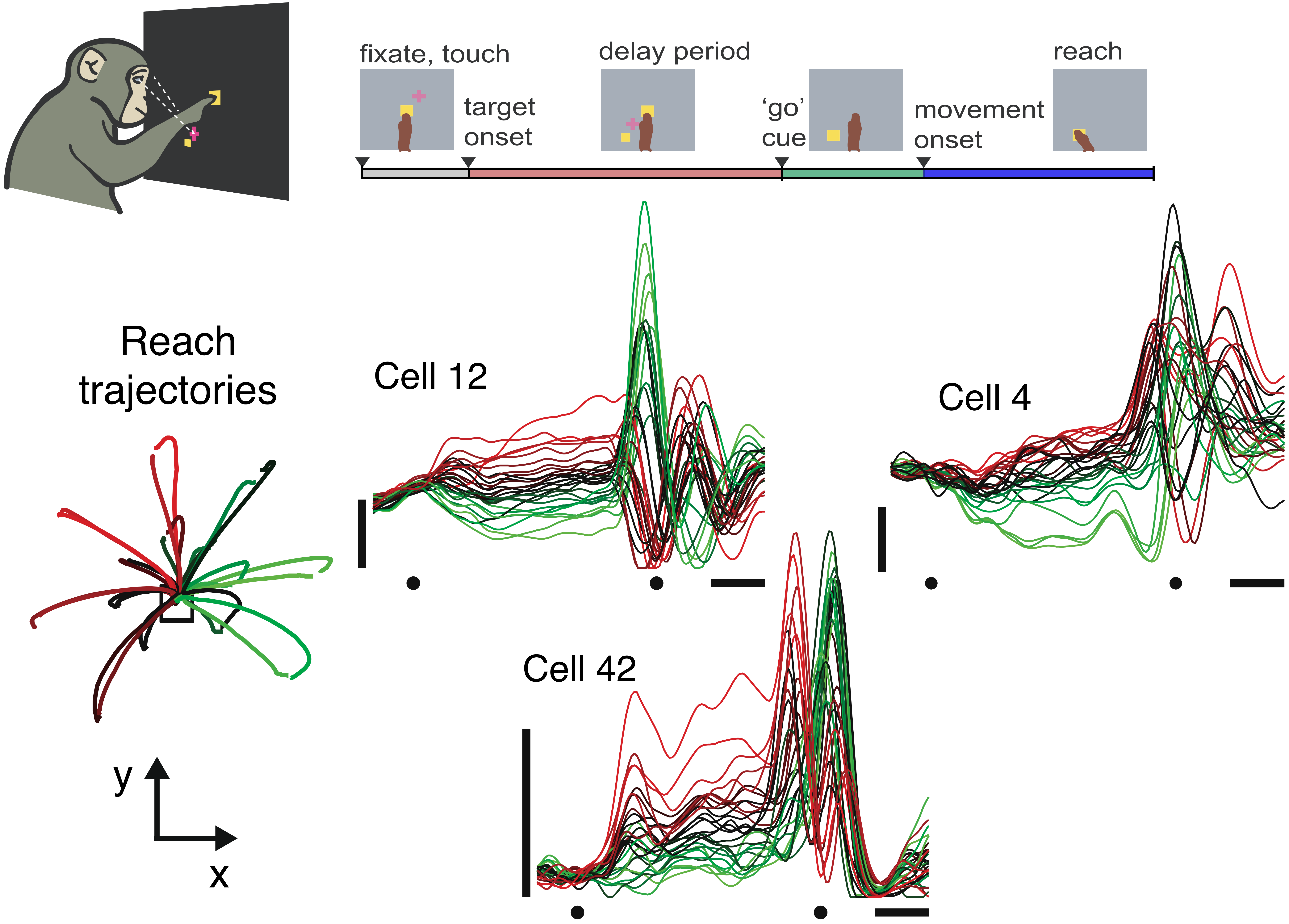
Reach trajectories



Single neuron firing rate (PSTH)

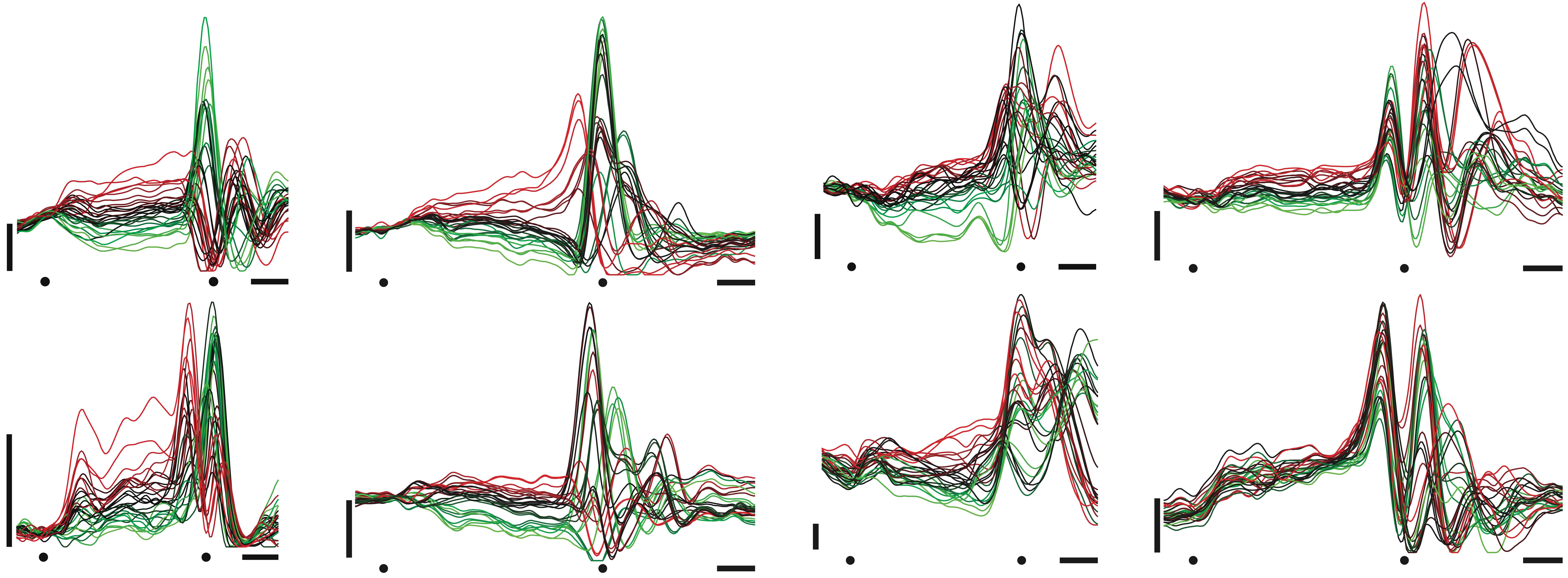


Churchland*, Cunningham* … Shenoy, *Nature* (2012)



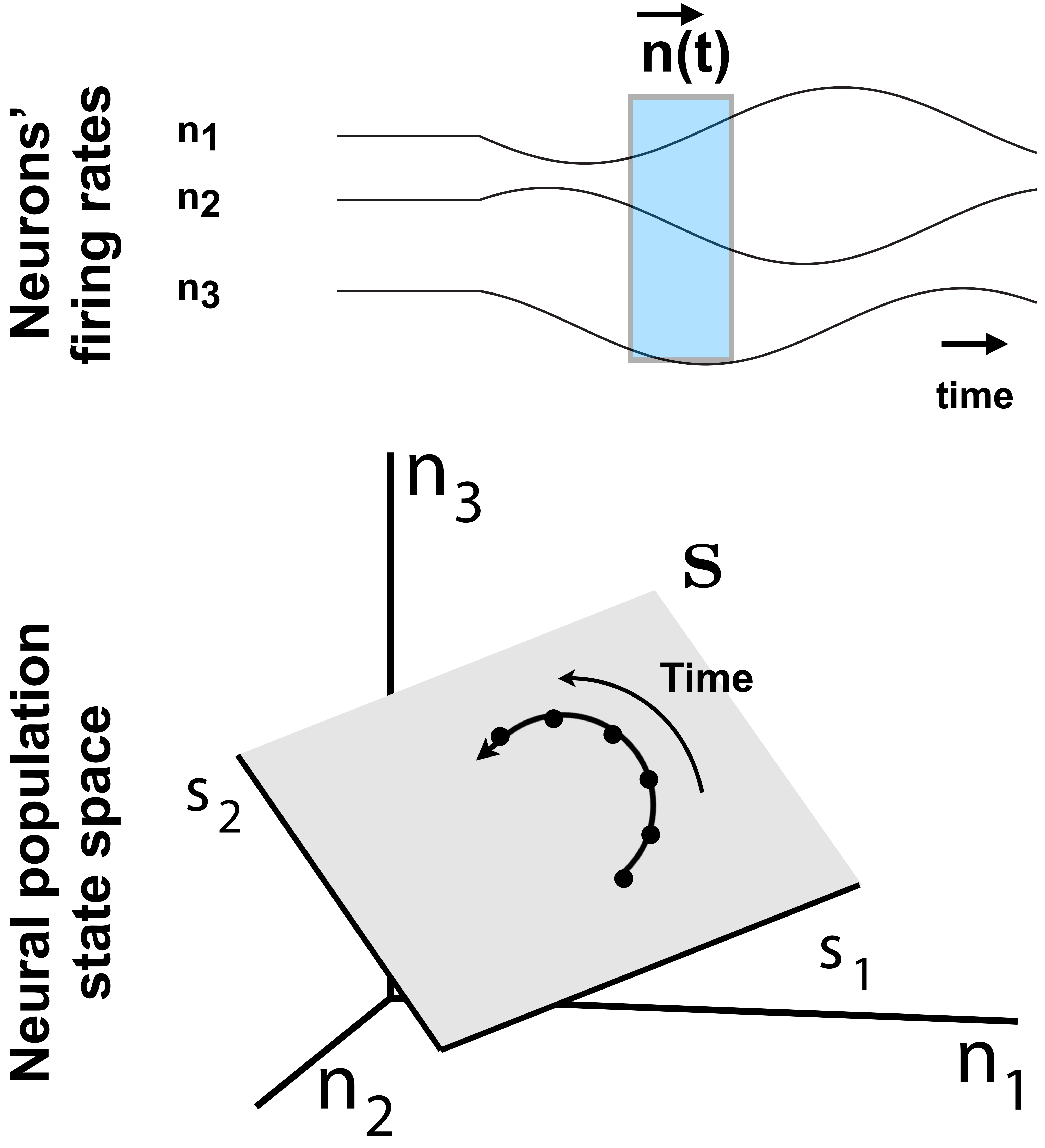
Churchland*, Cunningham* … Shenoy, *Nature* (2012)

Single-neuron data is confusing

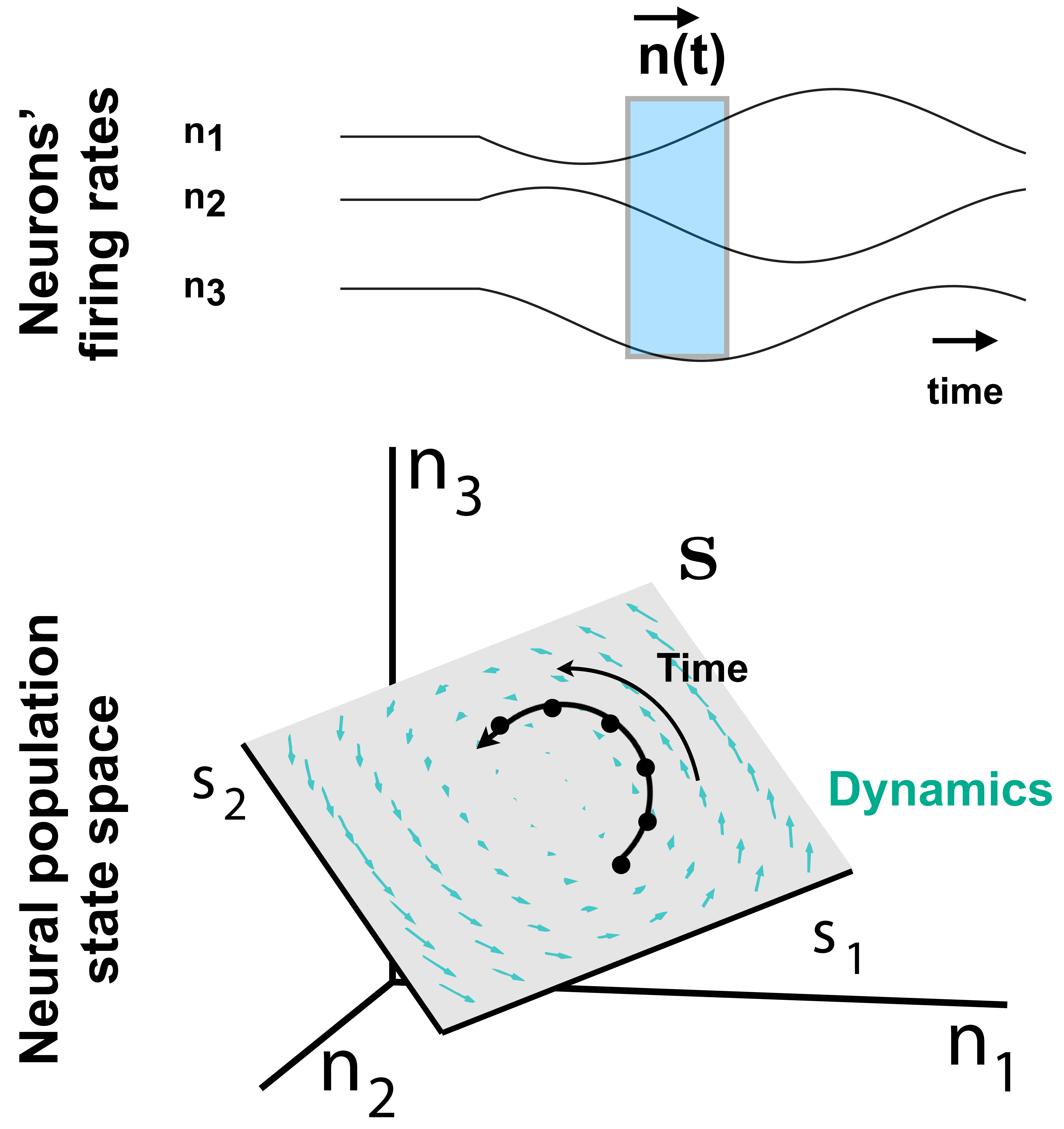


Churchland*, Cunningham* … Shenoy, *Nature* (2012)

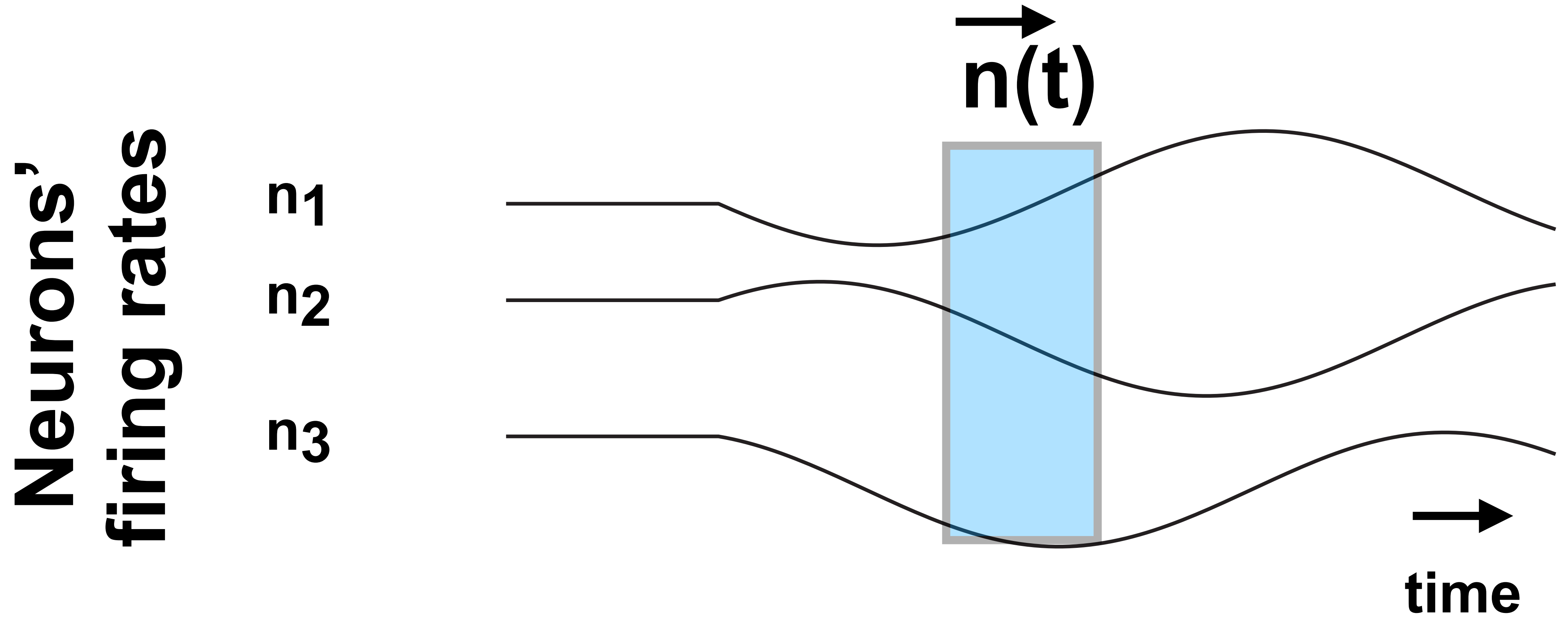
Uncovering neural population dynamics



Uncovering neural population dynamics

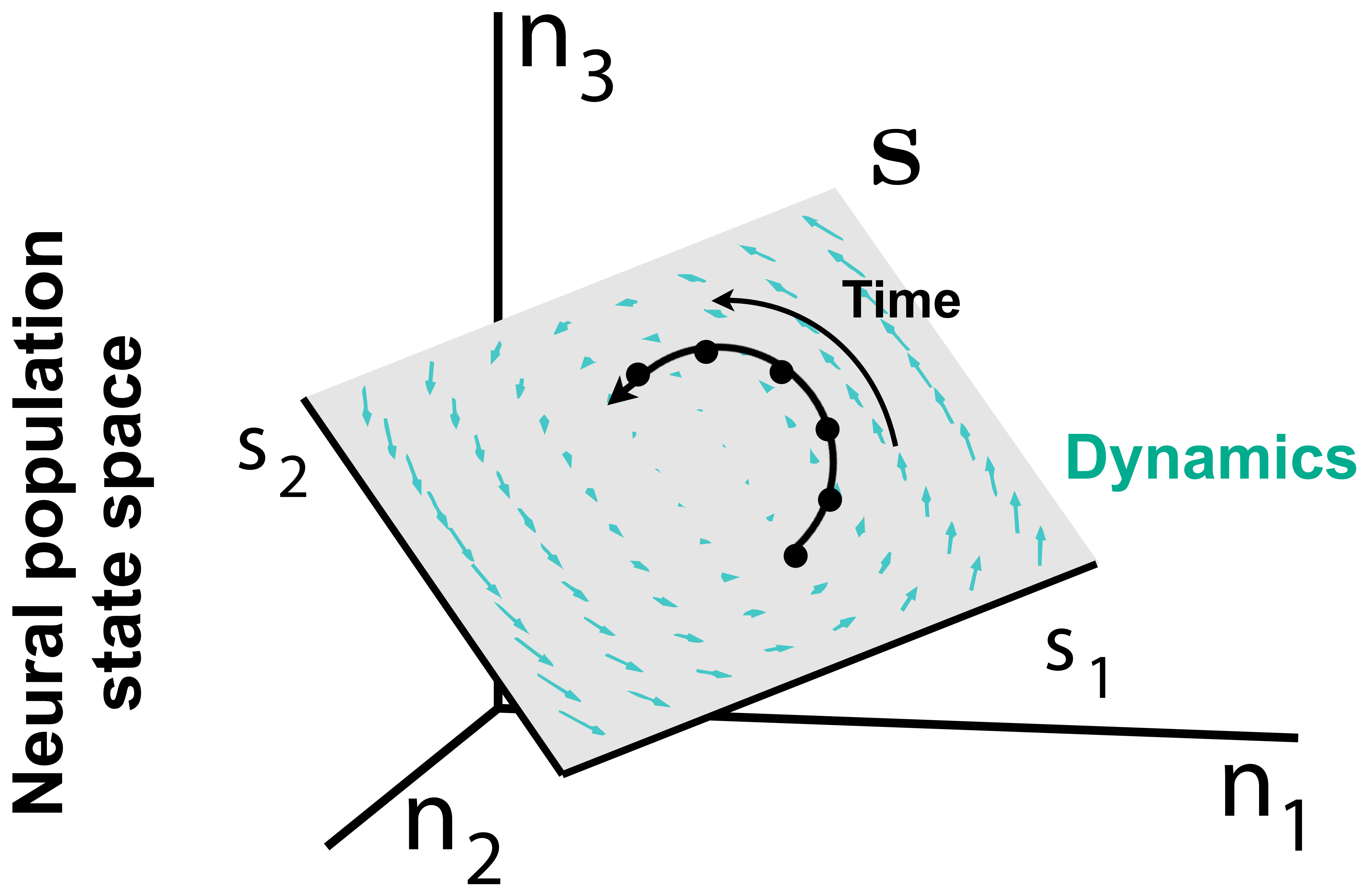


Uncovering neural population dynamics



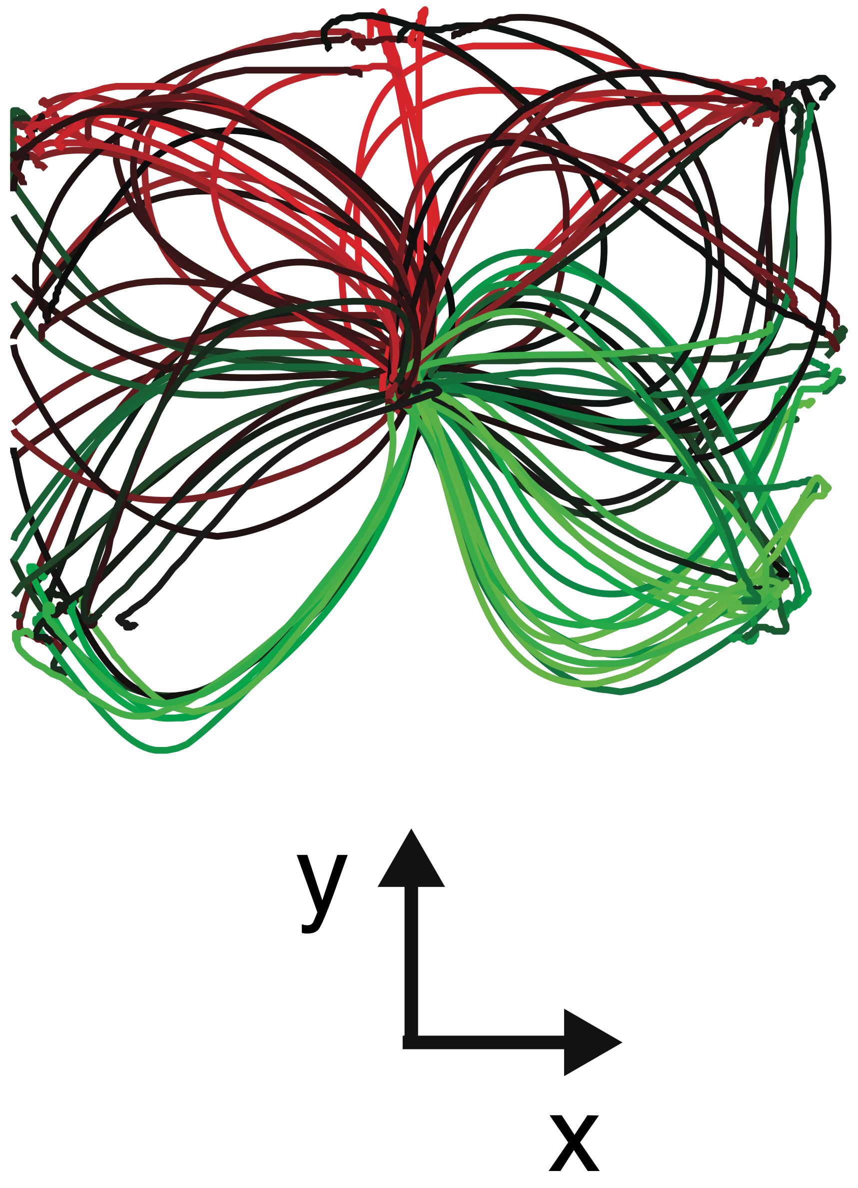
Predictable activity - modeled by *autonomous* dynamics

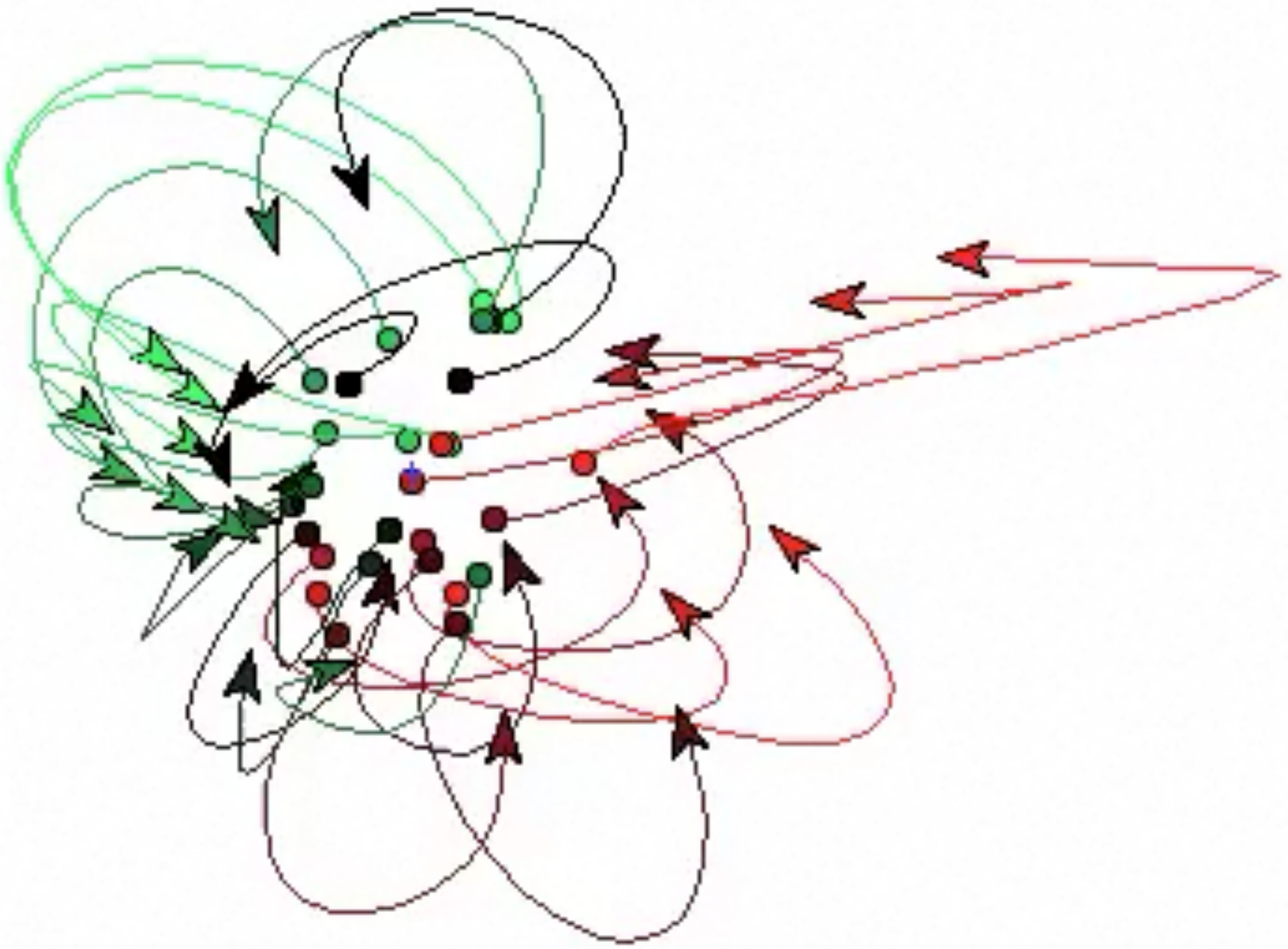
$$\frac{d\mathrm{s}}{dt} = f(\mathrm{s})$$



PCA / jPCA reveals underlying structure

Reach trajectories

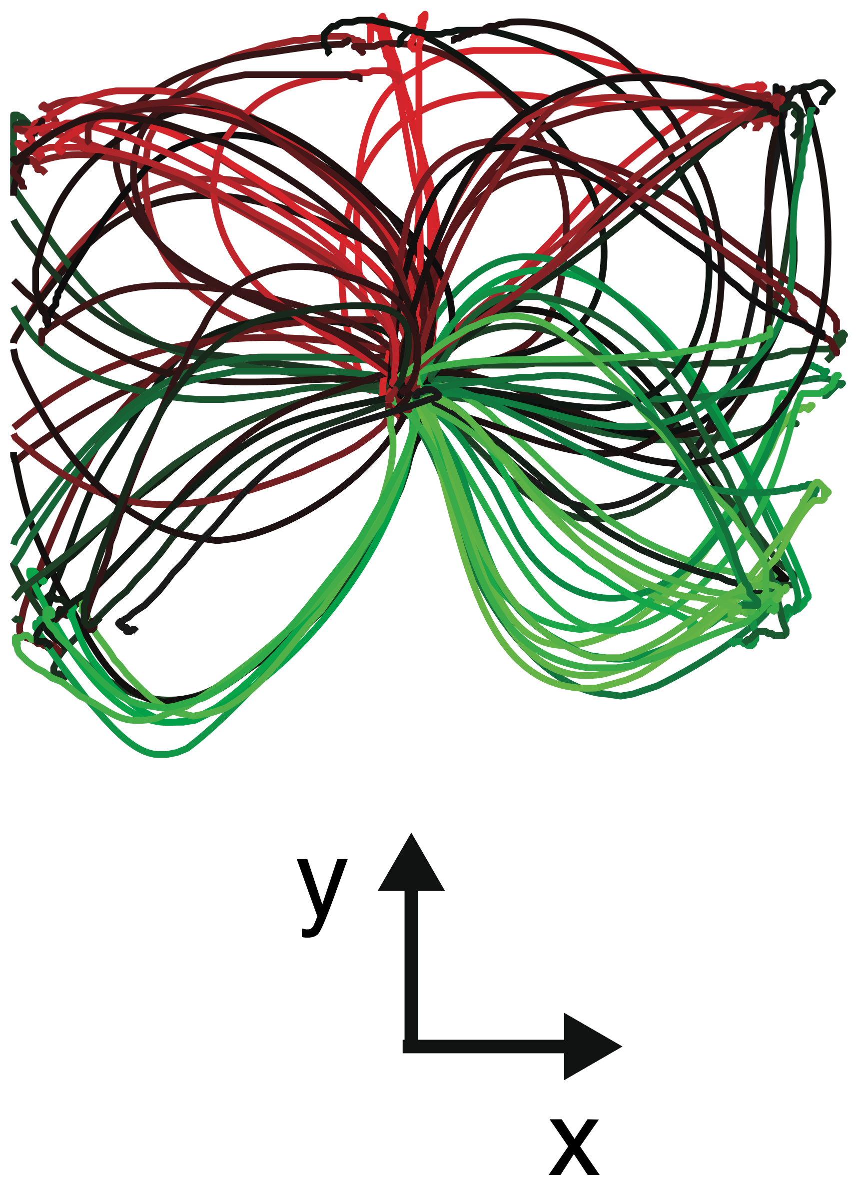


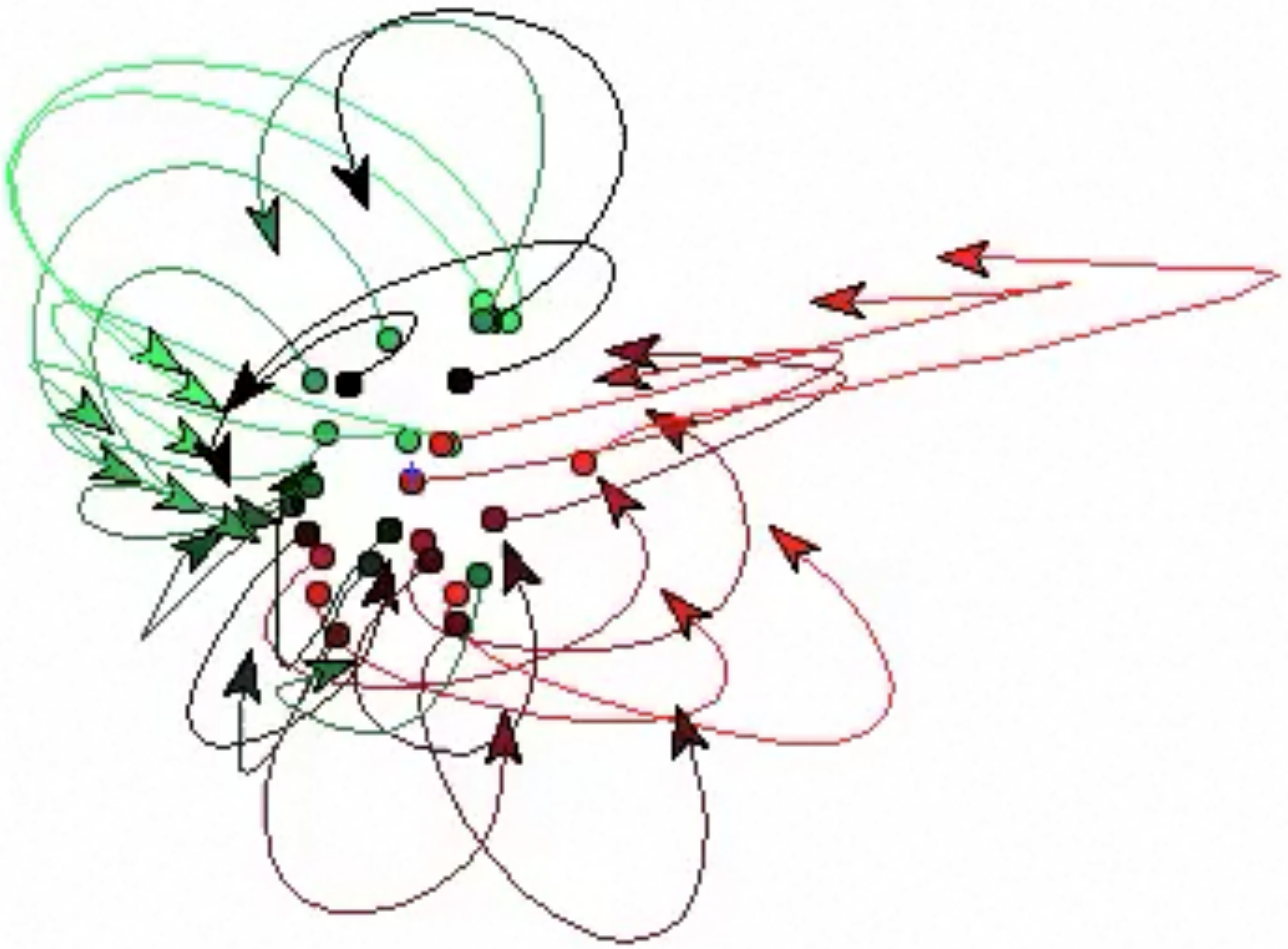


Churchland*, Cunningham* … Shenoy, *Nature* (2012)

PCA / jPCA reveals underlying structure

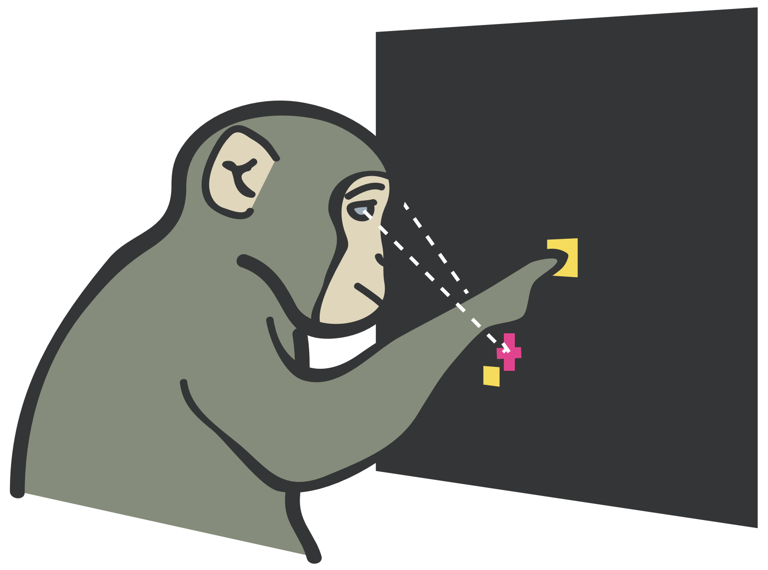
Reach
trajectories



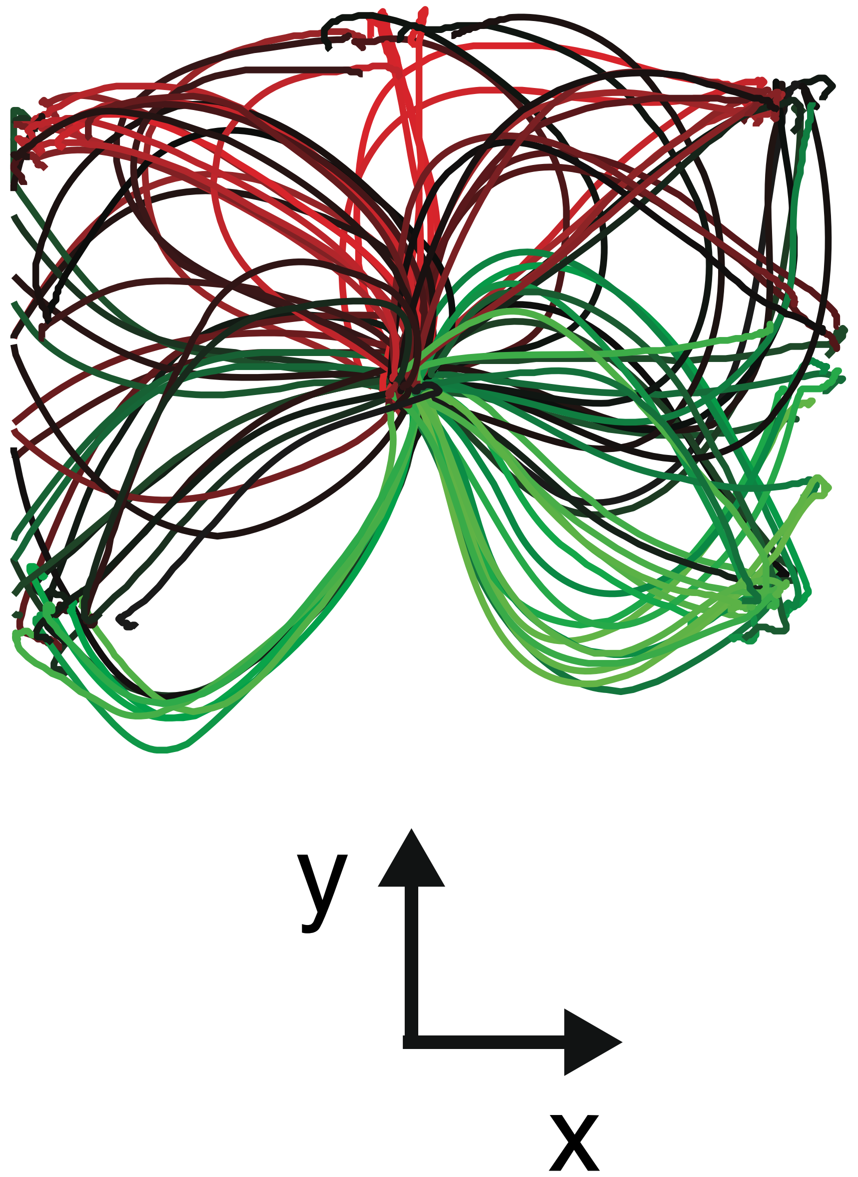


Churchland*, Cunningham* … Shenoy, *Nature* (2012)

Consistent rotational dynamics

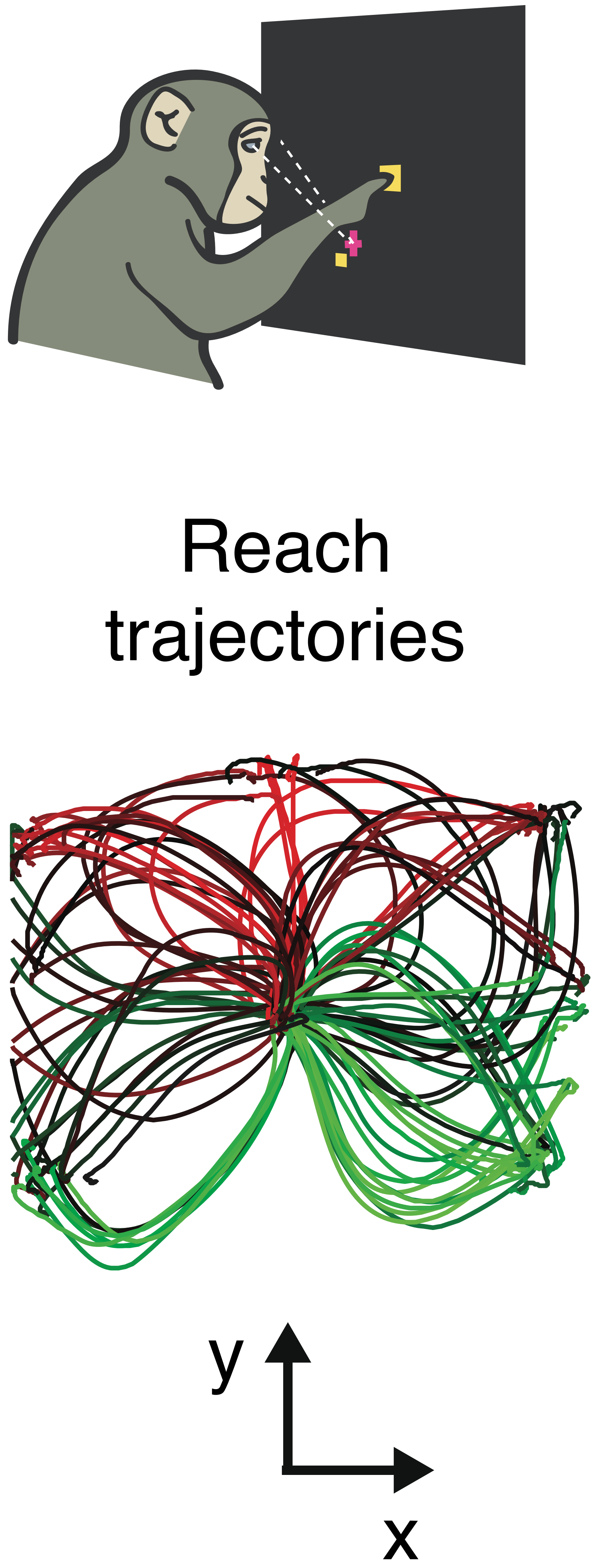


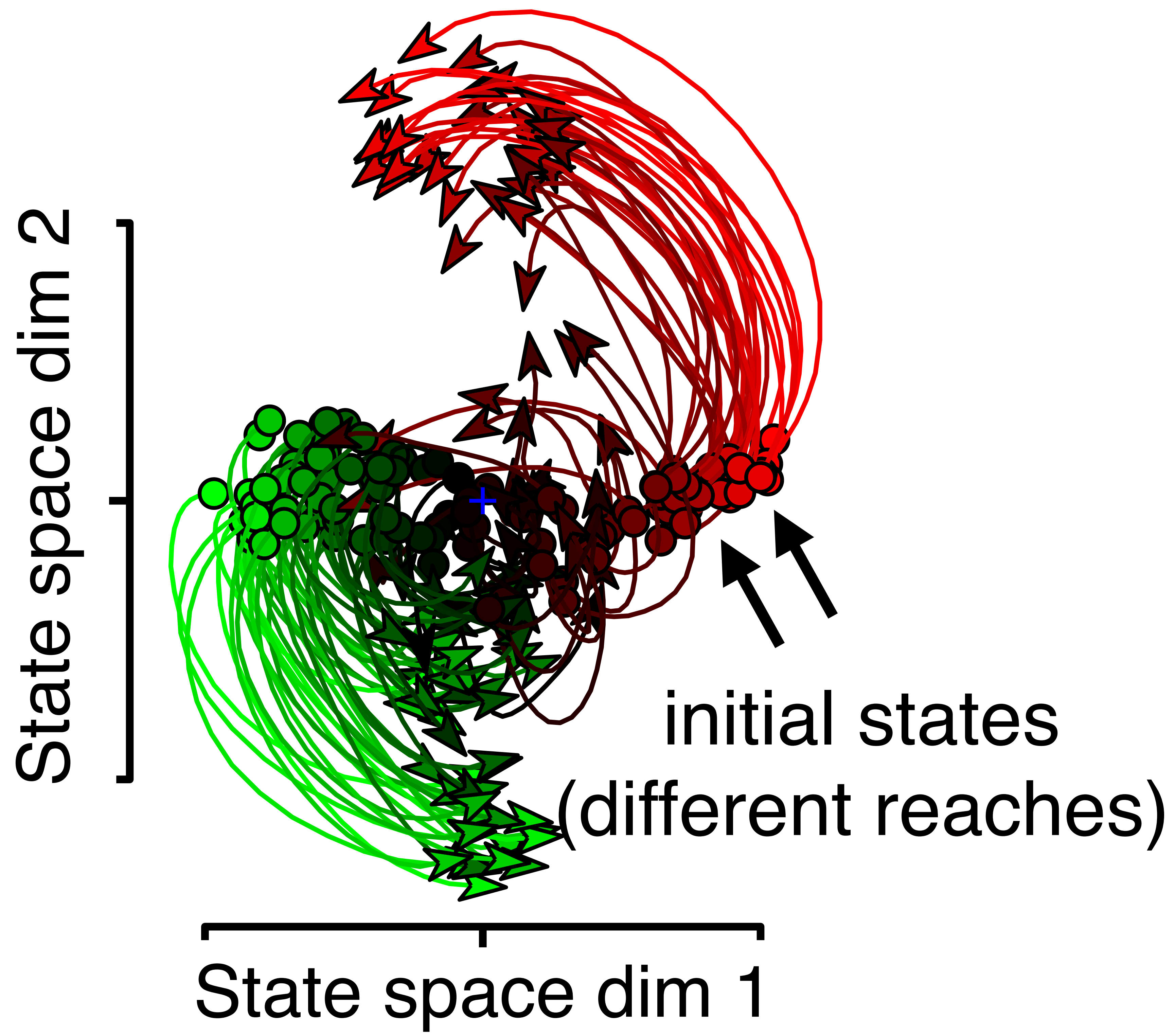
Reach
trajectories



Churchland*, Cunningham* … Shenoy, *Nature* (2012)

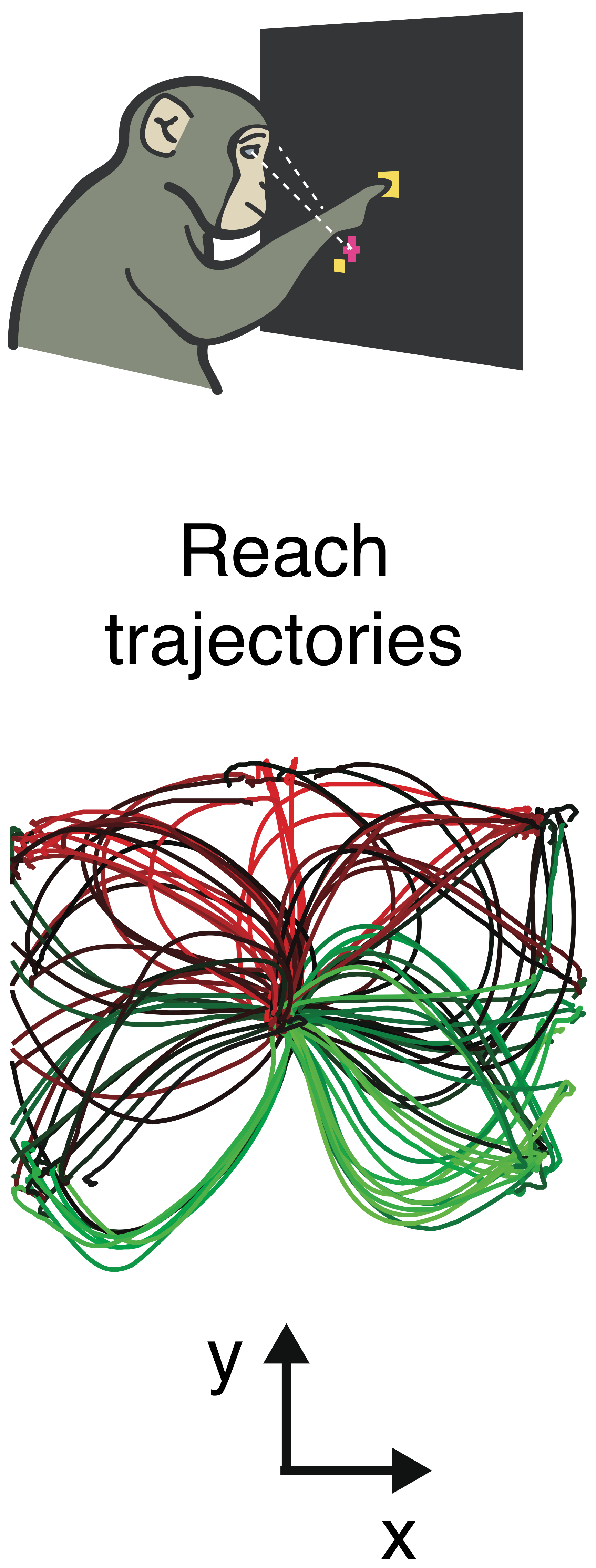
Consistent rotational dynamics

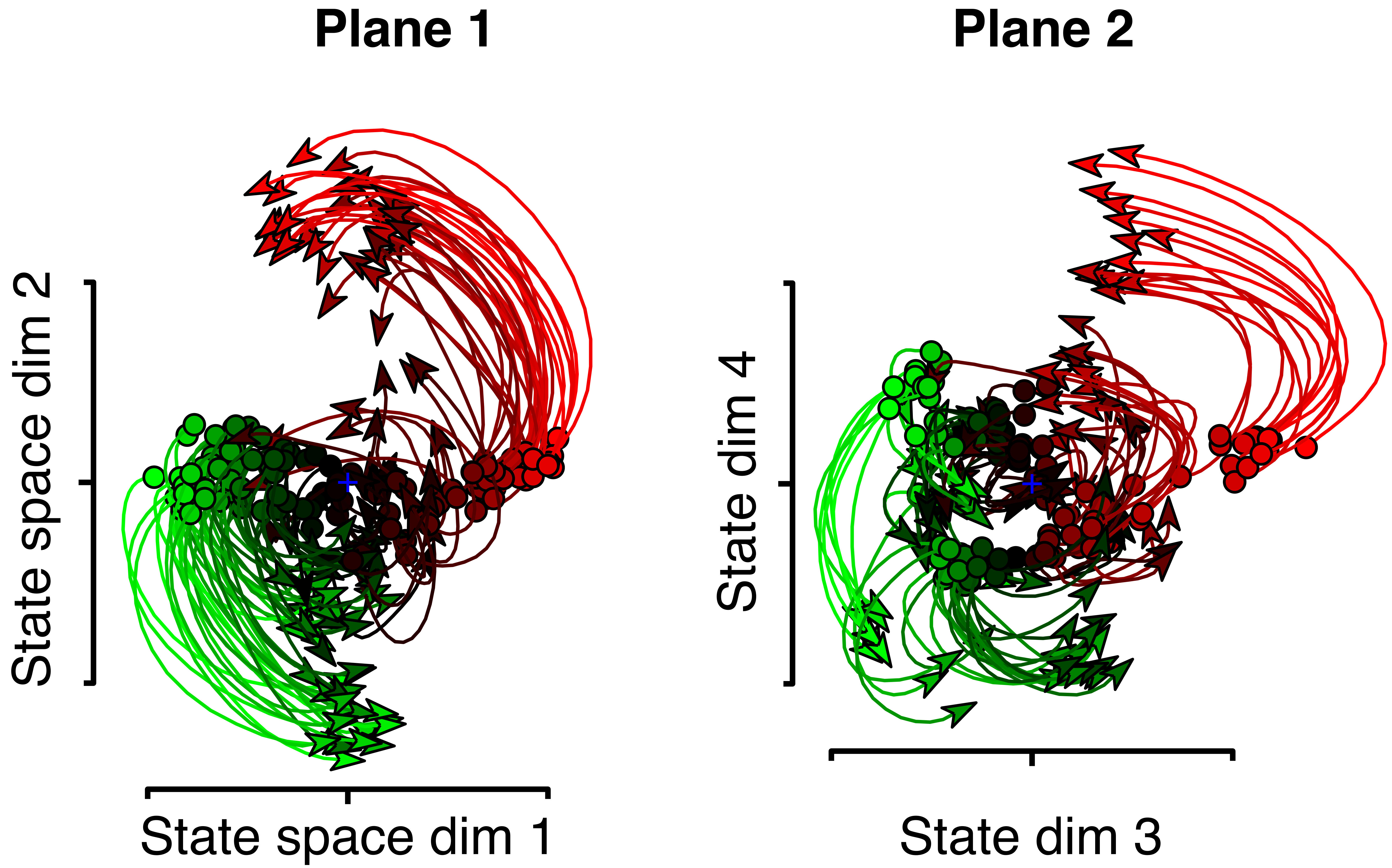




Churchland*, Cunningham* … Shenoy, *Nature* (2012)

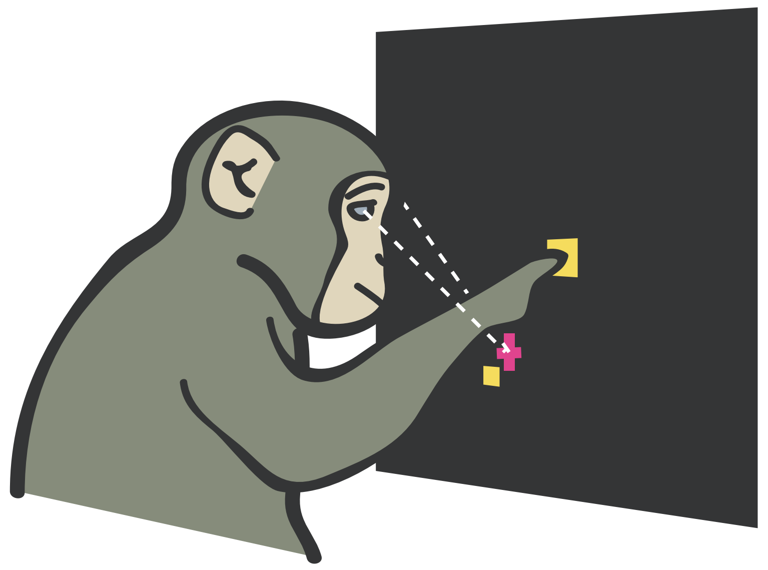
Consistent rotational dynamics



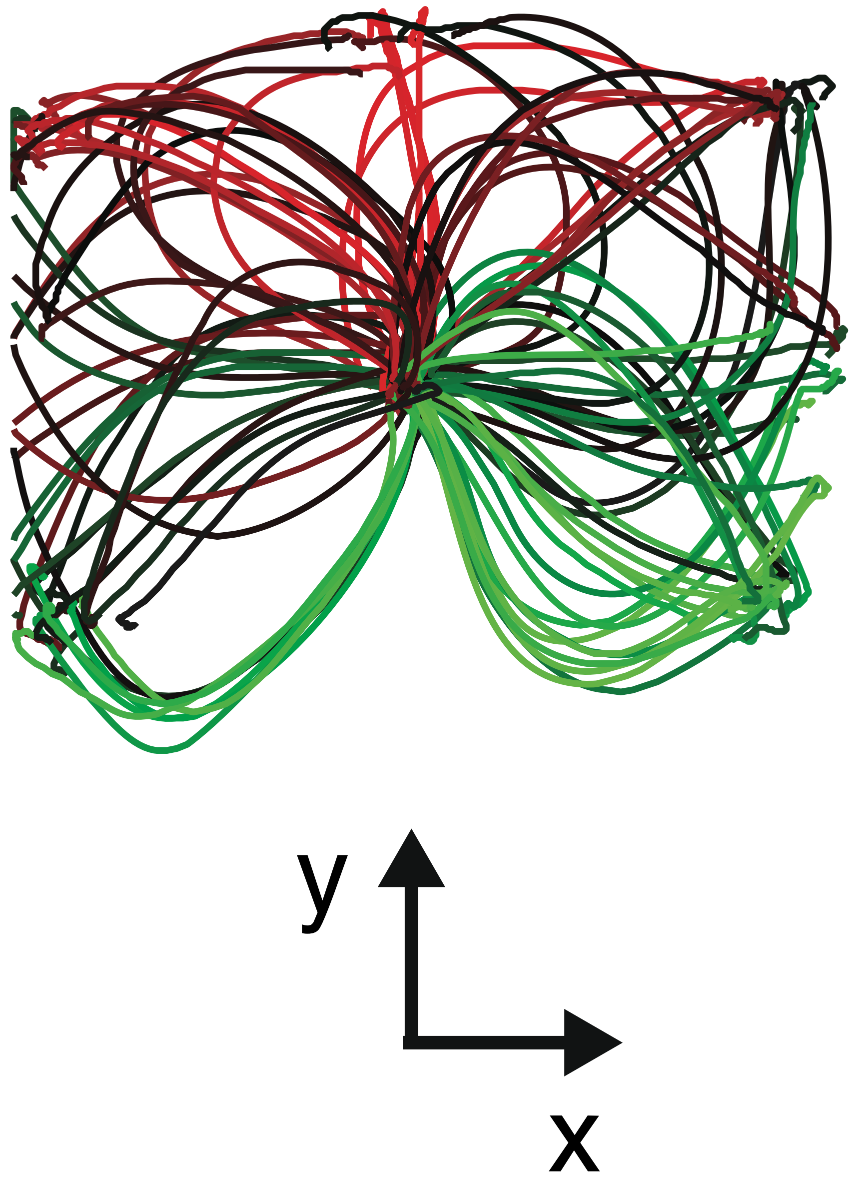


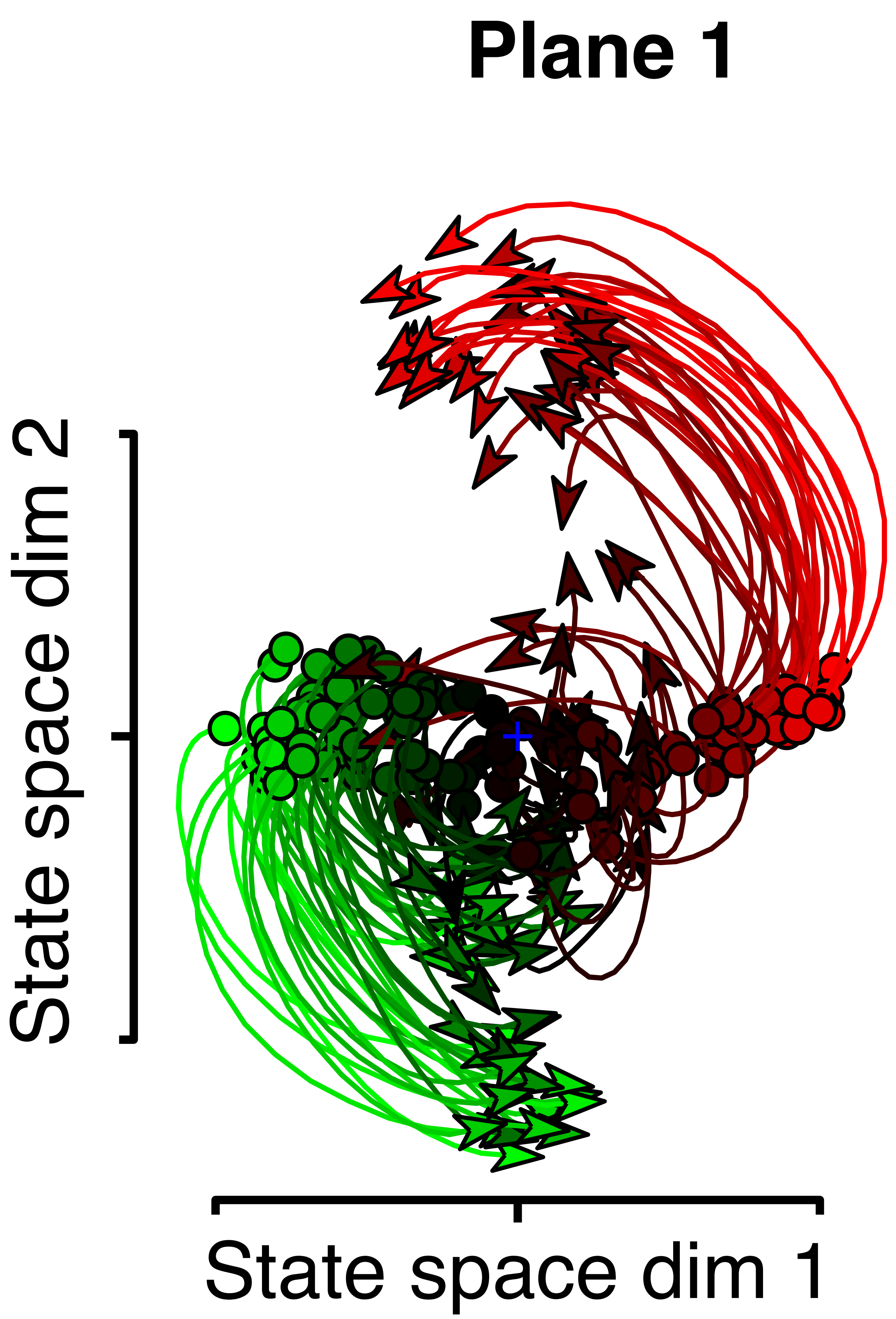
Churchland*, Cunningham* … Shenoy, *Nature* (2012)

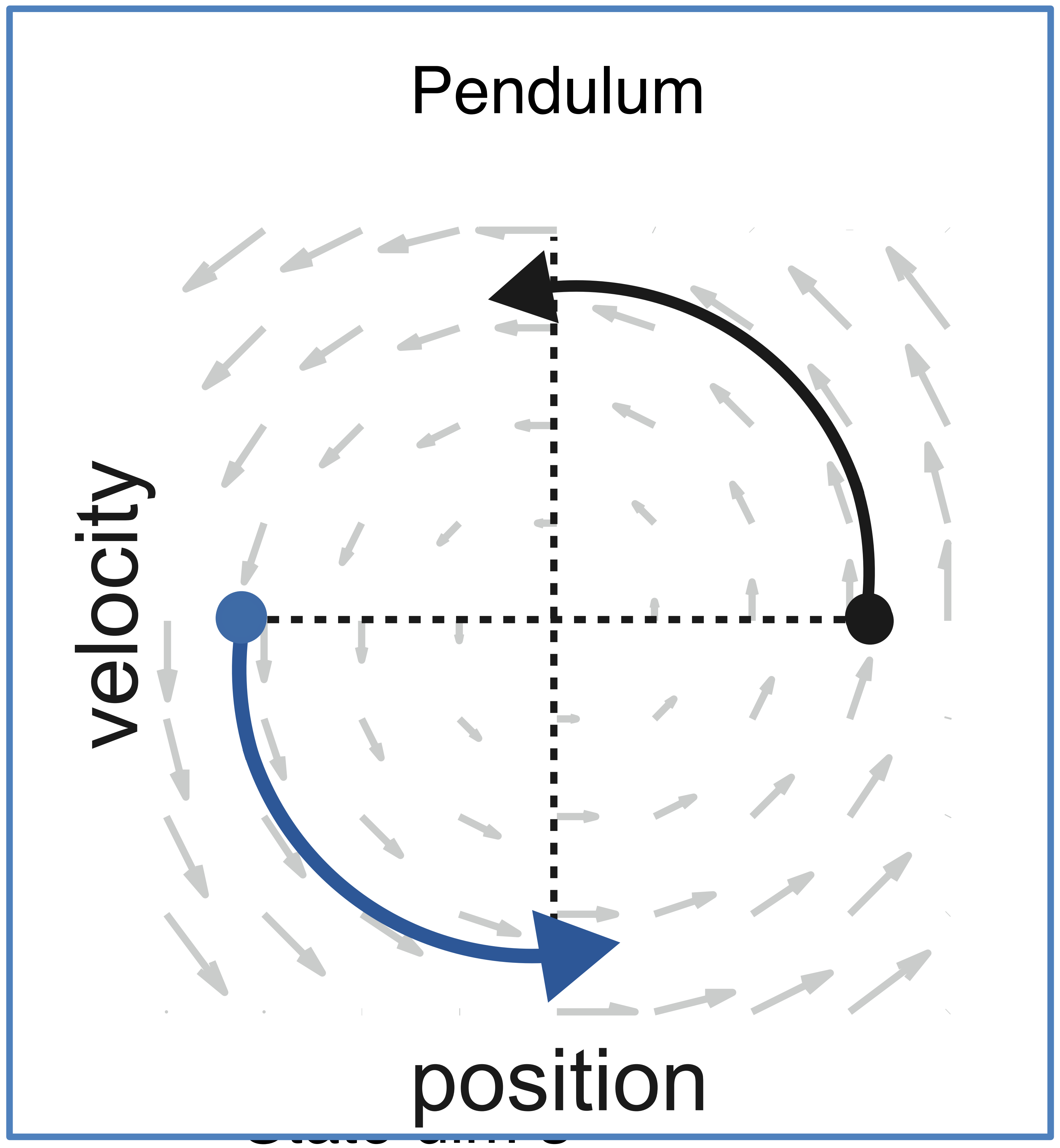
Consistent rotational dynamics



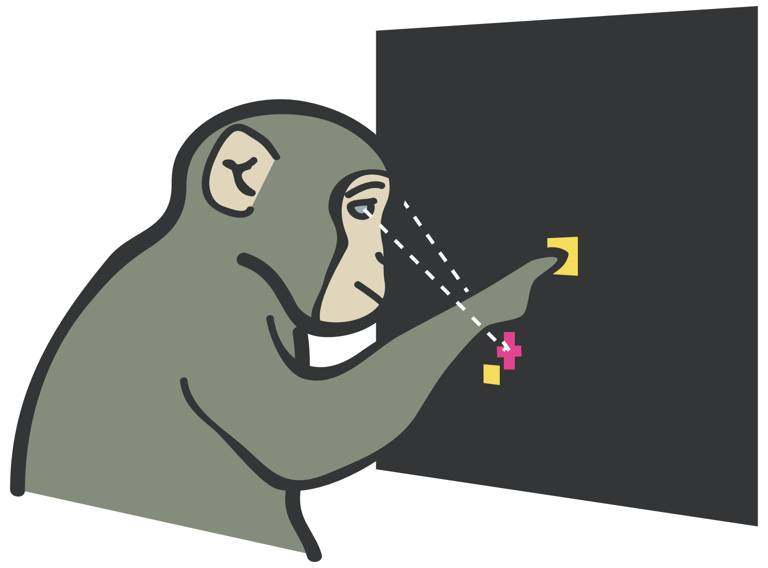
Reach trajectories



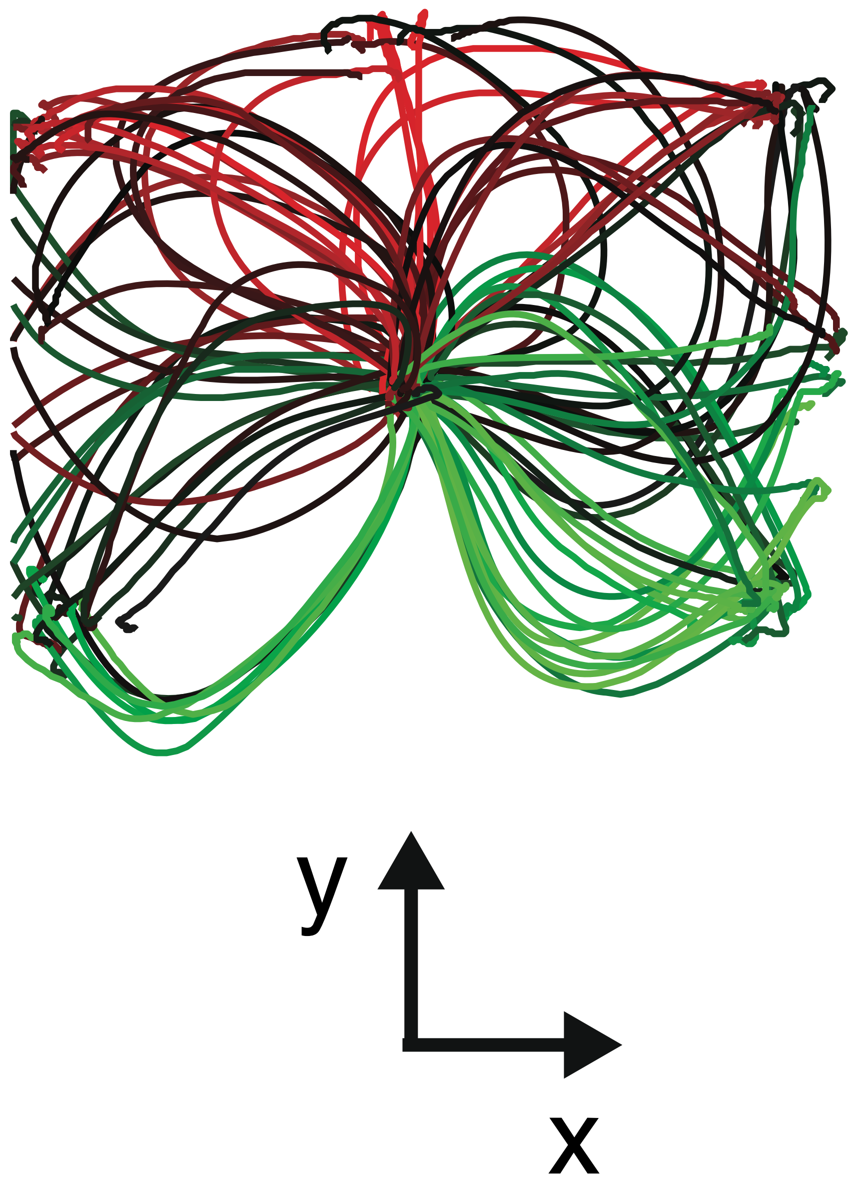




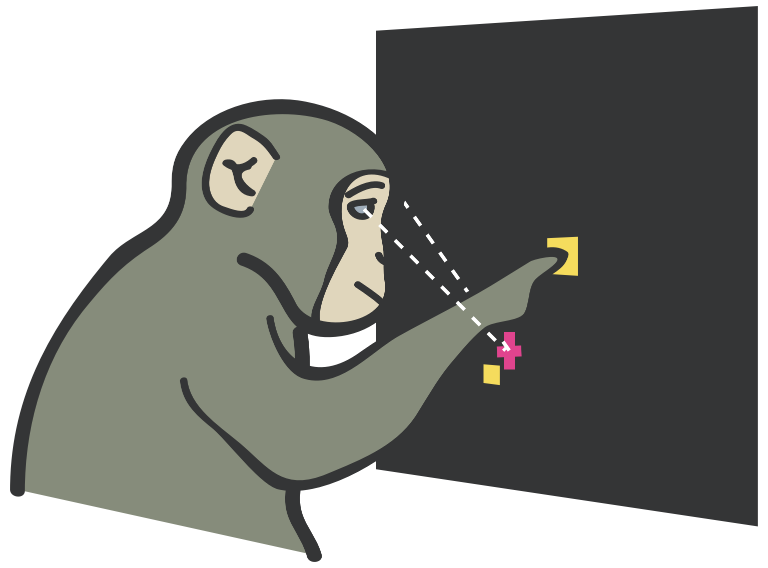
Churchland*, Cunningham* … Shenoy, *Nature* (2012)



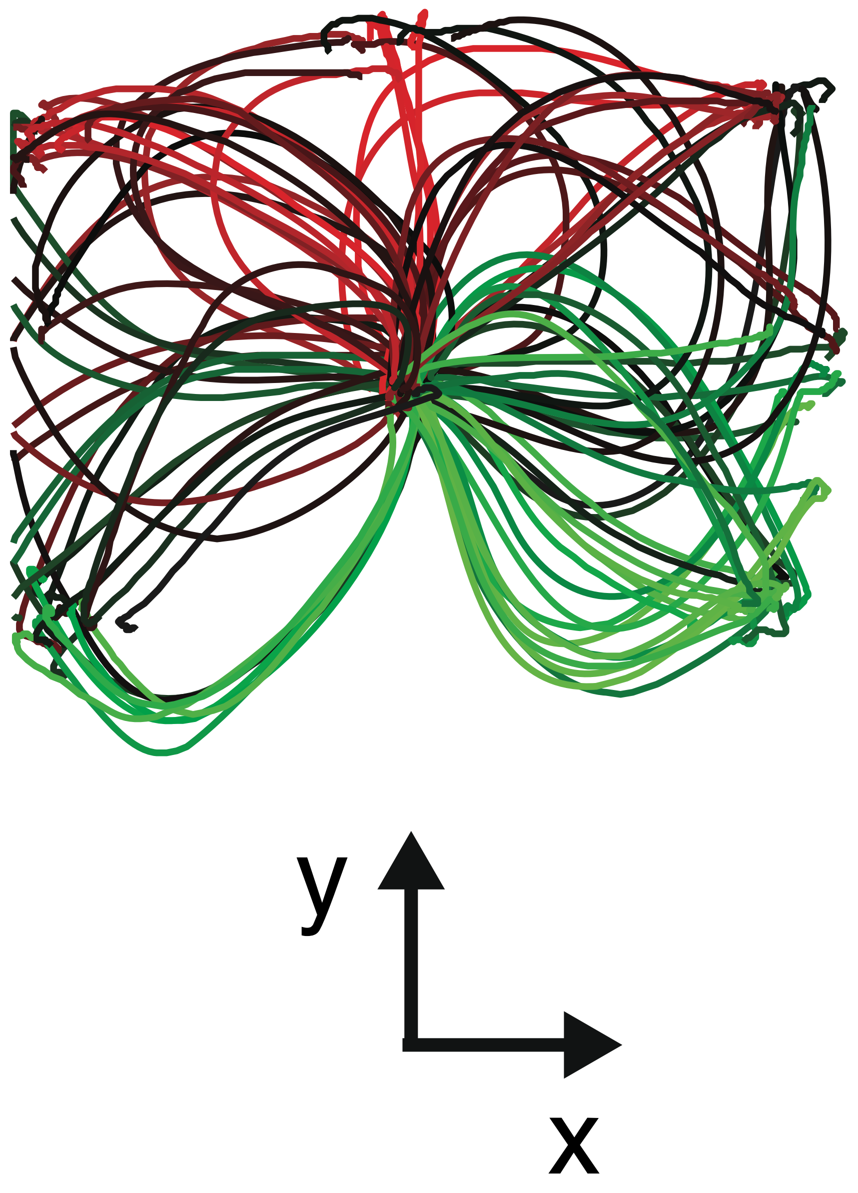
Reach trajectories



Churchland*, Cunningham* … Shenoy, *Nature* (2012)



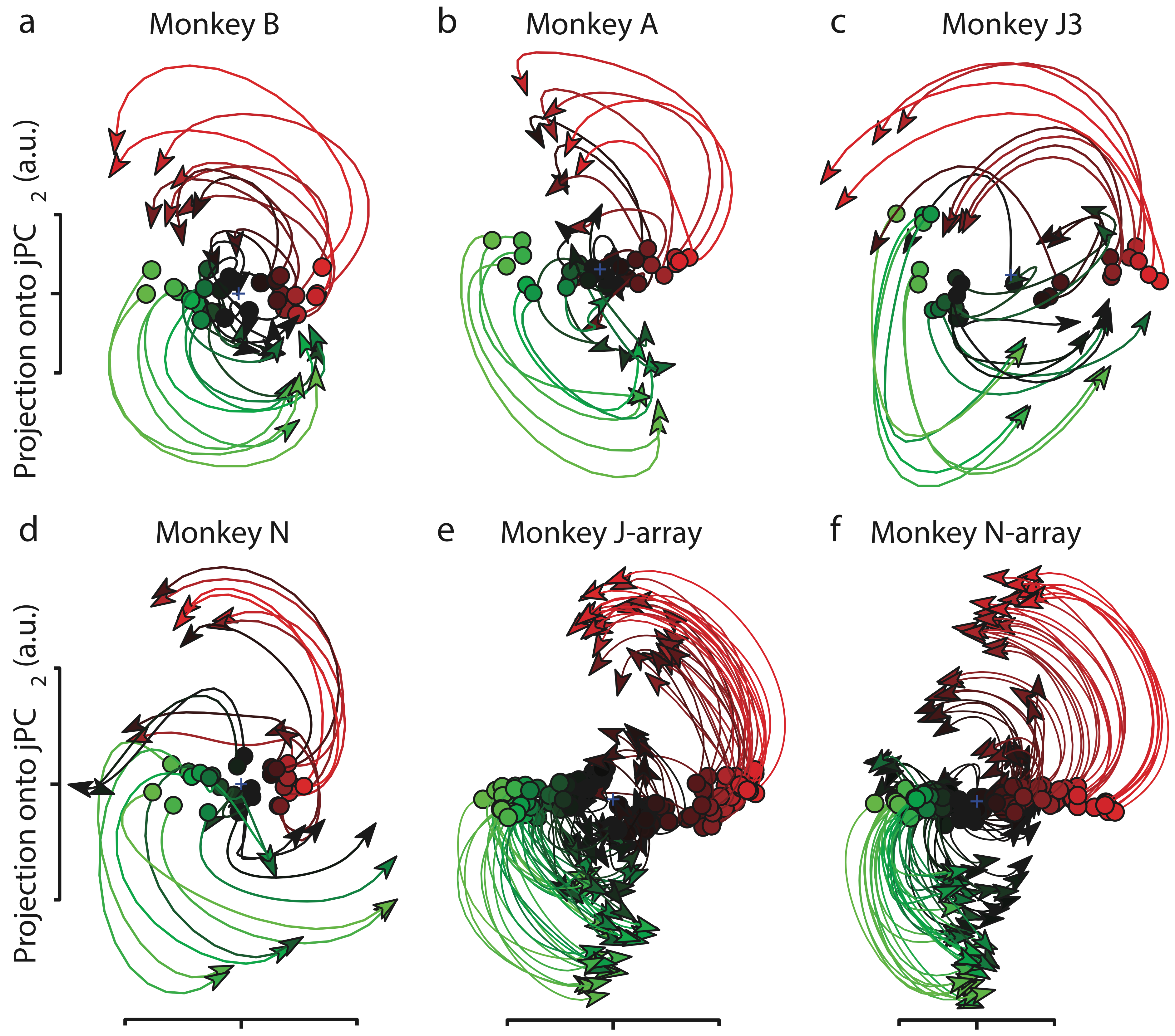
Reach
trajectories



Churchland*, Cunningham* … Shenoy, *Nature* (2012)

Rotational dynamics in monkeys & humans

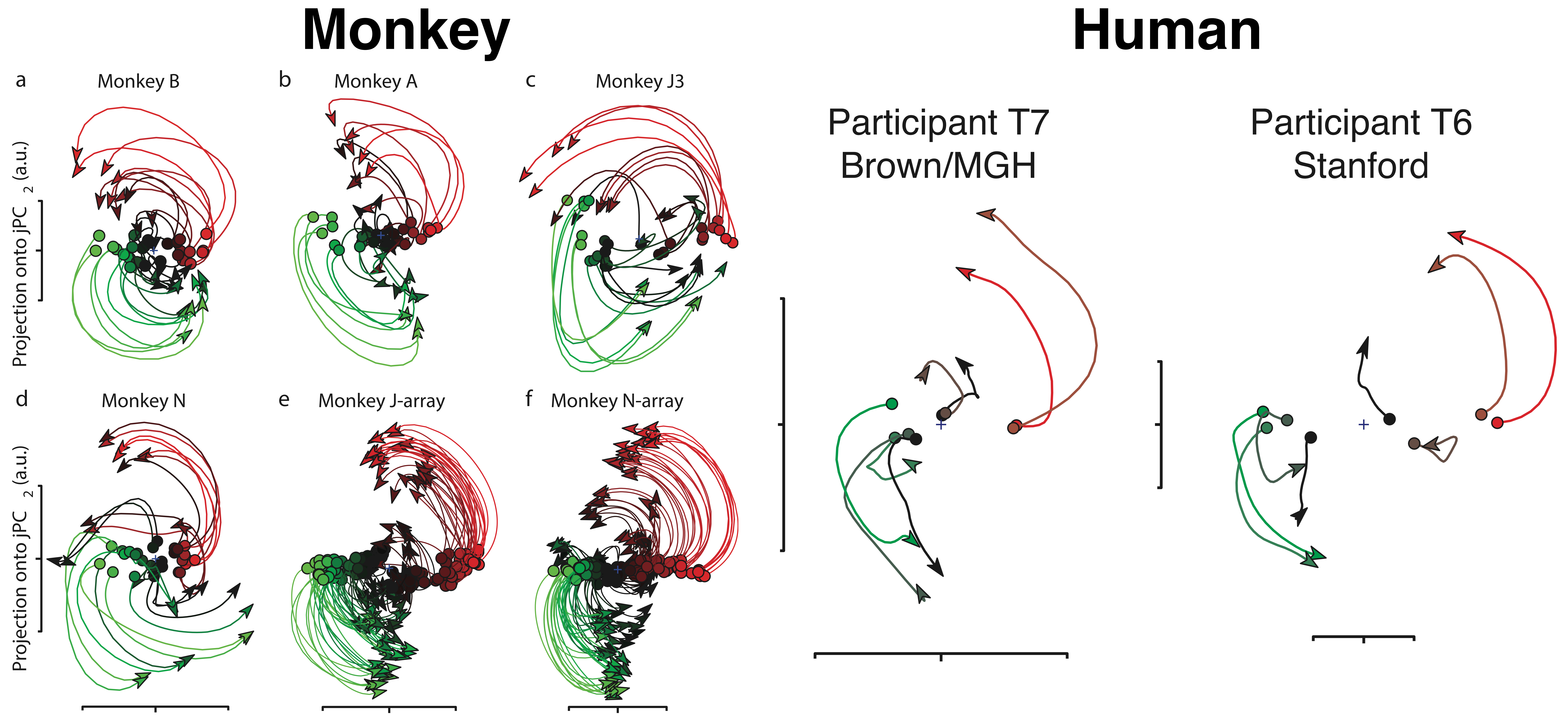
Monkey



Churchland*, Cunningham* … Shenoy, *Nature* (2012)

Pandarinath, … Hochberg, Henderson*, Shenoy*, *eLife* (2015)

Rotational dynamics in monkeys & humans



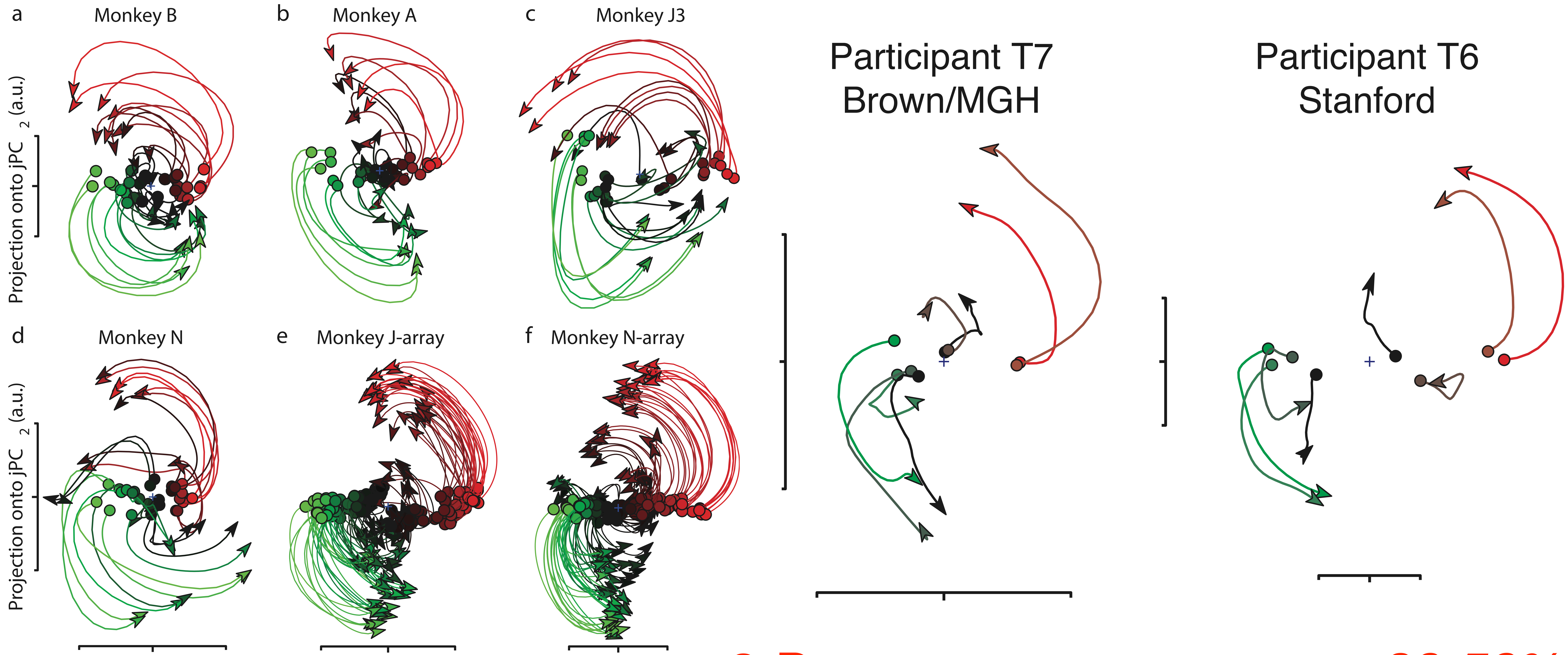
Churchland*, Cunningham* … Shenoy, *Nature* (2012)

Pandarinath, … Hochberg, Henderson*, Shenoy*, *eLife* (2015)

Rotational dynamics in monkeys & humans

Monkey

Human



2-D state-space represents ***30-50%*** of data variance

Churchland*, Cunningham* … Shenoy, *Nature* (2012)

Pandarinath, … Hochberg, Henderson*, Shenoy*, *eLife* (2015)

What you'll hear about in this lecture

Neural network basics

Deep autoencoders

Intro to neural population dynamics

Population analyses shed light on network-level computation

Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations

Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations

- *Population analyses* focus on internal processes, rather than representation of externally-measurable variables

Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations

- *Population analyses* focus on internal processes, rather than representation of externally-measurable variables

- Getting enough neurons used to require **highly-stereotyped behaviors**.

Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations

- *Population analyses* focus on internal processes, rather than representation of externally-measurable variables

- Getting enough neurons used to require **highly-stereotyped behaviors**.

Session 1:
Record neuron
set A

Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations

- *Population analyses* focus on internal processes, rather than representation of externally-measurable variables

- Getting enough neurons used to require **highly-stereotyped behaviors**.

Session 1: Record neuron set A

Session 2: Record neuron set B

Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations
- *Population analyses* focus on internal processes, rather than representation of externally-measurable variables
- Getting enough neurons used to require **highly-stereotyped behaviors**.

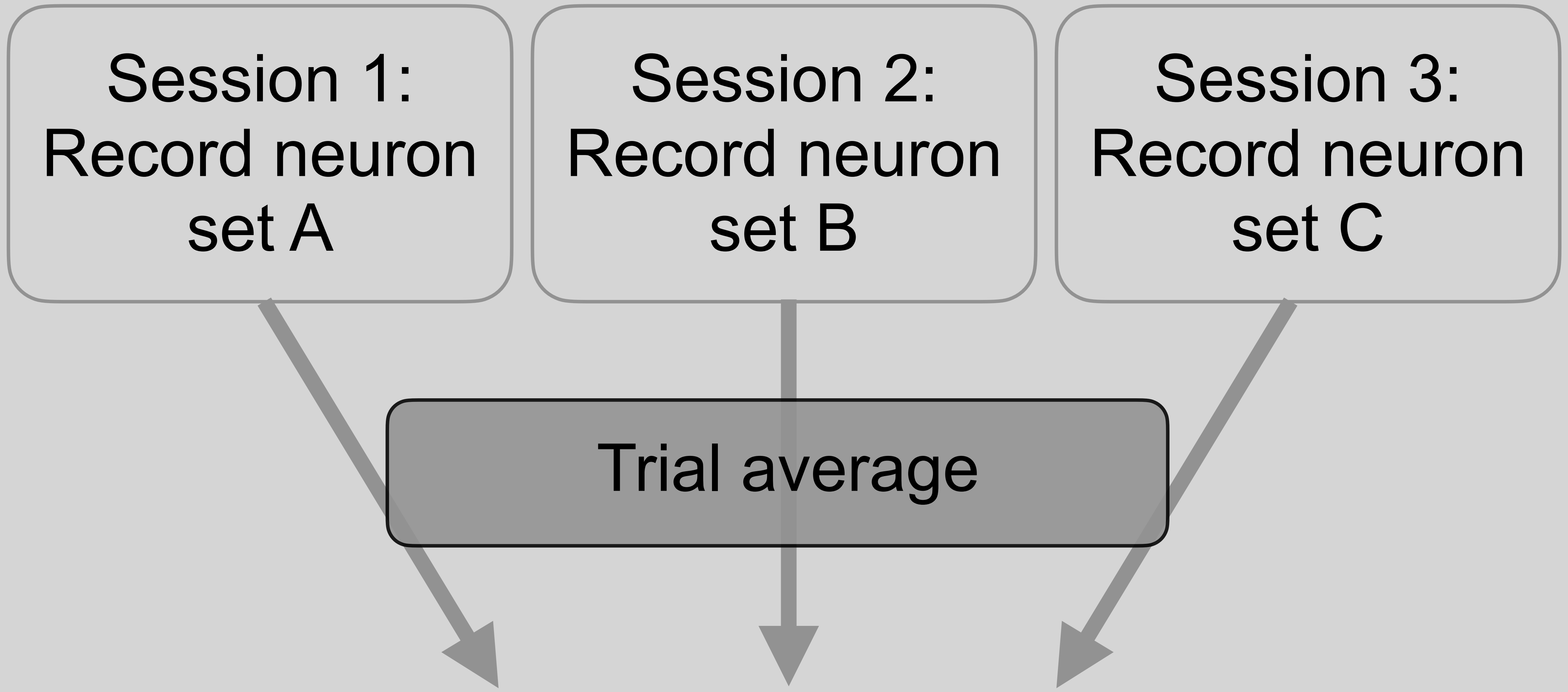
Session 1: Record neuron set A	Session 2: Record neuron set B	Session 3: Record neuron set C

Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations

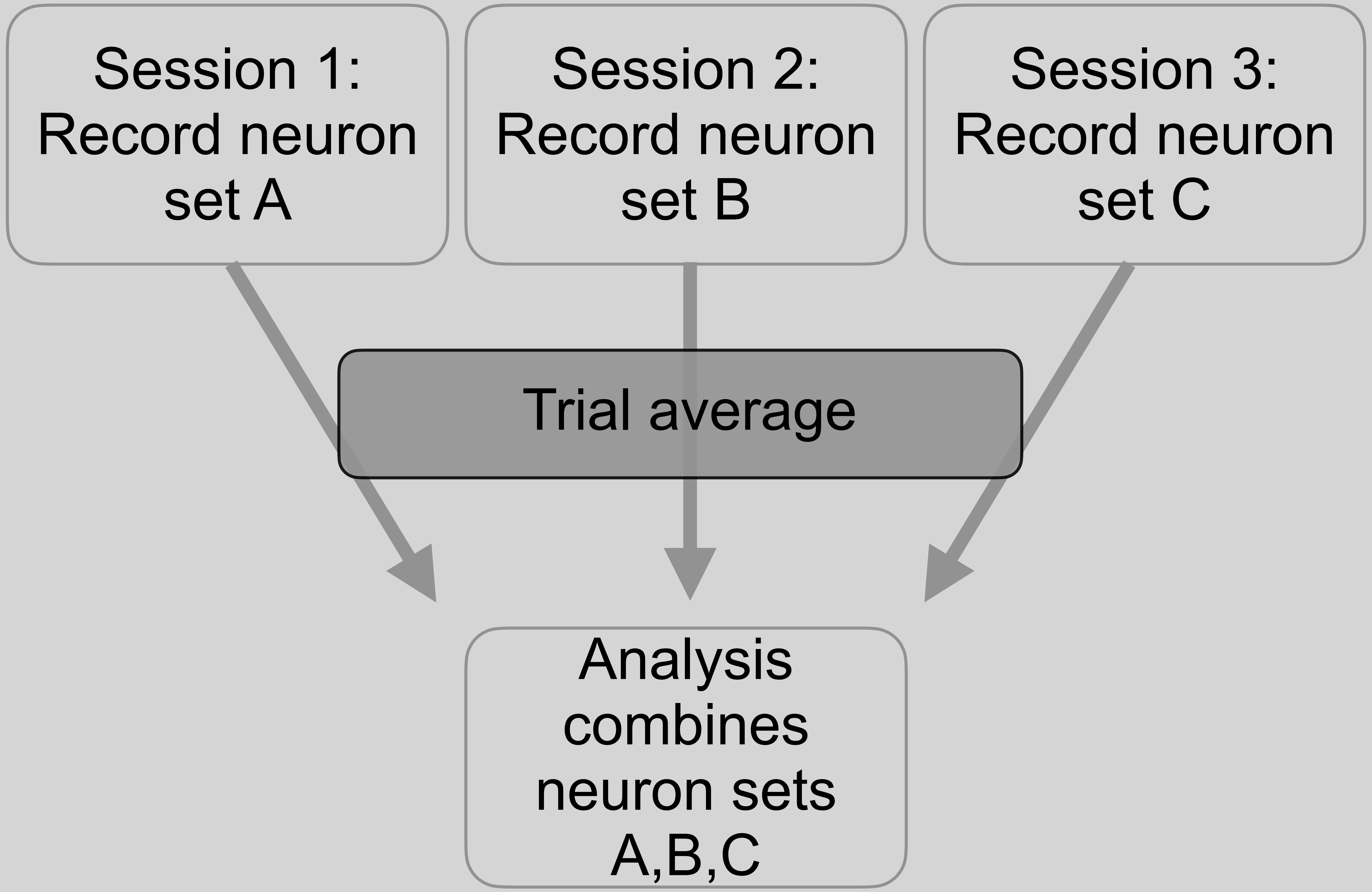
- *Population analyses* focus on internal processes, rather than representation of externally-measurable variables

- Getting enough neurons used to require **highly-stereotyped behaviors**.



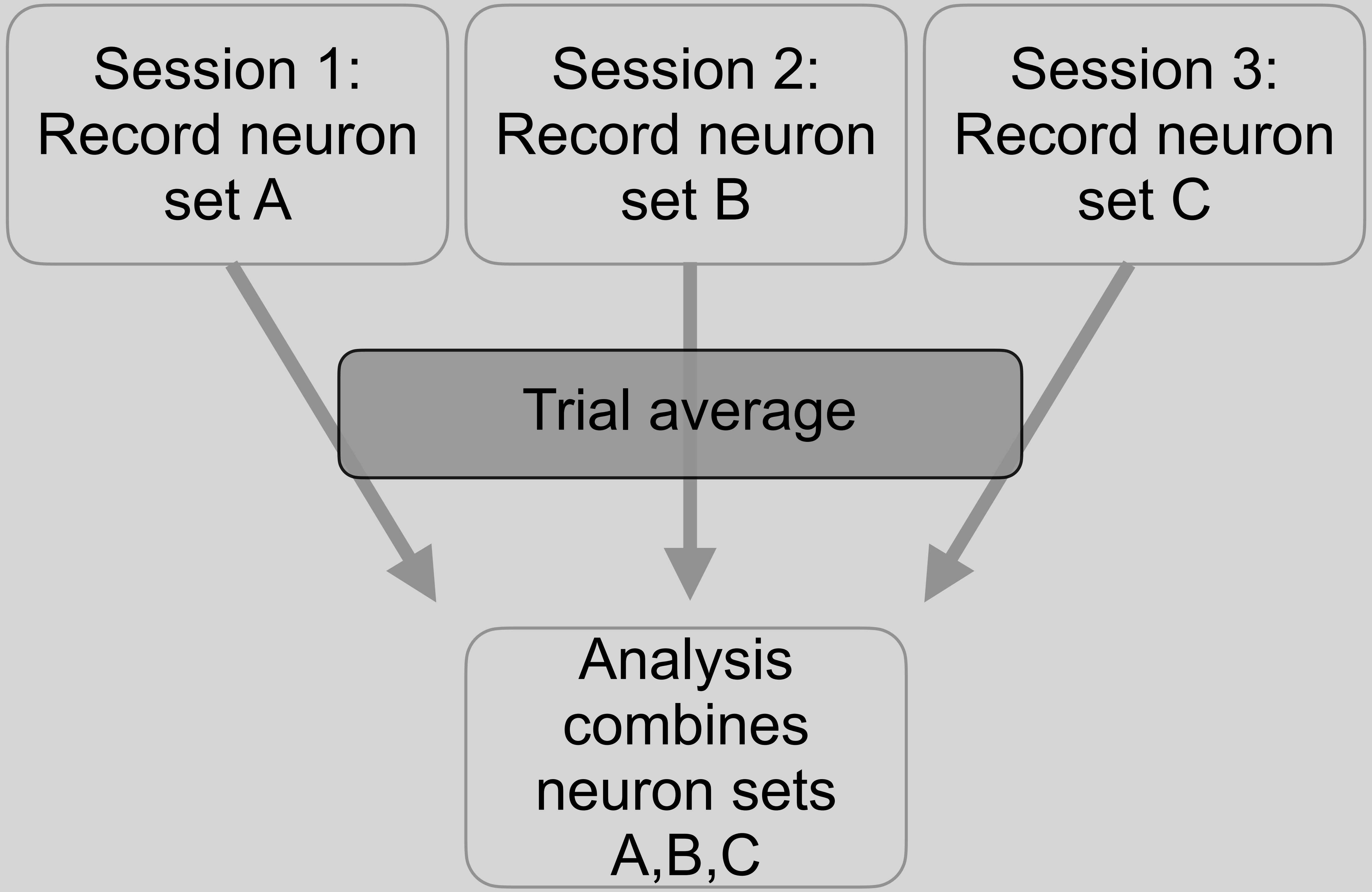
Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations
- *Population analyses* focus on internal processes, rather than representation of externally-measurable variables
- Getting enough neurons used to require **highly-stereotyped behaviors**.

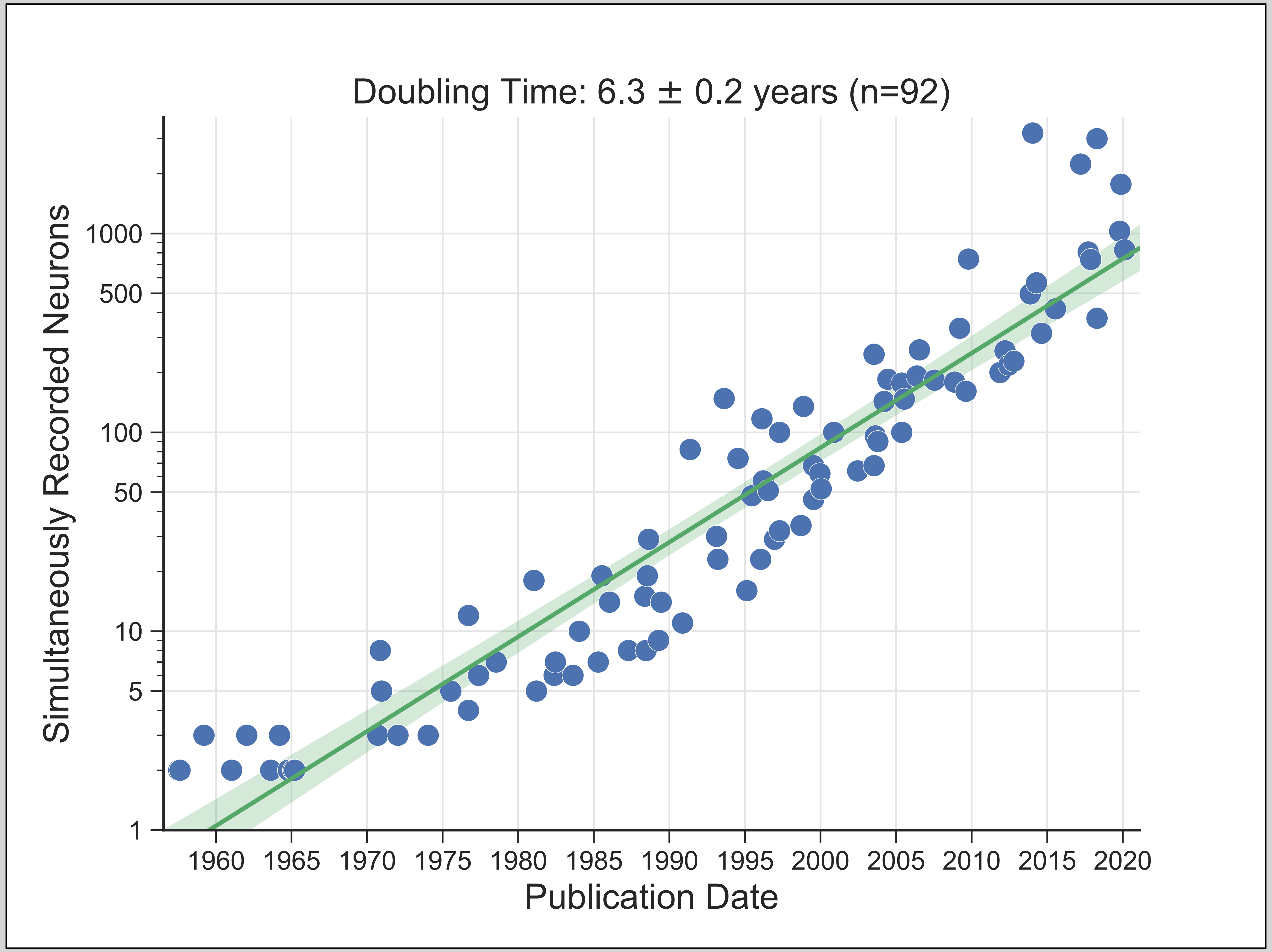


Population analyses shed light on network-level computation

- Function arises from the *collective activity* of neural populations
- *Population analyses* focus on internal processes, rather than representation of externally-measurable variables
- Getting enough neurons used to require **highly-stereotyped behaviors**.
- But this is changing…



Recording capacity is increasing dramatically

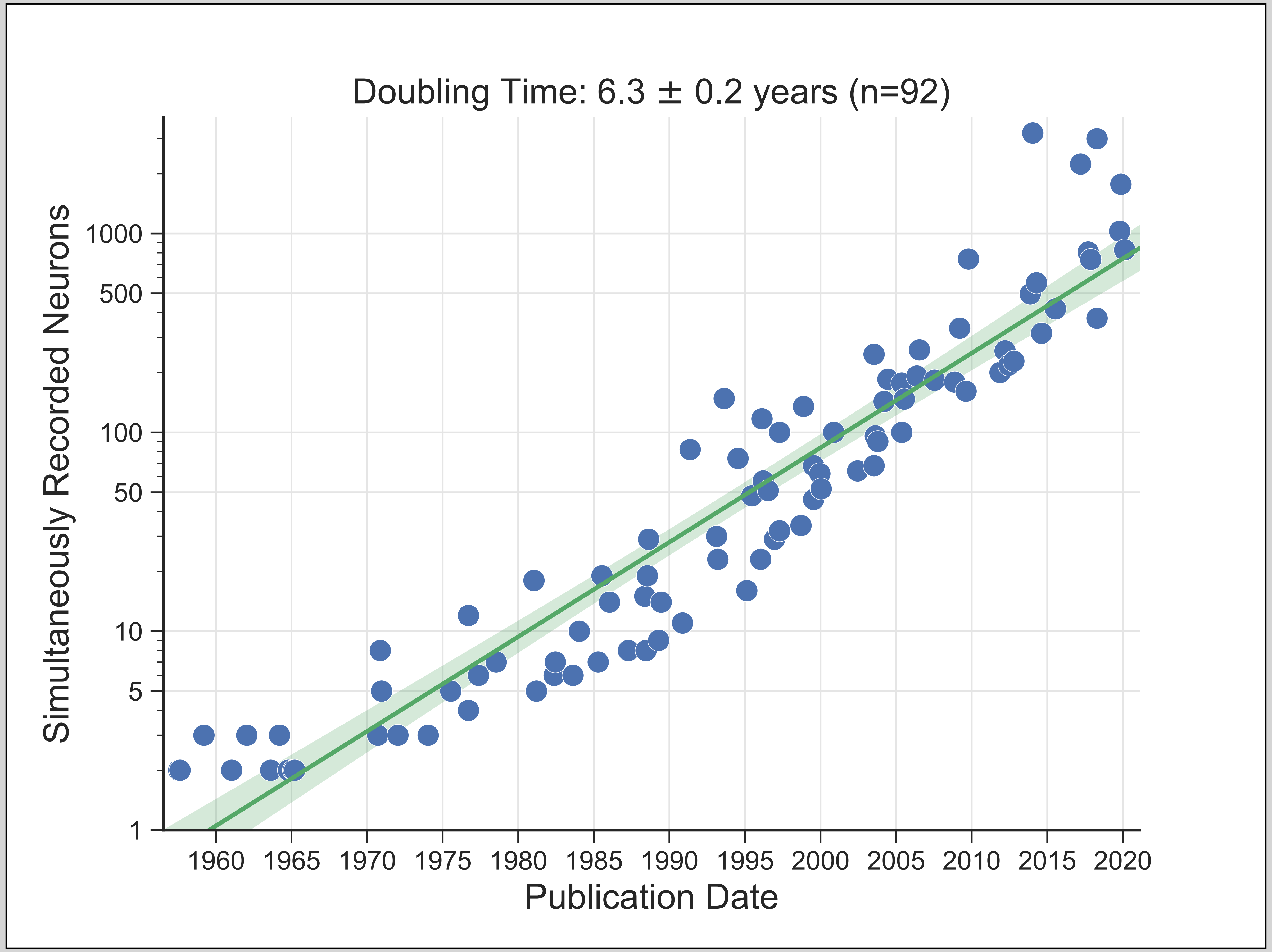


Stevenson & Kording, *Nat Neuro* 2011

updates: stevenson.lab.uconn.edu

Recording capacity is increasing dramatically

- Capacity doubling every ~6 years



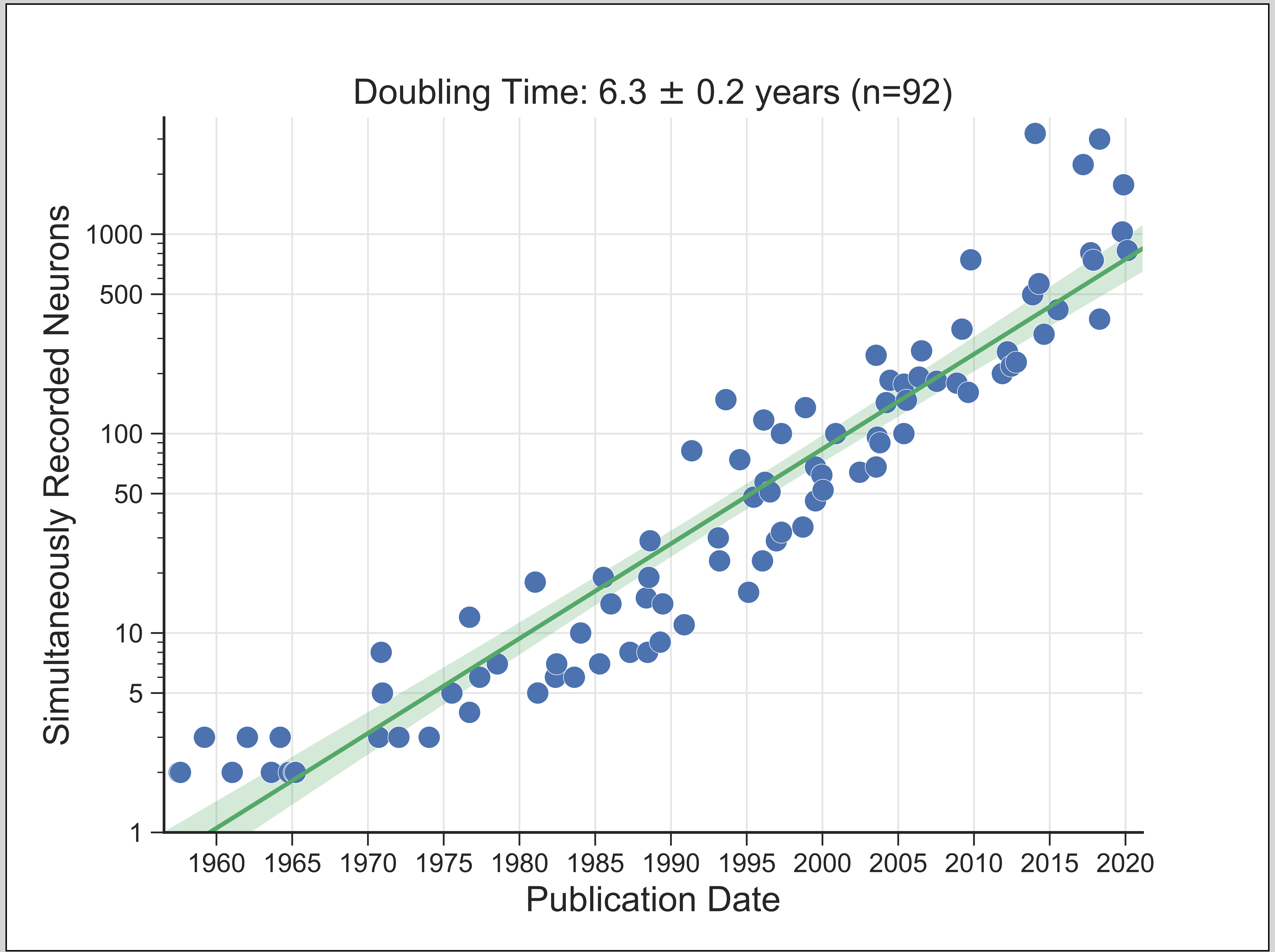
Stevenson & Kording, *Nat Neuro* 2011

updates: stevenson.lab.uconn.edu

Recording capacity is increasing dramatically

- Capacity doubling every ~6 years

- Key opportunity: large-scale recording + new ML ➔ high-resolution *single-trial* analyses



Stevenson & Kording, *Nat Neuro* 2011

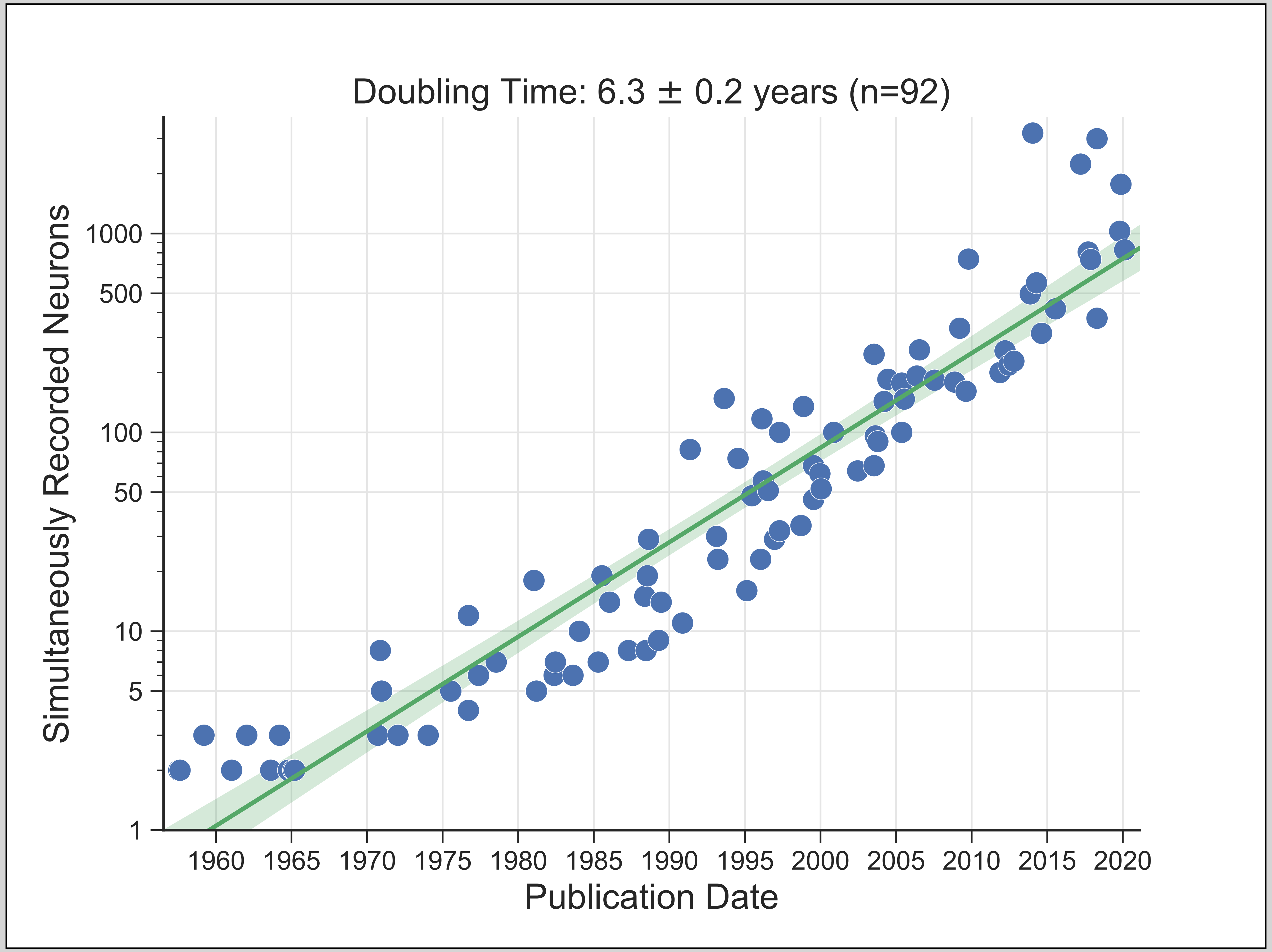
updates: stevenson.lab.uconn.edu

Recording capacity is increasing dramatically

- Capacity doubling every ~6 years

- <u>Key opportunity</u>: large-scale recording + new ML ➔ high-resolution *single-trial* analyses

- Can study neural processes that were previously challenging



Stevenson & Kording, *Nat Neuro* 2011

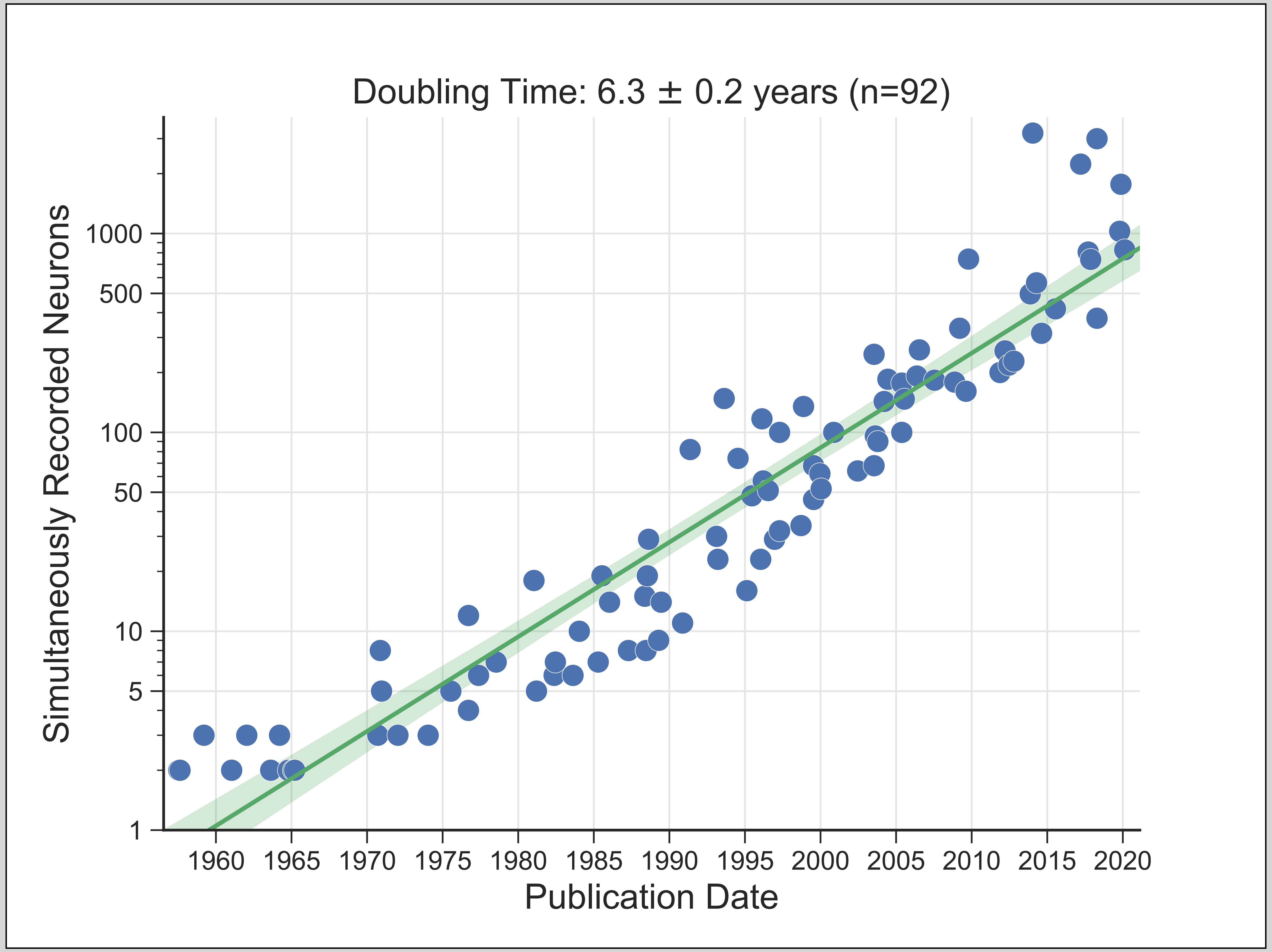
updates: stevenson.lab.uconn.edu

Recording capacity is increasing dramatically

- Capacity doubling every ~6 years

- Key opportunity: large-scale recording + new ML ➔ high-resolution *single-trial* analyses

- Can study neural processes that were previously challenging
  - Non-stereotyped - complex behaviors, different model systems



Stevenson & Kording, *Nat Neuro* 2011

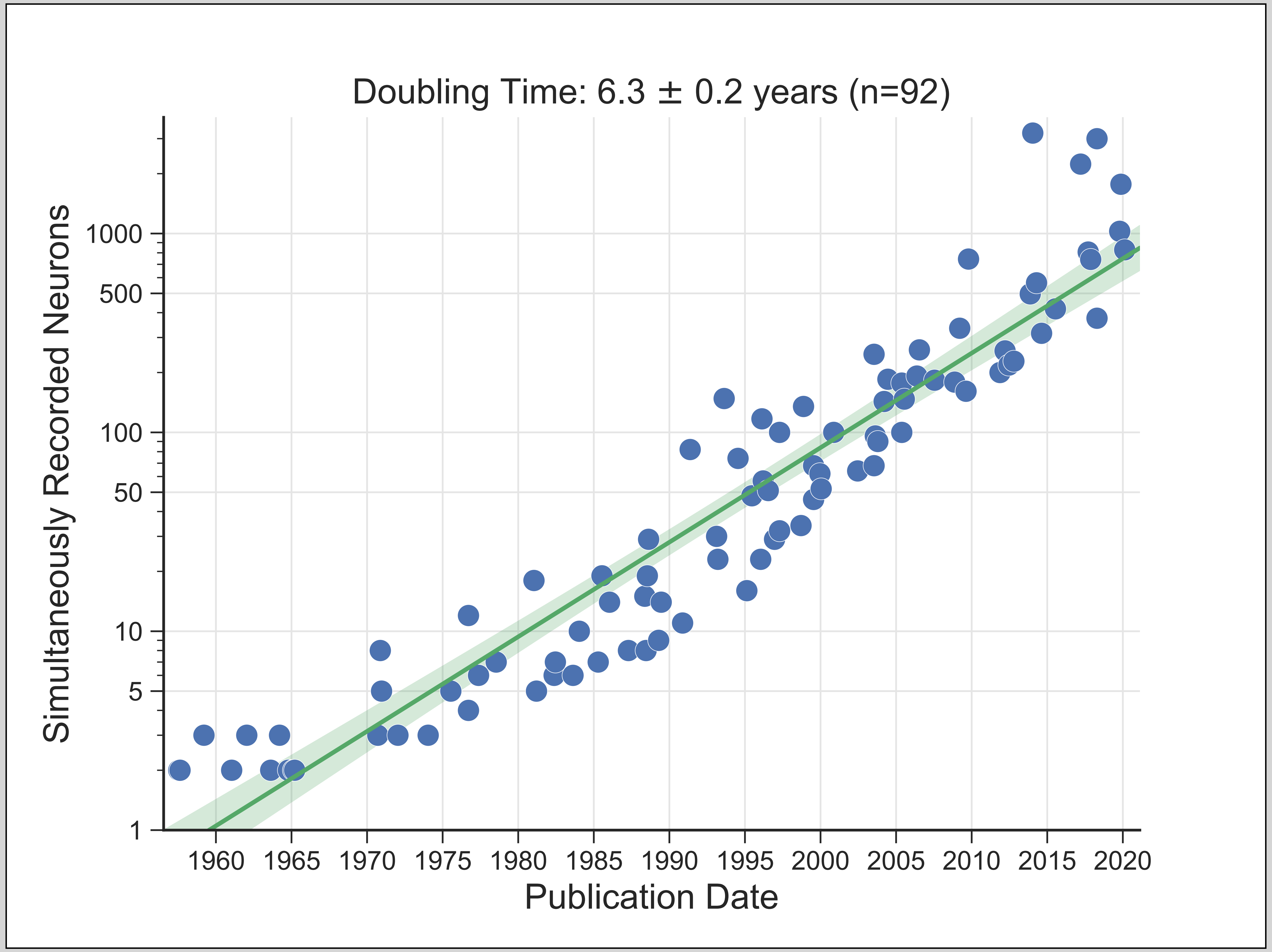
updates: stevenson.lab.uconn.edu

Recording capacity is increasing dramatically

- Capacity doubling every ~6 years

- Key opportunity: large-scale recording + new ML ➔ high-resolution *single-trial* analyses

- Can study neural processes that were previously challenging
  - Non-stereotyped - complex behaviors, different model systems
  - Cases where single-trial variability is important, e.g., cognitive processes



Stevenson & Kording, *Nat Neuro* 2011

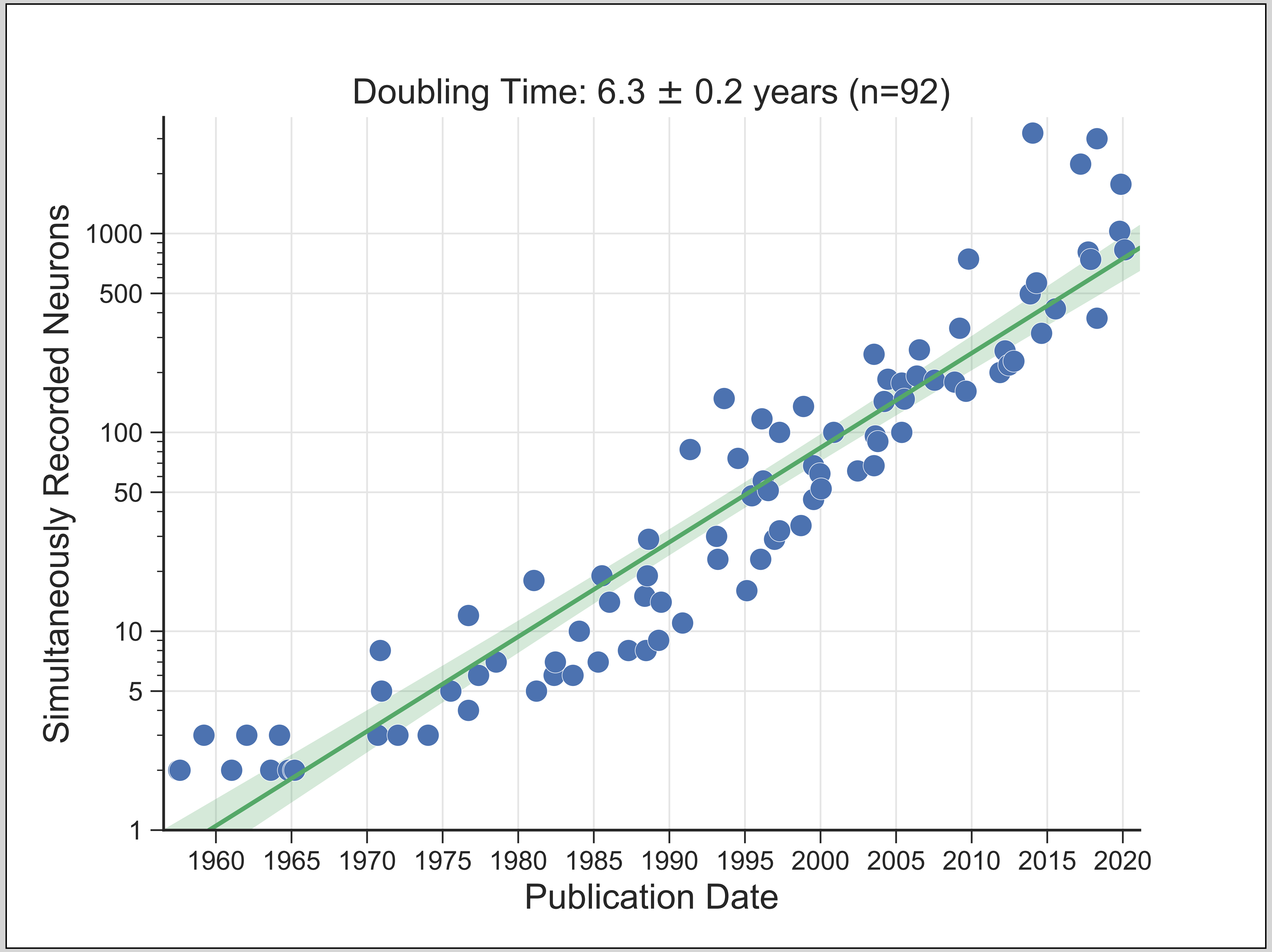
updates: stevenson.lab.uconn.edu

Recording capacity is increasing dramatically

- Capacity doubling every ~6 years

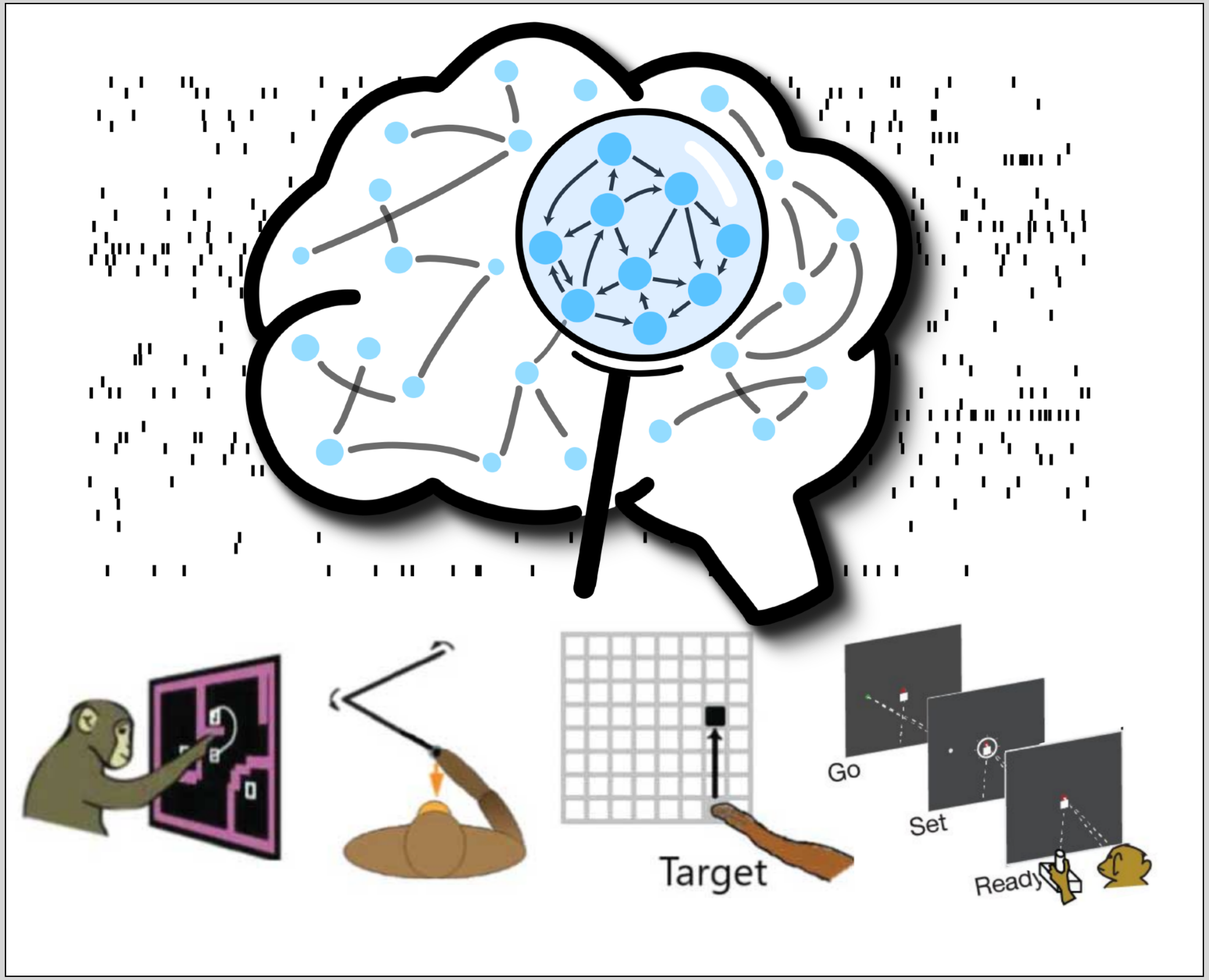
- Key opportunity: large-scale recording + new ML ➔ high-resolution *single-trial* analyses

- Can study neural processes that were previously challenging
  - Non-stereotyped - complex behaviors, different model systems
  - Cases where single-trial variability is important, e.g., cognitive processes
  - Brain-machine interfaces



Stevenson & Kording, *Nat Neuro* 2011

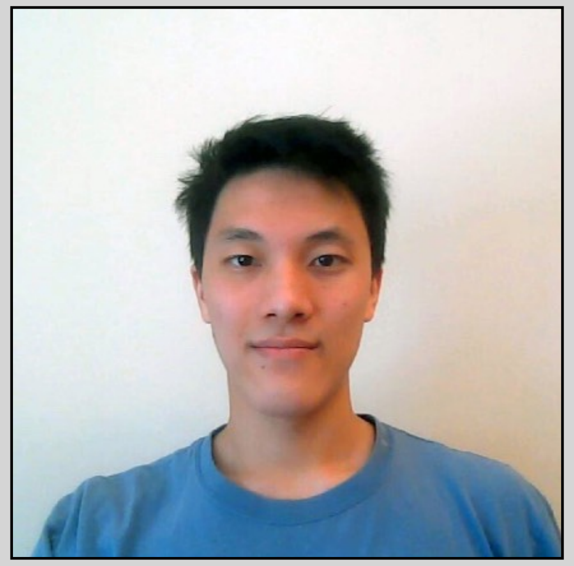
updates: stevenson.lab.uconn.edu



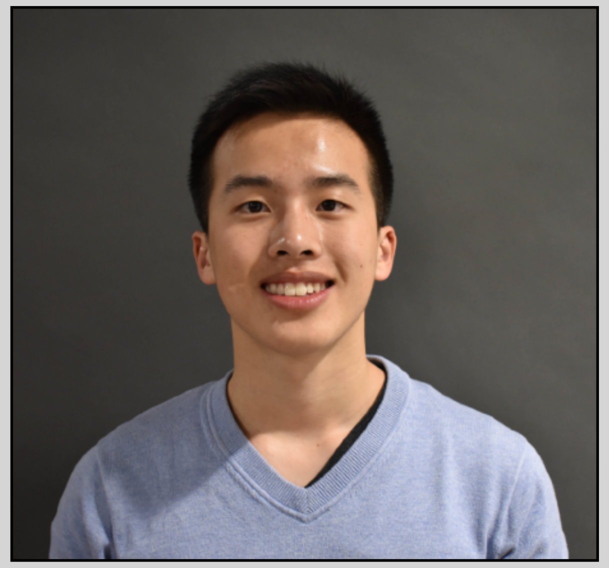
Neural Latents Benchmark ’21

A benchmark for latent variable models of neural data

Pe*, Ye*…Pandarinath, *NeurIPS 2021*
http://neurallatents.github.io



Felix Pe



Joel Ye

Pain points around model development

Pain points around model development

Users

Pain points around model development

Users

- Unclear which approaches perform well

Pain points around model development

Users

- Unclear which approaches perform well
- Unsure whether an approach is relevant to their data

Pain points around model development

Users

- Unclear which approaches perform well
- Unsure whether an approach is relevant to their data
- Must adapt to different APIs

Pain points around model development

Users

- Unclear which approaches perform well
- Unsure whether an approach is relevant to their data
- Must adapt to different APIs

Developers

Pain points around model development

Users

- Unclear which approaches perform well
- Unsure whether an approach is relevant to their data
- Must adapt to different APIs

Developers

- Determine relevant datasets and evaluation criteria *de novo*

Pain points around model development

Users

- Unclear which approaches perform well
- Unsure whether an approach is relevant to their data
- Must adapt to different APIs

Developers

- Determine relevant datasets and evaluation criteria *de novo*
- Responsible for determining best points of comparison

Pain points around model development

Users

- Unclear which approaches perform well
- Unsure whether an approach is relevant to their data
- Must adapt to different APIs

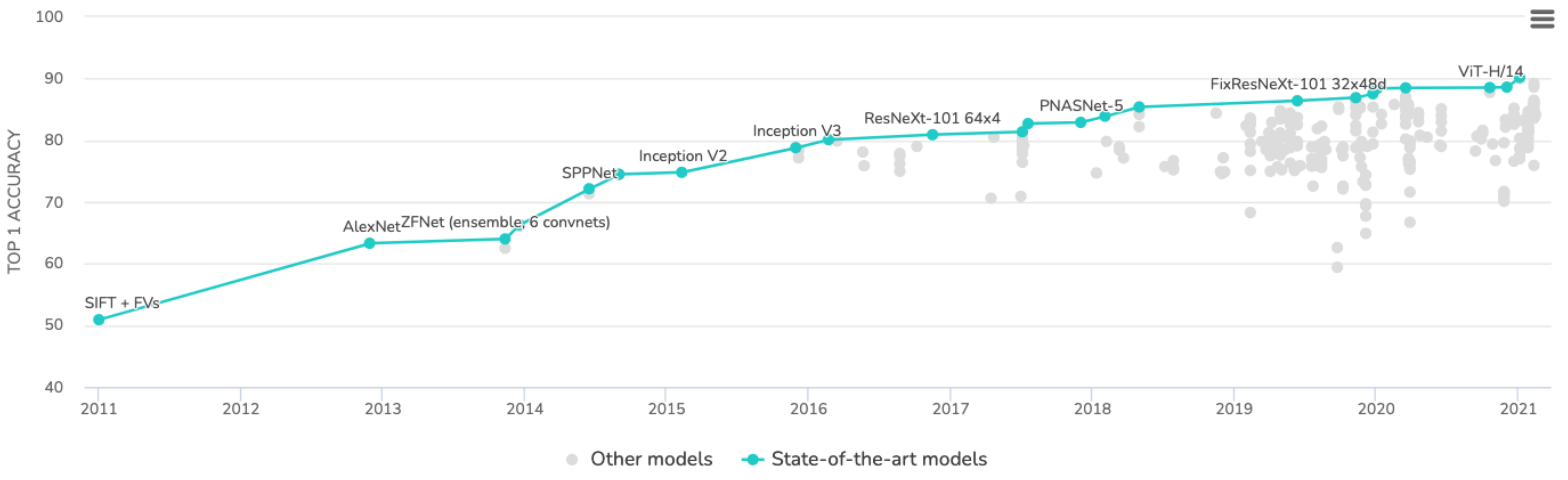
Developers

- Determine relevant datasets and evaluation criteria *de novo*
- Responsible for determining best points of comparison
- Forced to get other peoples' models to work!

Benchmarks facilitate progress

Image Classification on ImageNet

Leaderboard | Dataset



View: Top 1 Accuracy | All models | Edit

RANK	MODEL	TOP 1 ACCURACY	TOP 5 ACCURACY	NUMBER OF PARAMS	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	**Meta Pseudo Labels** (EfficientNet-L2)	90.2%	98.8%	480M	✓	Meta Pseudo Labels			2021
2	**Meta Pseudo Labels** (EfficientNet-B6-Wide)	90%	98.7%	390M	✓	Meta Pseudo Labels			2021

- Hosted model evaluation on private, held-out data

EvalAI

- Hosted model evaluation on private, held-out data

- Example notebooks

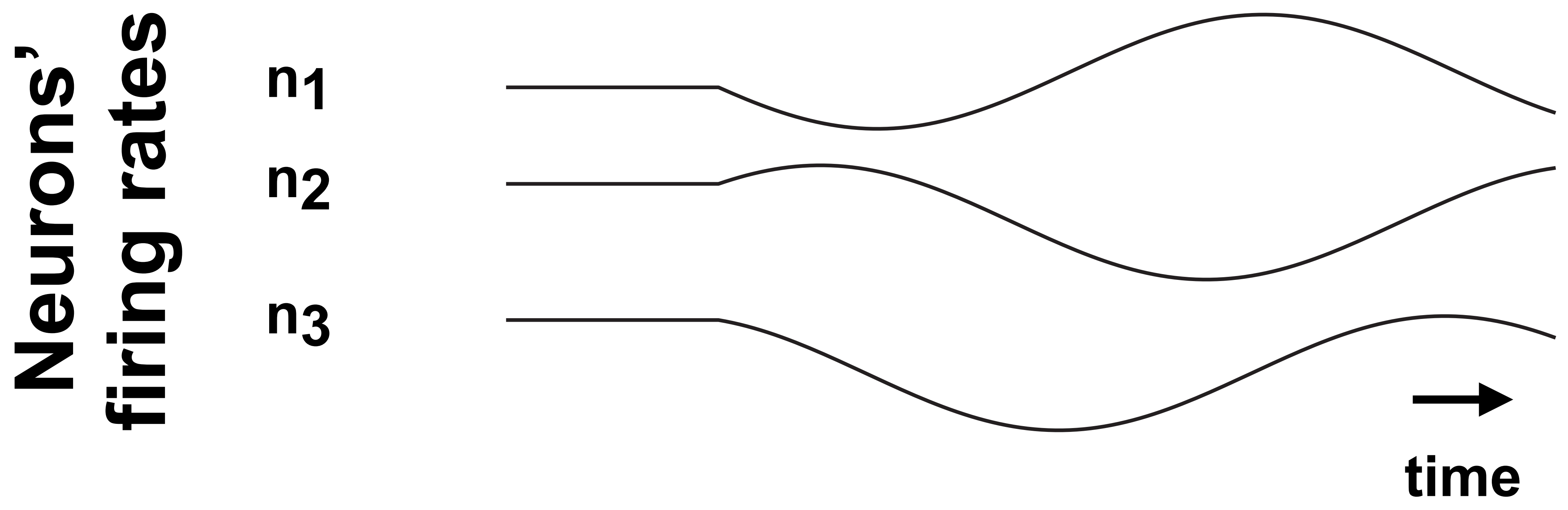
- Hosted model evaluation on private, held-out data
- Example notebooks
- Standardized dataset formatting / APIs

- Hosted model evaluation on private, held-out data
- Example notebooks
- Standardized dataset formatting / APIs
- Competition prizes!!

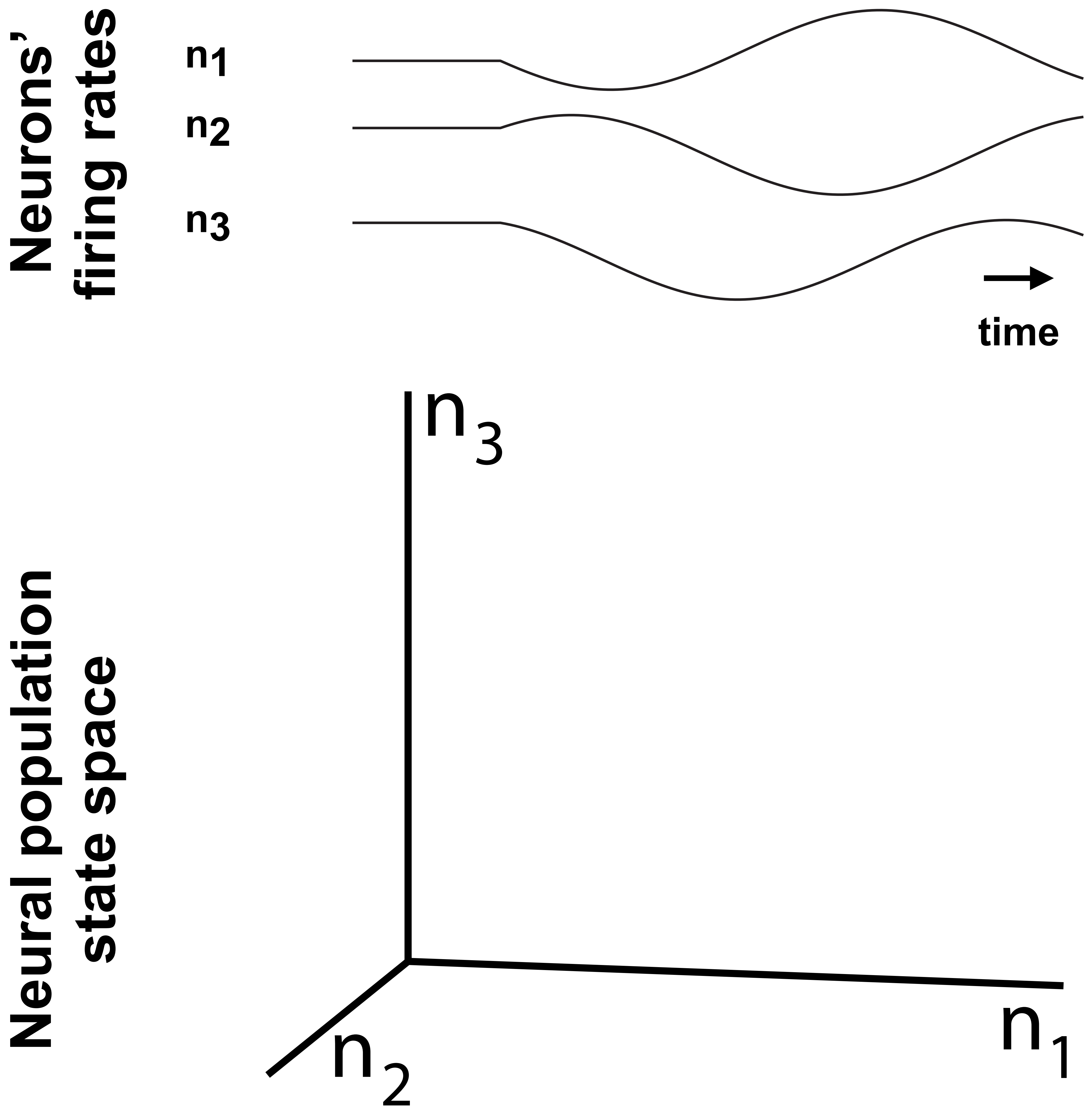

MC_Maze
MC_RTT
Area2_Bump
DMFC_RSG
Neuron count
Firing rate (spk/s)
Target onset
Go cue
Target acquisition
Bump
Ready
Set
Go
137 Channels
98 Channels
49 Channels
45 Channels
100 ms

Changes in neurons’ firing rates are coordinated

Changes in neurons' firing rates are coordinated

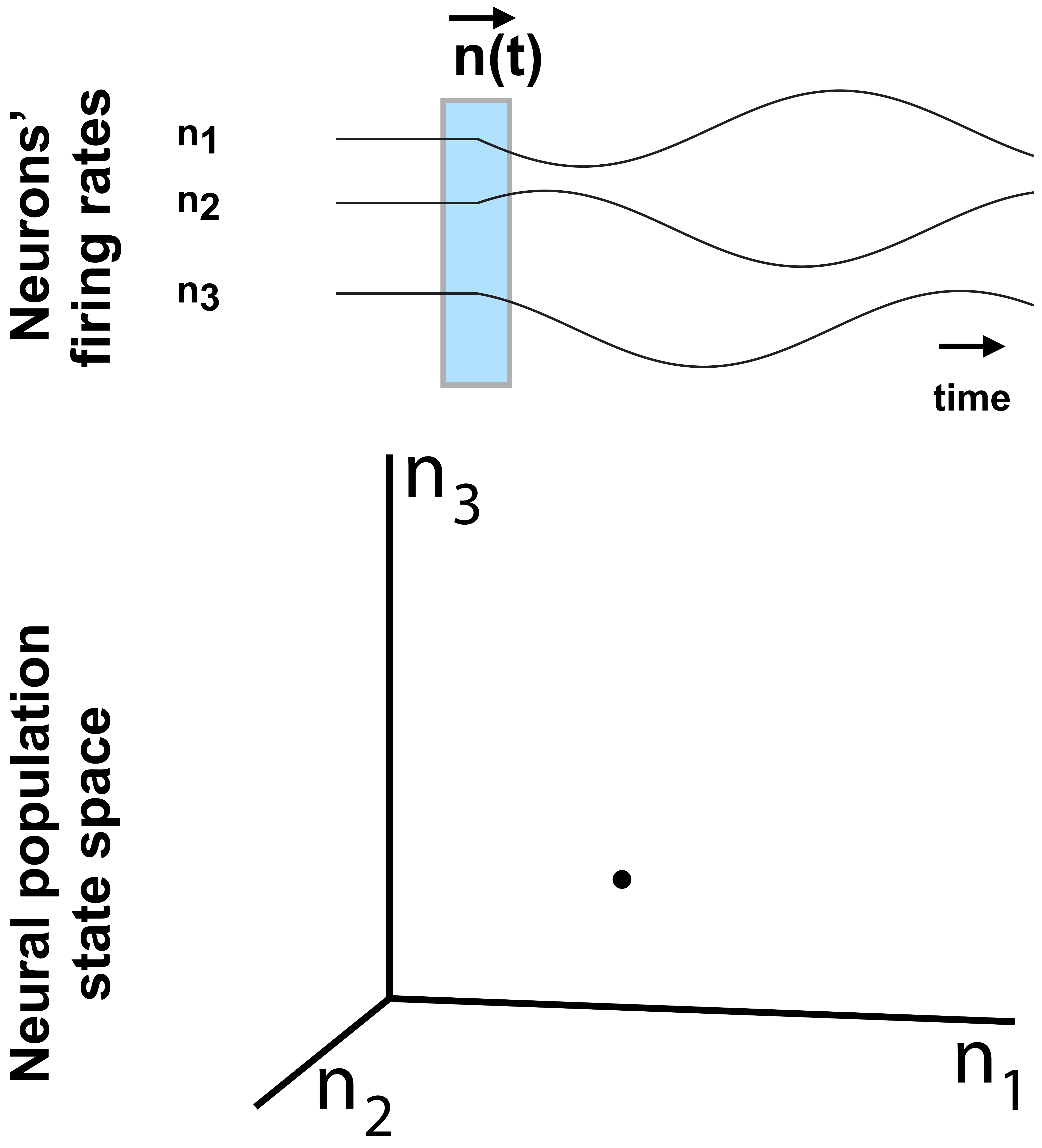


Changes in neurons' firing rates are coordinated



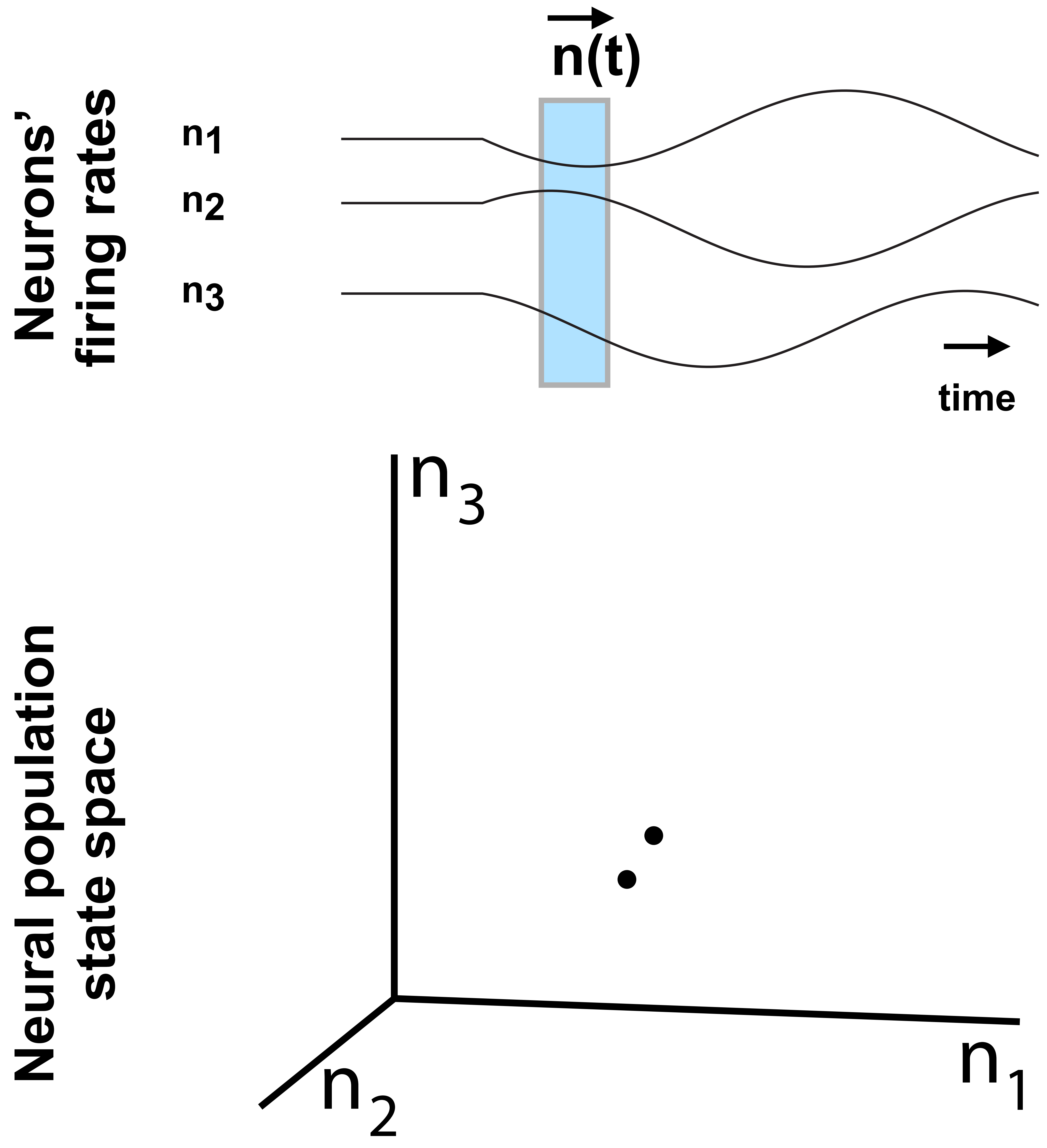
Adapted from Cunningham & Yu, *Nat Neuro* 2014

Changes in neurons' firing rates are coordinated



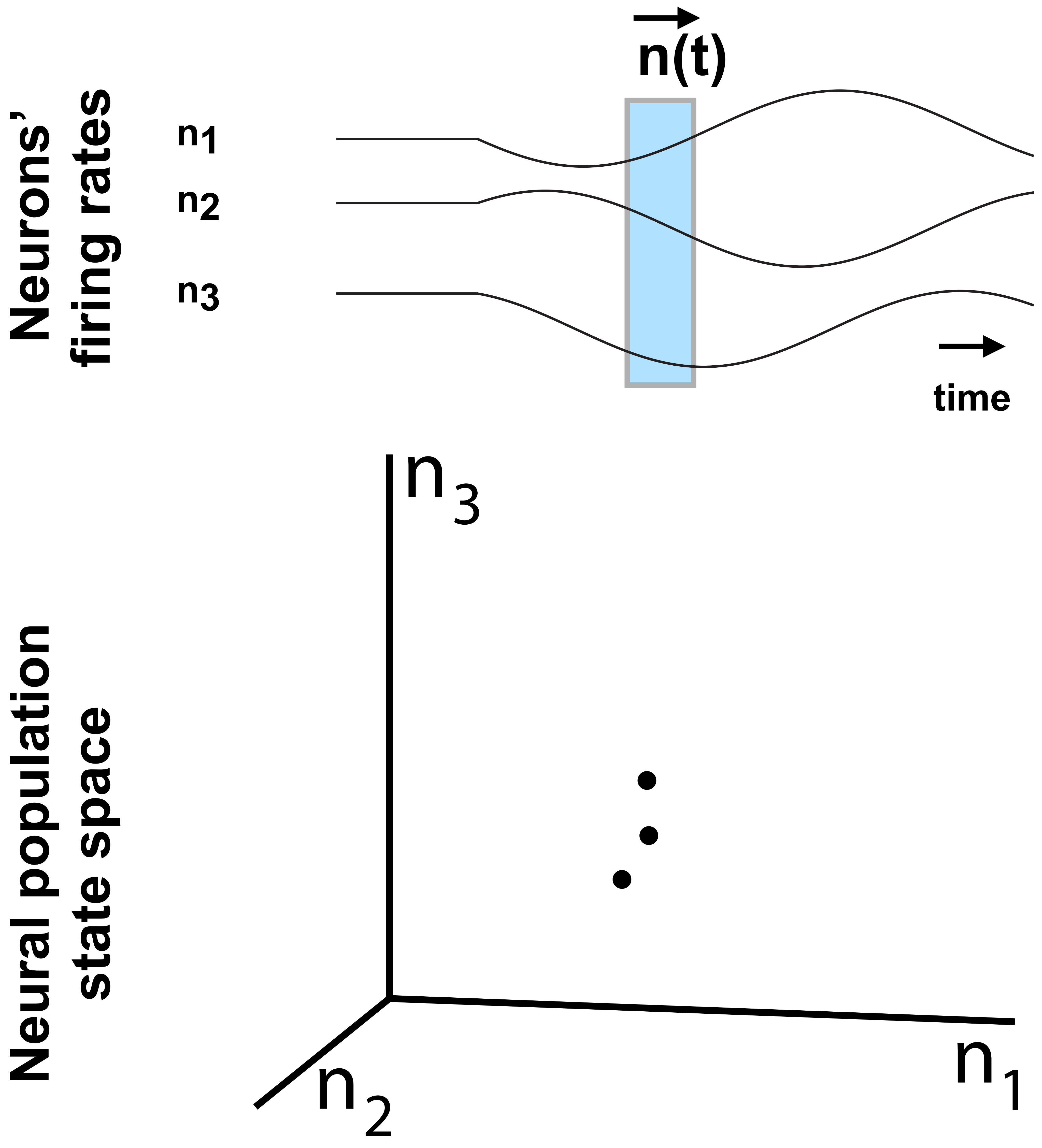
Adapted from Cunningham & Yu, *Nat Neuro* 2014

Changes in neurons’ firing rates are coordinated



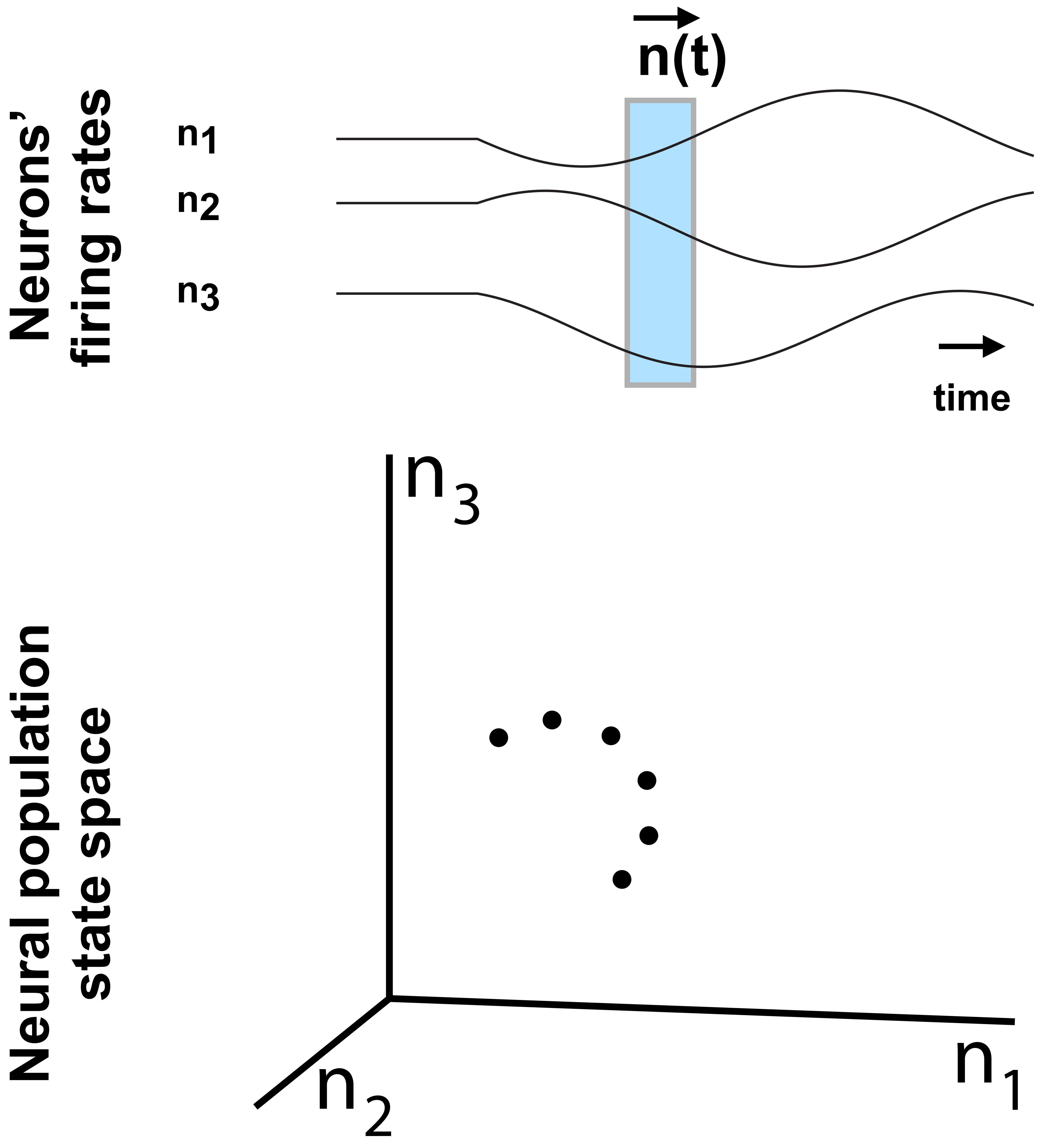
Adapted from Cunningham & Yu, *Nat Neuro* 2014

Changes in neurons’ firing rates are coordinated



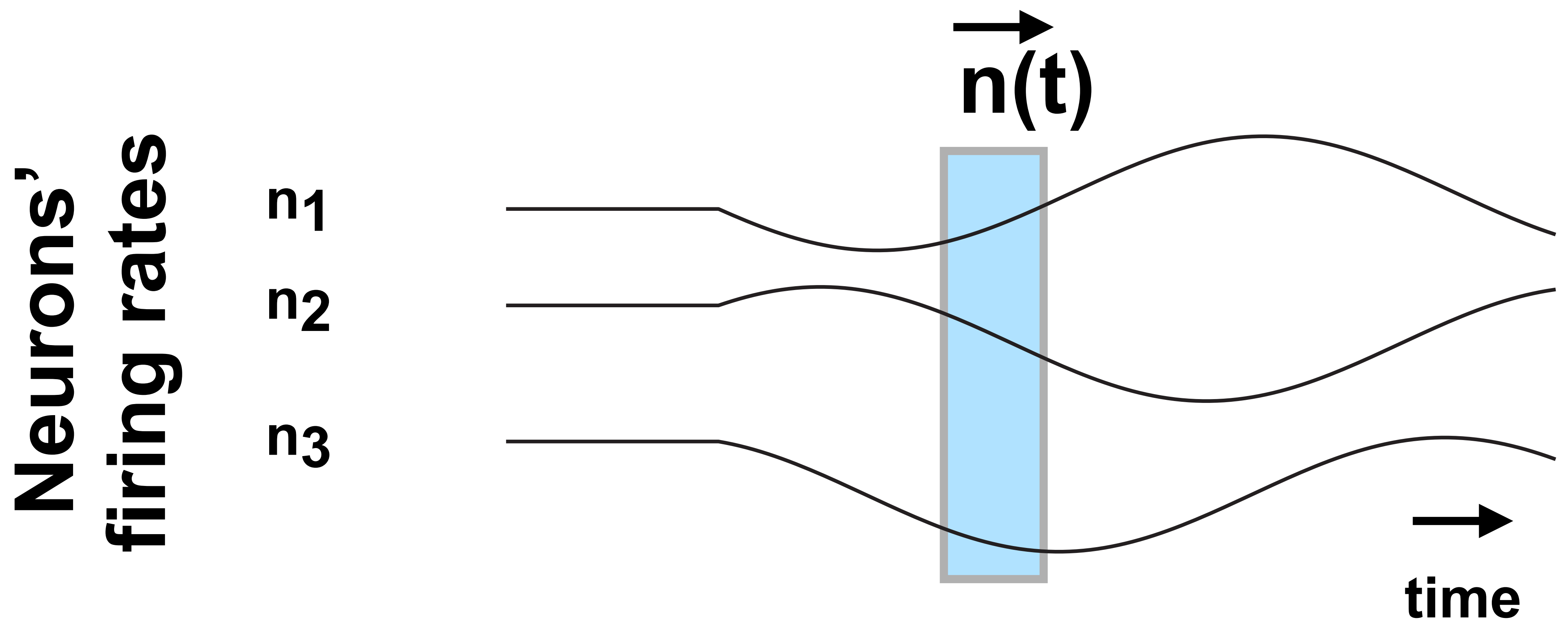
Adapted from Cunningham & Yu, *Nat Neuro* 2014

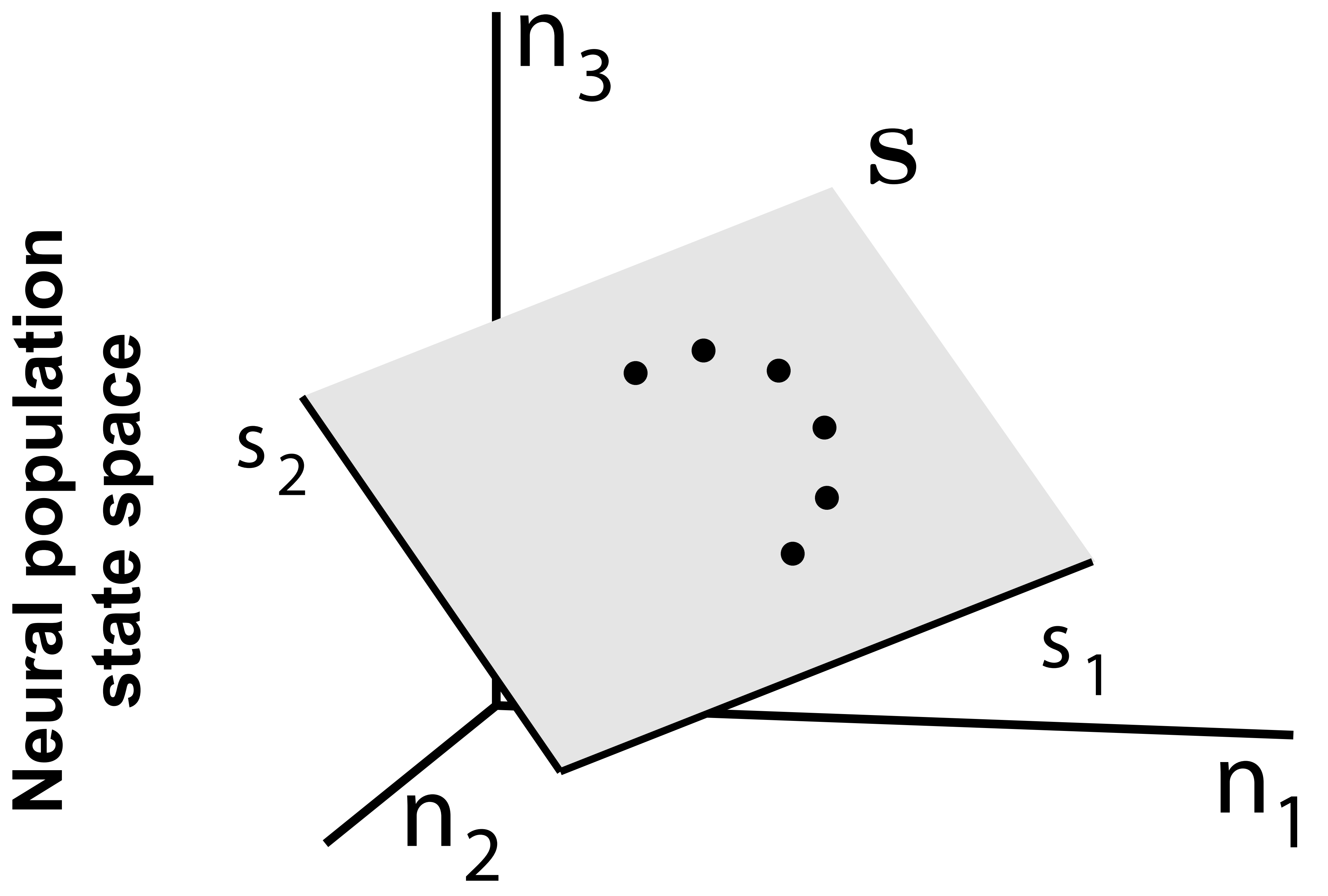
Changes in neurons' firing rates are coordinated



Adapted from Cunningham & Yu, *Nat Neuro* 2014

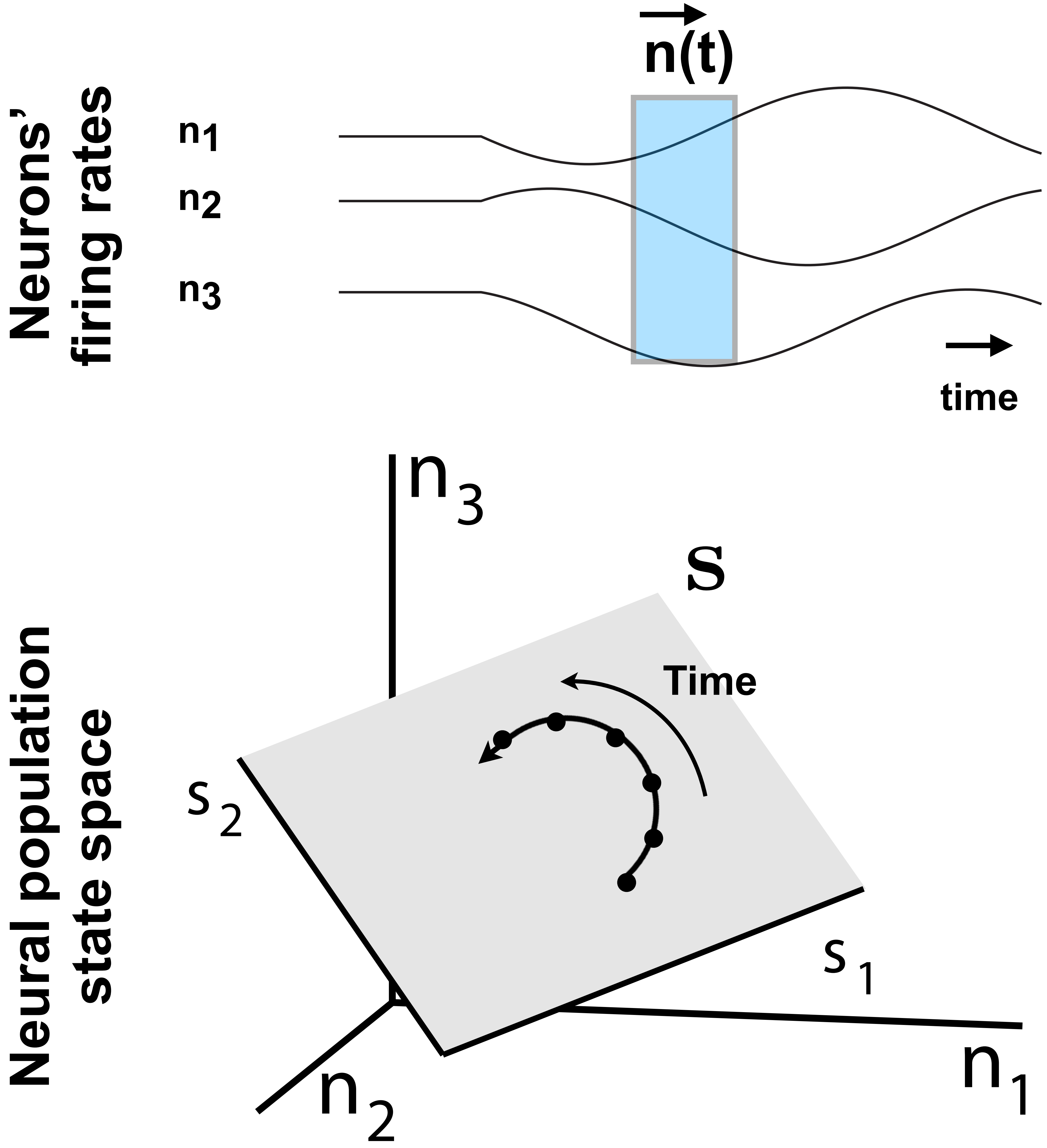
Changes in neurons' firing rates are coordinated



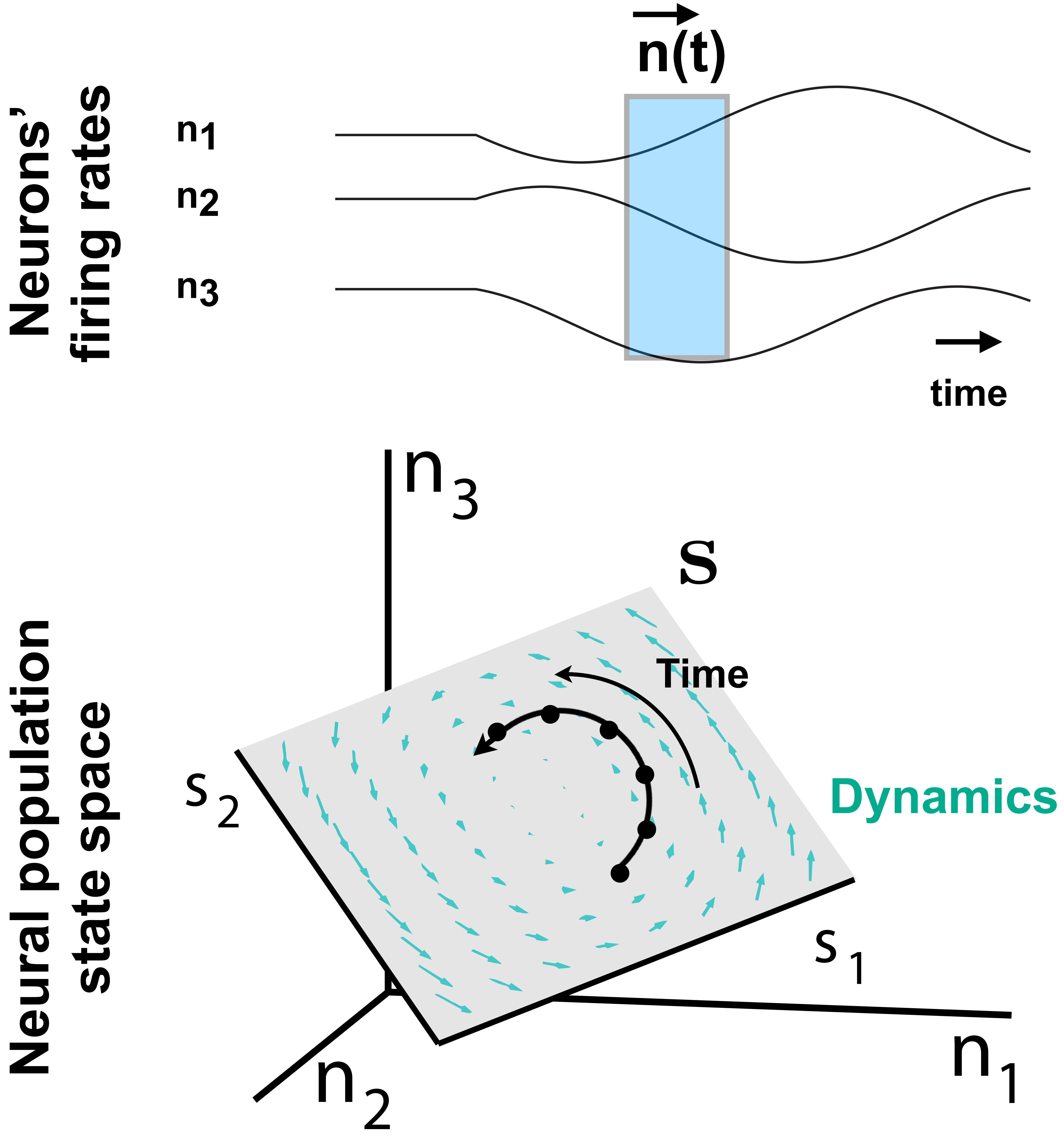


Adapted from Cunningham & Yu, *Nat Neuro* 2014

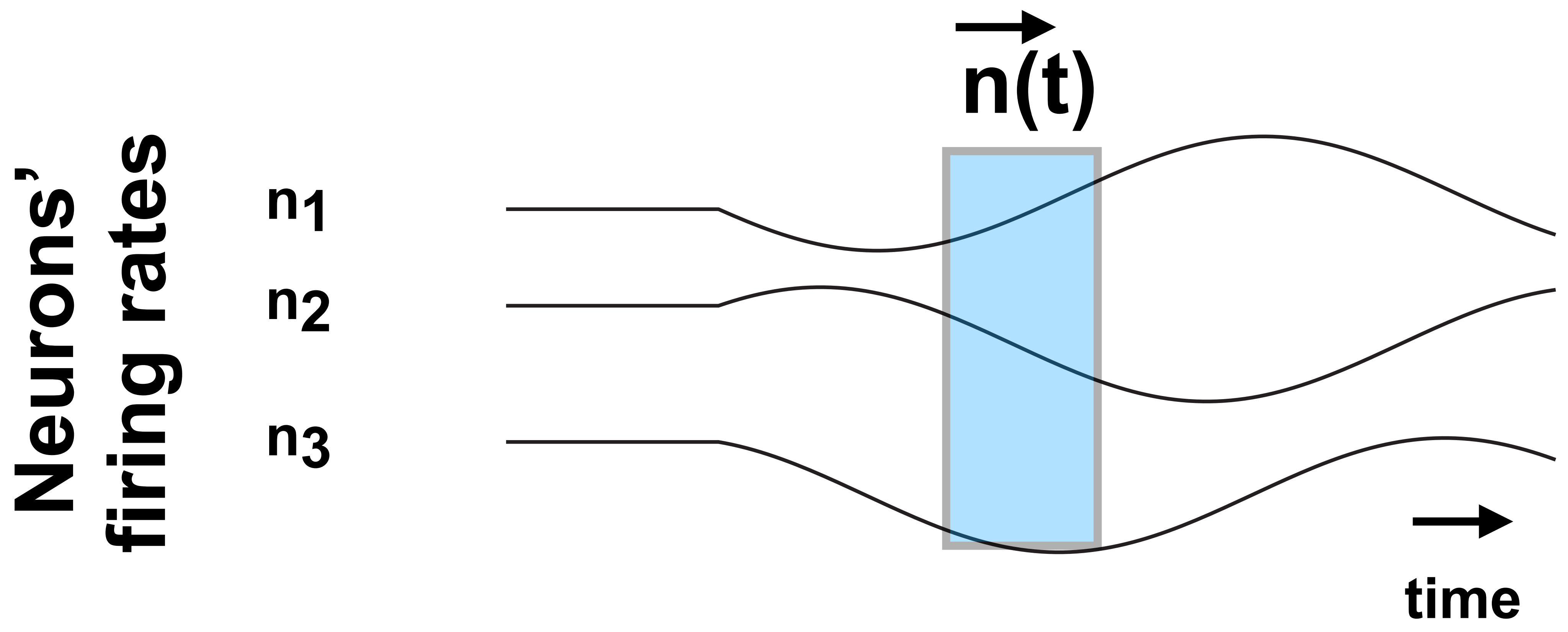
Neural population dynamics



Neural population dynamics

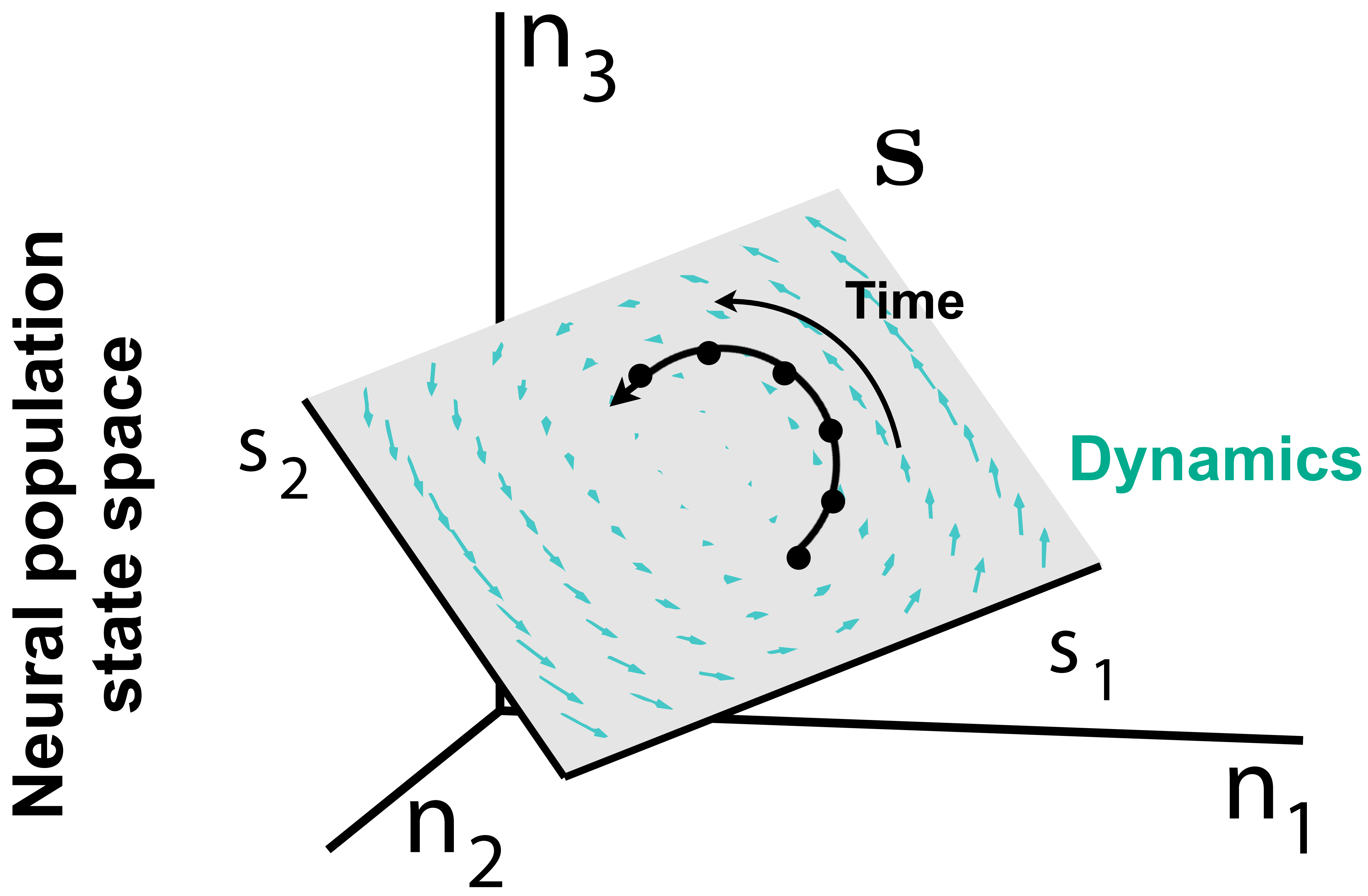


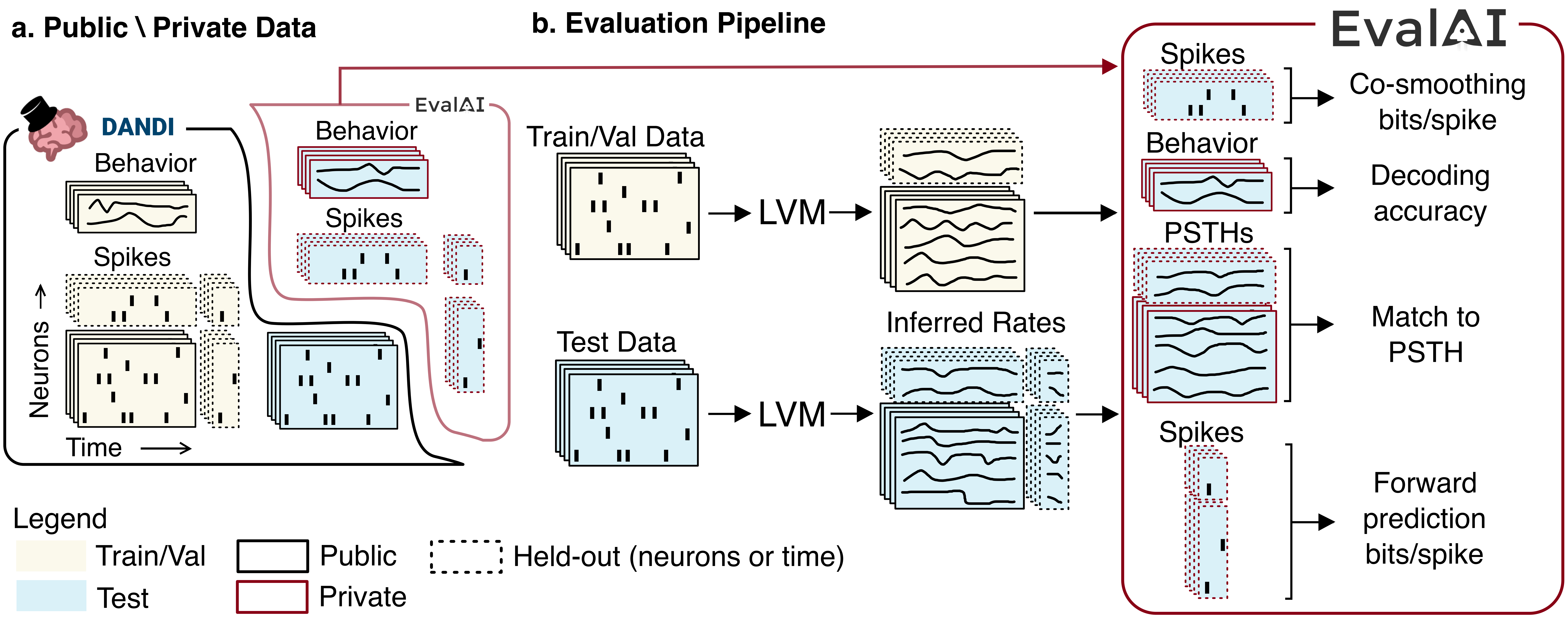
Neural population dynamics



Predictable activity - modeled by *autonomous* dynamics

$$\frac{d\mathrm{s}}{dt} = f(\mathrm{s})$$




a. Public \ Private Data
b. Evaluation Pipeline
EvalAI
DANDI
Behavior
Spikes
Neurons →
Time →
EvalAI
Behavior
Spikes
Train/Val Data
LVM
Test Data
LVM
Inferred Rates
Spikes
Co-smoothing bits/spike
Behavior
Decoding accuracy
PSTHs
Match to PSTH
Spikes
Forward prediction bits/spike
Legend
Train/Val
Test
Public
Private
Held-out (neurons or time)

Leaderboards - Phase I

Maze

B - Baseline   * - Private   V - Verified

Rank	Participant team	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)	Last submission at
1	AE Studio. (AESMTE3)	0.3676	0.9114	0.6683	0.2589	5 days ago
2	AE Studio (AESMTE1)	0.3599	0.9105	0.6641	0.2470	1 month ago
3	Hennequin Lab (iLQR-VAE)	0.3559	0.8840	0.6062	0.1480	7 days ago
4	Neural Latents (AutoLFADS) B	0.3364	0.9097	0.6360	0.2349	3 months ago
5	Churchland Lab (MINT) B	0.3304	0.9121	0.7496	0.2076	2 months ago
6	Neural Latents (NDT) B	0.3229	0.8862	0.5308	0.2206	3 months ago
7	NCLab	0.3039	0.7581	-1.0294	-0.0061	11 days ago
8	Neural Latents (SLDS) B	0.2249	0.7947	0.5330	-1.1579	3 months ago
9	Neural Latents (Spike smoothing) B	0.2109	0.6238	0.1853	E	4 months ago
10	Neural Latents (GPFA) B	0.1872	0.6399	0.5150	E	3 months ago

Random Target

Rank	Participant team	co-bps (↑)	vel R2 (↑)	fp-bps (↑)
1	AE Studio. (AESMTE3)	0.2053	0.6334	0.1344
2	AE Studio (AESMTE1)	0.1927	0.6627	0.1229
3	Neural Latents (AutoLFADS) B	0.1868	0.6167	0.1213
4	Neural Latents (NDT) B	0.1749	0.5656	0.0970
5	Churchland Lab (MINT) B	0.1676	0.5953	0.1012
6	Neural Latents (SLDS) B	0.1649	0.5206	0.0620
7	Neural Latents (GPFA) B	0.1548	0.5339	E
8	Neural Latents (Spike smoothing) B	0.1468	0.4142	E
9	NCLab (S2S)	0.1021	0.3117	E

Area 2

Rank	Participant team	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)
1	AE Studio. (AESMTE3)	0.2860	0.8999	0.7109	0.1603
2	AE Studio (AESMTE1)	0.2801	0.8675	0.6367	0.1523
3	Churchland Lab (MINT) B	0.2735	0.8877	0.9135	0.1483
4	Neural Latents (NDT) B	0.2623	0.8672	0.6619	0.1184
5	Neural Latents (AutoLFADS) B	0.2569	0.8492	0.6318	0.1505
6	NCLab	0.2388	0.7174	-0.1357	E
7	Neural Latents (SLDS) B	0.1960	0.7385	0.5740	0.0242
8	Neural Latents (GPFA) B	0.1680	0.5975	0.5289	E
9	Neural Latents (Spike smoothing) B	0.1544	0.5736	0.2084	E

DMFC

Rank	Participant team	co-bps (↑)	tp corr (↓)	psth R2 (↑)	fp-bps (↑)
1	AE Studio. (AESMTE3)	0.1886	-0.7601	0.6064	0.1828
2	Neural Latents (AutoLFADS) B	0.1829	-0.8248	0.6359	0.1844
3	Churchland Lab (MINT) B	0.1821	-0.6929	0.7013	0.1650
4	AE Studio (AESMTE1)	0.1733	-0.6189	0.5267	0.1511
5	Neural Latents (NDT) B	0.1720	-0.5624	0.4377	0.1404
6	Neural Latents (SLDS) B	0.1243	-0.5412	0.3372	-0.0418
7	Neural Latents (Spike smoothing) B	0.1202	-0.5139	0.2993	E
8	Neural Latents (GPFA) B	0.1176	-0.3763	0.2142	E
9	NE (SRNN)	-0.0078	-0.0319	-0.6191	E
10	NCLab (SMLseq)	-10.7694	0.6388	-403.9015	-10.4915

Leaderboards - Phase I

Maze

B - Baseline   * - Private   V - Verified

Rank	Participant team	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)	Last submission at
1	AE Studio. (AESMTE3)	0.3676	0.9114	0.6683	0.2589	5 days ago
2	AE Studio (AESMTE1)	0.3599	0.9105	0.6641	0.2470	1 month ago
3	Hennequin Lab (iLQR-VAE)	0.3559	0.8840	0.6062	0.1480	7 days ago
4	Neural Latents (AutoLFADS) B	0.3364	0.9097	0.6360	0.2349	3 months ago
5	Churchland Lab (MINT) B	0.3304	0.9121	0.7496	0.2076	2 months ago
6	Neural Latents (NDT) B	0.3229	0.8862	0.5308	0.2206	3 months ago
7	NCLab	0.3039	0.7581	-1.0294	-0.0061	11 days ago
8	Neural Latents (SLDS) B	0.2249	0.7947	0.5330	-1.1579	3 months ago
9	Neural Latents (Spike smoothing) B	0.2109	0.6238	0.1853	E	4 months ago
10	Neural Latents (GPFA) B	0.1872	0.6399	0.5150	E	3 months ago

Random Target

Rank	Participant team	co-bps (↑)	vel R2 (↑)	fp-bps (↑)
1	AE Studio. (AESMTE3)	0.2053	0.6334	0.1344
2	AE Studio (AESMTE1)	0.1927	0.6627	0.1229
3	Neural Latents (AutoLFADS) B	0.1868	0.6167	0.1213
4	Neural Latents (NDT) B	0.1749	0.5656	0.0970
5	Churchland Lab (MINT) B	0.1676	0.5953	0.1012
6	Neural Latents (SLDS) B	0.1649	0.5206	0.0620
7	Neural Latents (GPFA) B	0.1548	0.5339	E
8	Neural Latents (Spike smoothing) B	0.1468	0.4142	E
9	NCLab (S2S)	0.1021	0.3117	E

Area 2

Rank	Participant team	co-bps (↑)	vel R2 (↑)	psth R2 (↑)	fp-bps (↑)
1	AE Studio. (AESMTE3)	0.2860	0.8999	0.7109	0.1603
2	AE Studio (AESMTE1)	0.2801	0.8675	0.6367	0.1523
3	Churchland Lab (MINT) B	0.2735	0.8877	0.9135	0.1483
4	Neural Latents (NDT) B	0.2623	0.8672	0.6619	0.1184
5	Neural Latents (AutoLFADS) B	0.2569	0.8492	0.6318	0.1505
6	NCLab	0.2388	0.7174	-0.1357	E
7	Neural Latents (SLDS) B	0.1960	0.7385	0.5740	0.0242
8	Neural Latents (GPFA) B	0.1680	0.5975	0.5289	E
9	Neural Latents (Spike smoothing) B	0.1544	0.5736	0.2084	E

DMFC

Rank	Participant team	co-bps (↑)	tp corr (↓)	psth R2 (↑)	fp-bps (↑)
1	AE Studio. (AESMTE3)	0.1886	-0.7601	0.6064	0.1828
2	Neural Latents (AutoLFADS) B	0.1829	-0.8248	0.6359	0.1844
3	Churchland Lab (MINT) B	0.1821	-0.6929	0.7013	0.1650
4	AE Studio (AESMTE1)	0.1733	-0.6189	0.5267	0.1511
5	Neural Latents (NDT) B	0.1720	-0.5624	0.4377	0.1404
6	Neural Latents (SLDS) B	0.1243	-0.5412	0.3372	-0.0418
7	Neural Latents (Spike smoothing) B	0.1202	-0.5139	0.2993	E
8	Neural Latents (GPFA) B	0.1176	-0.3763	0.2142	E
9	NE (SRNN)	-0.0078	-0.0319	-0.6191	E
10	NCLab (SMLseq)	-10.7694	0.6388	-403.9015	-10.4915

EvalAI