BioRoLe
2 0 2 2

1 Biologically-inspired approaches to robotics

Embodied artificial intelligence, neurorobotics and biorobotics

Bionics

Biomimetics

Biologically-inspired Robotics

BioRobotics

Embodied Artificial Intelligence

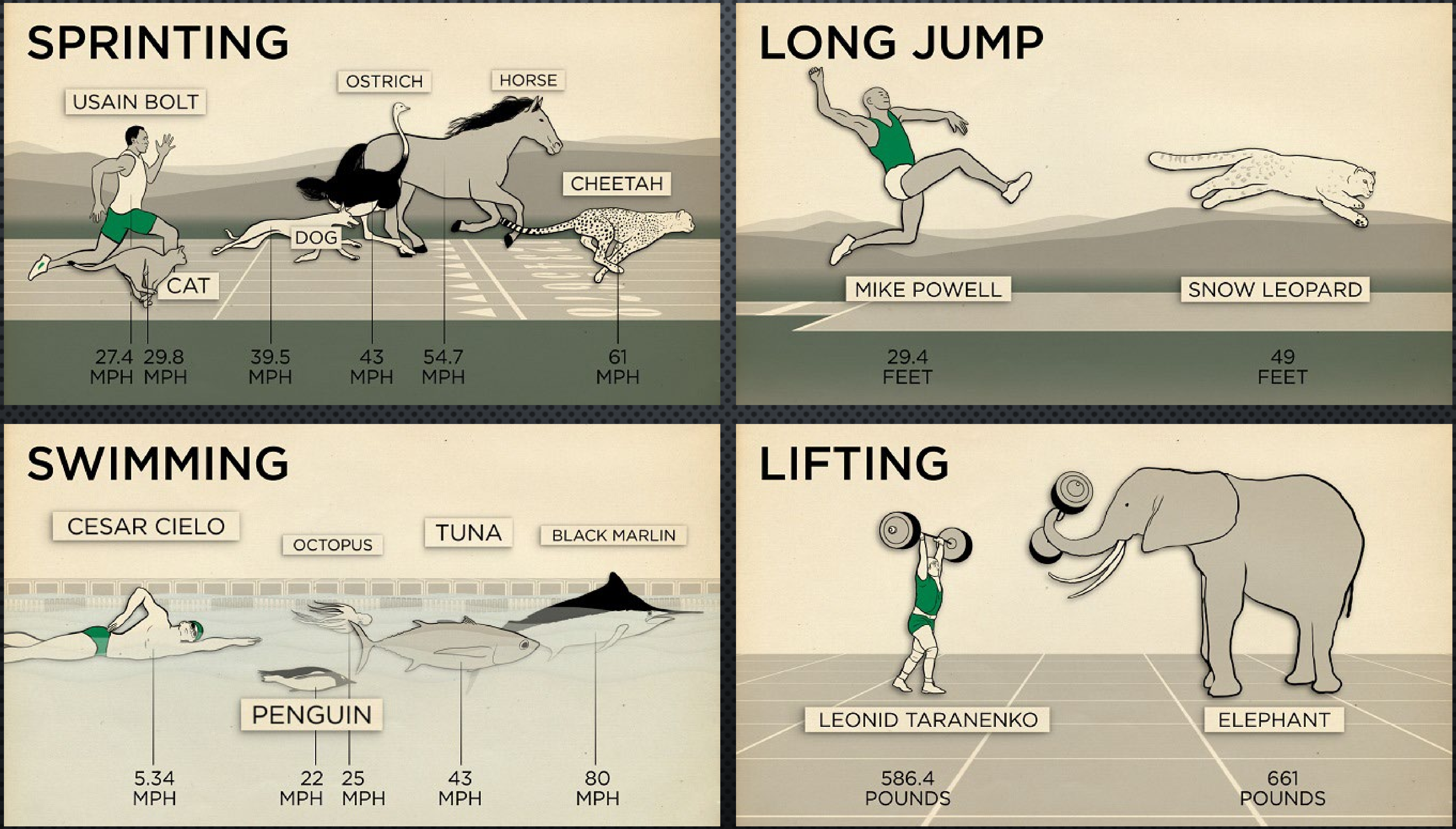
Neurorobotics

Biologically-inspired robotics

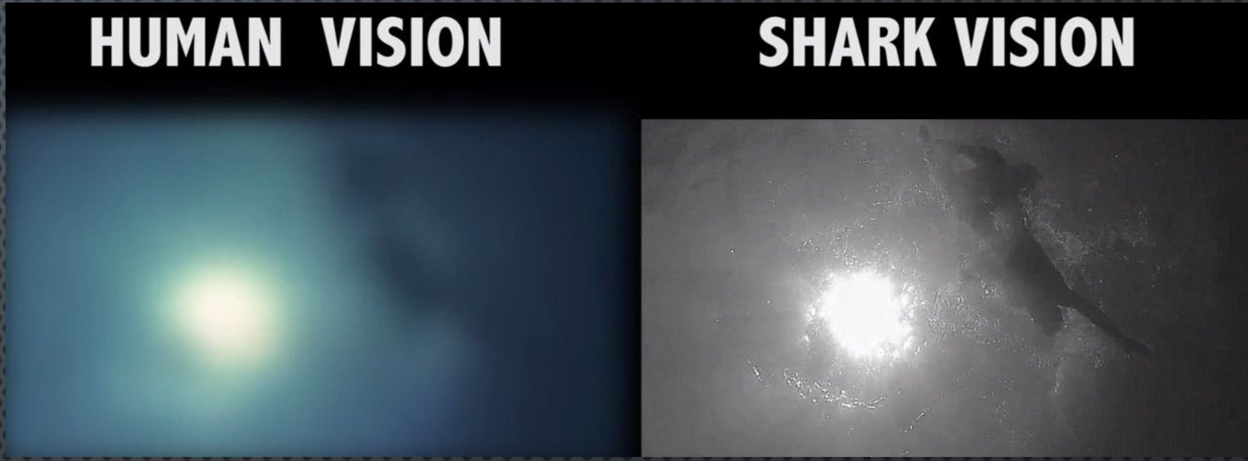
using biological systems as inspiration for robotic design…

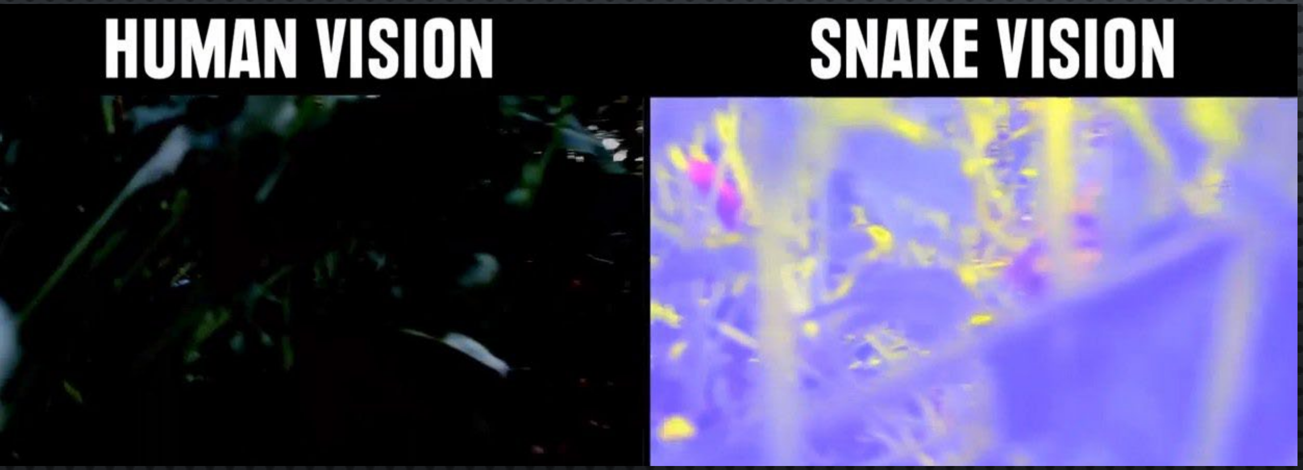
Images: Pfeifer, R. & Bongard, J. C. (2007), “How the Body Shapes the Way We Think: A New View of Intelligence”, MIT Press, Cambridge, MA

Why study animals?

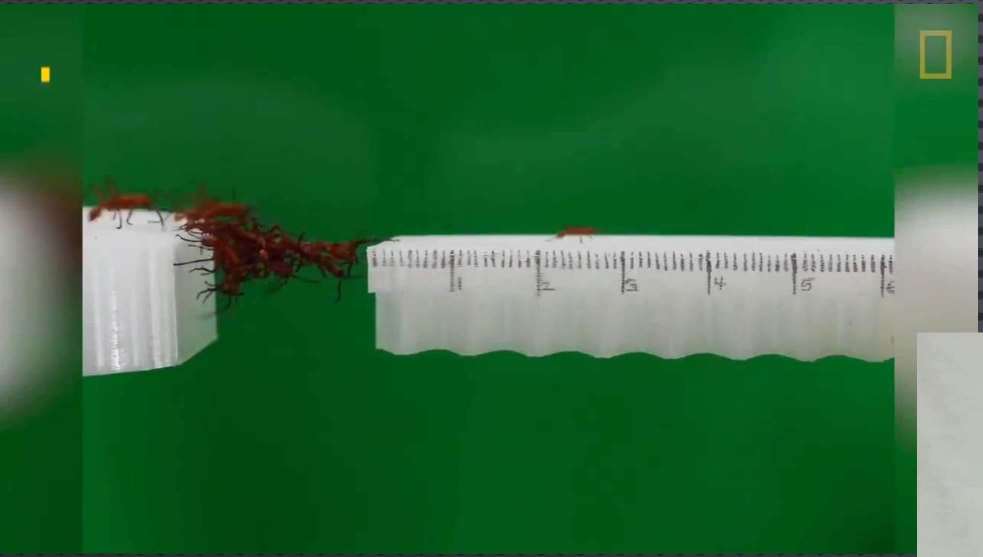


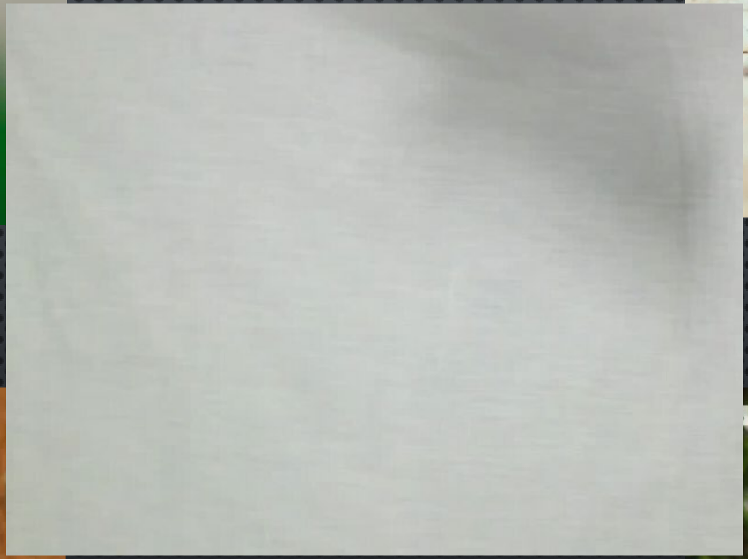
Why study animals?





Why study animals?

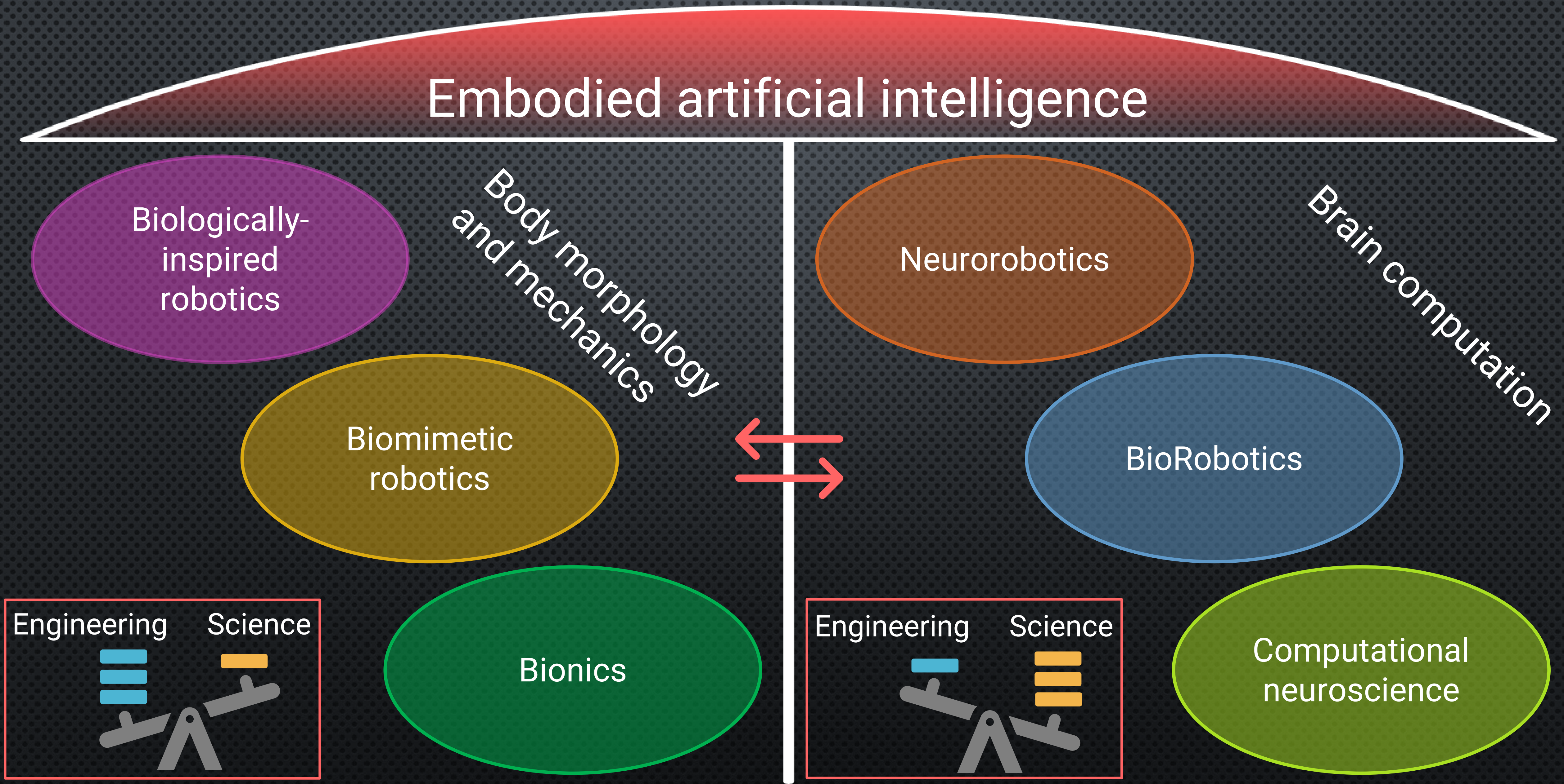




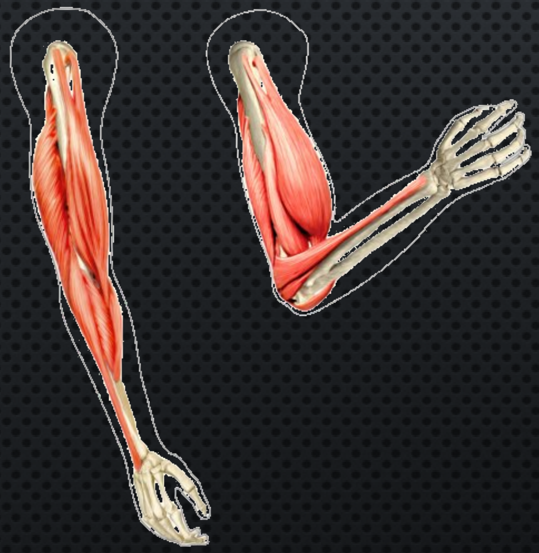
Still not convinced? Try doing this…

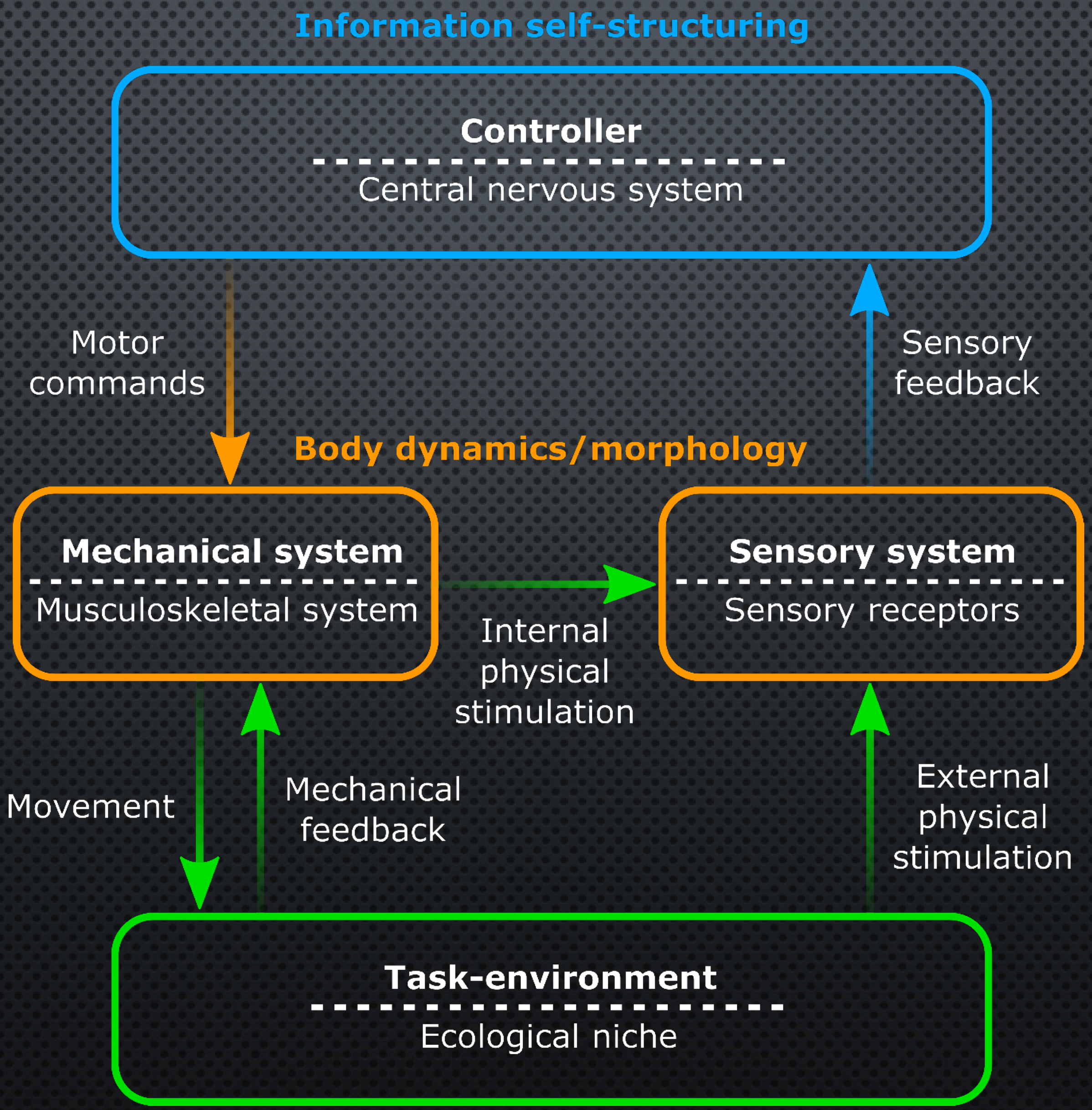
Or this…

The bigger (and confusing) picture



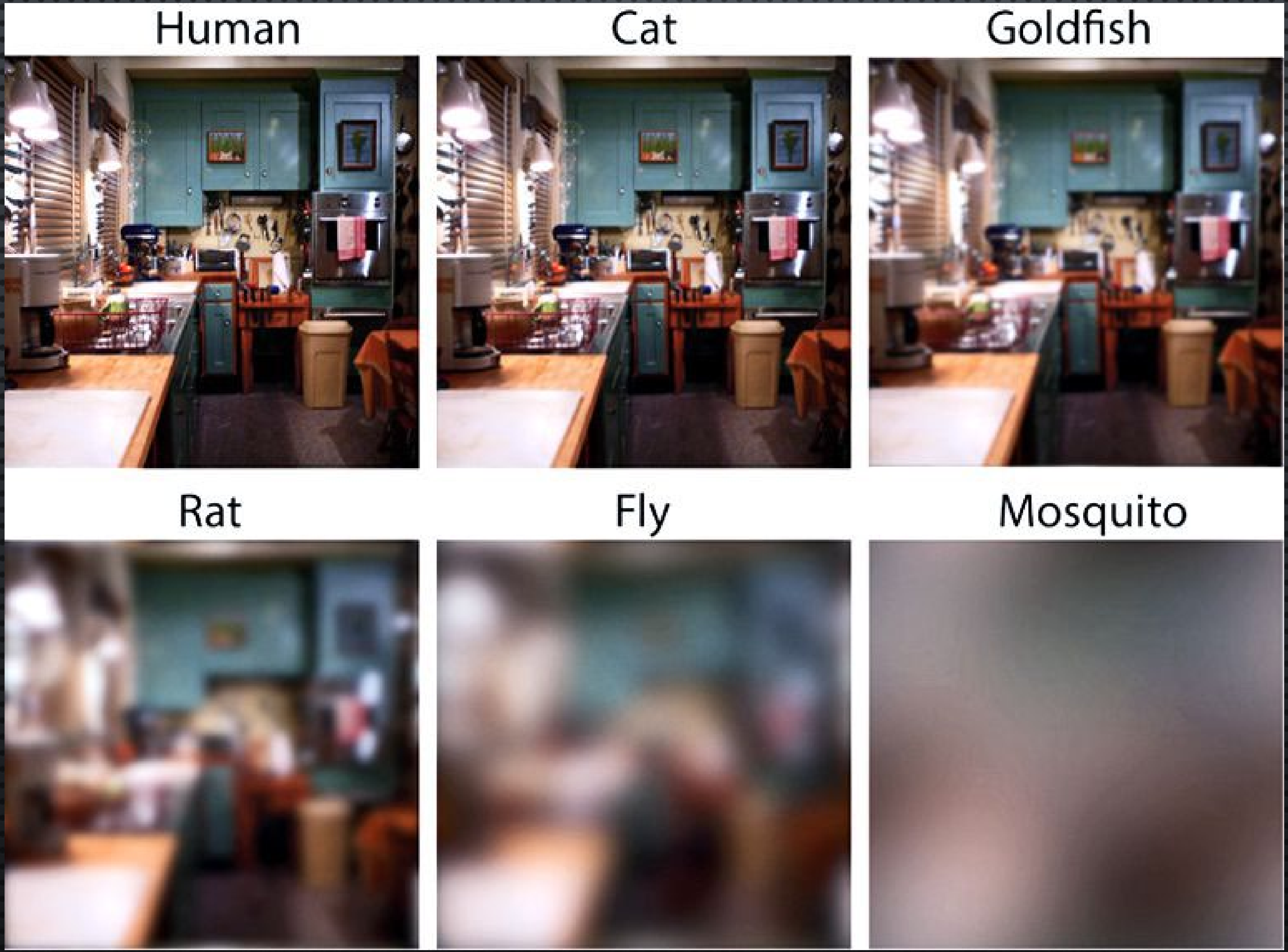
Embodied artificial intelligence



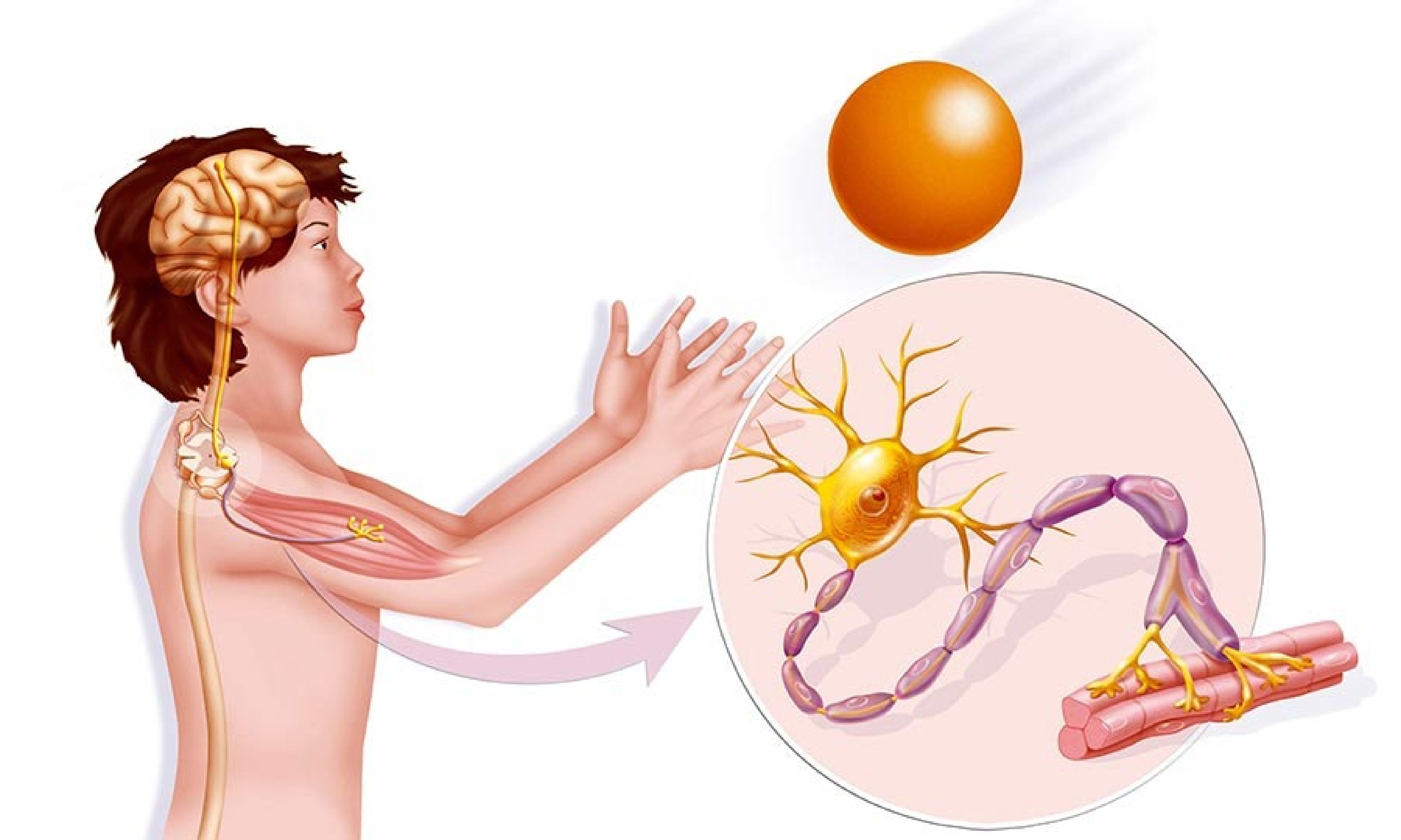


- Ecological niche is the role a species plays in the ecosystem and includes
  - the habitat (environment in which it is situated)
  - the way it interacts with other species as well as the environment via sensing and acting

Why is the environment / ecological niche important?



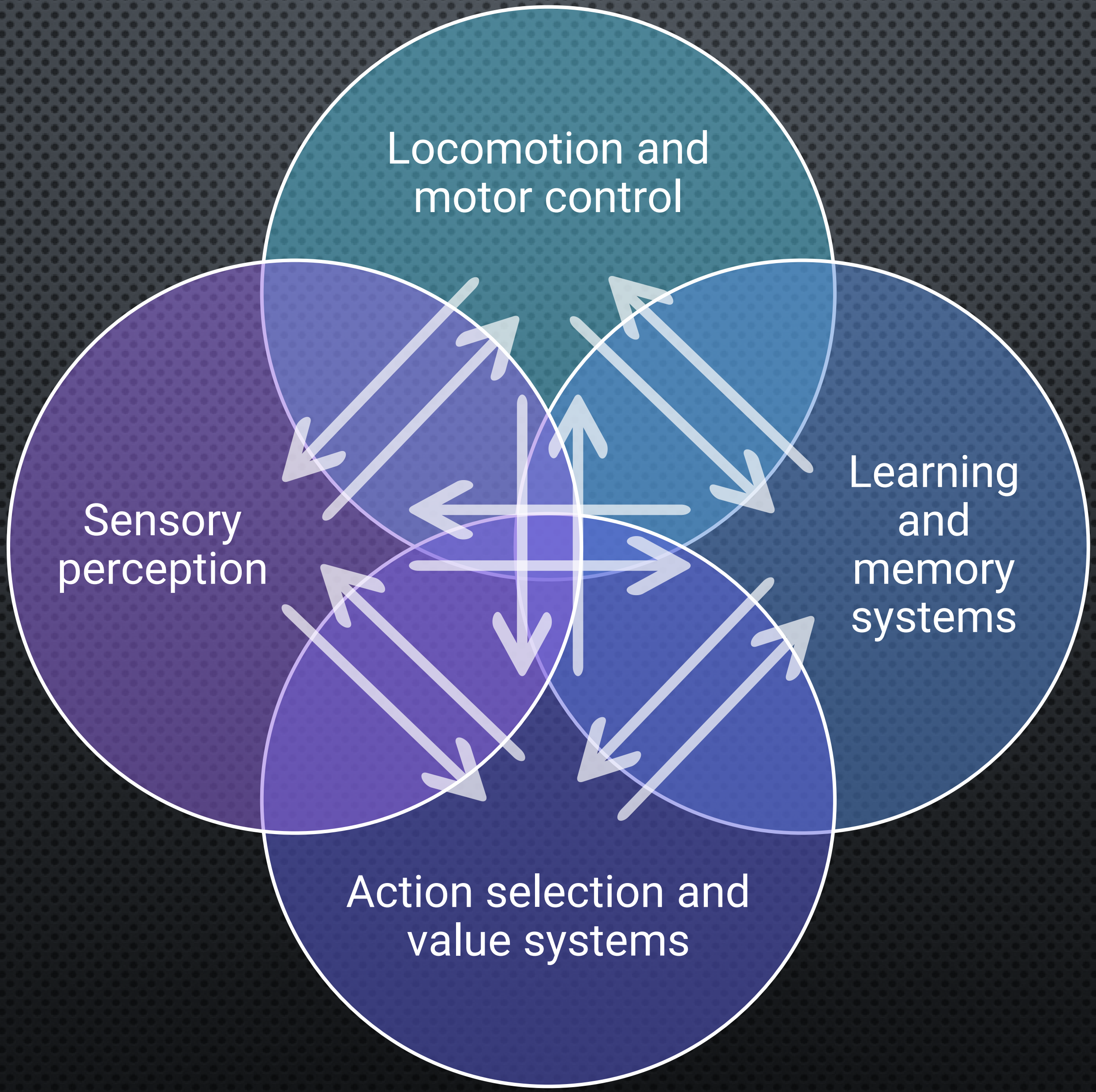
Sensorimotor control and learning: a holistic view of embodied AI



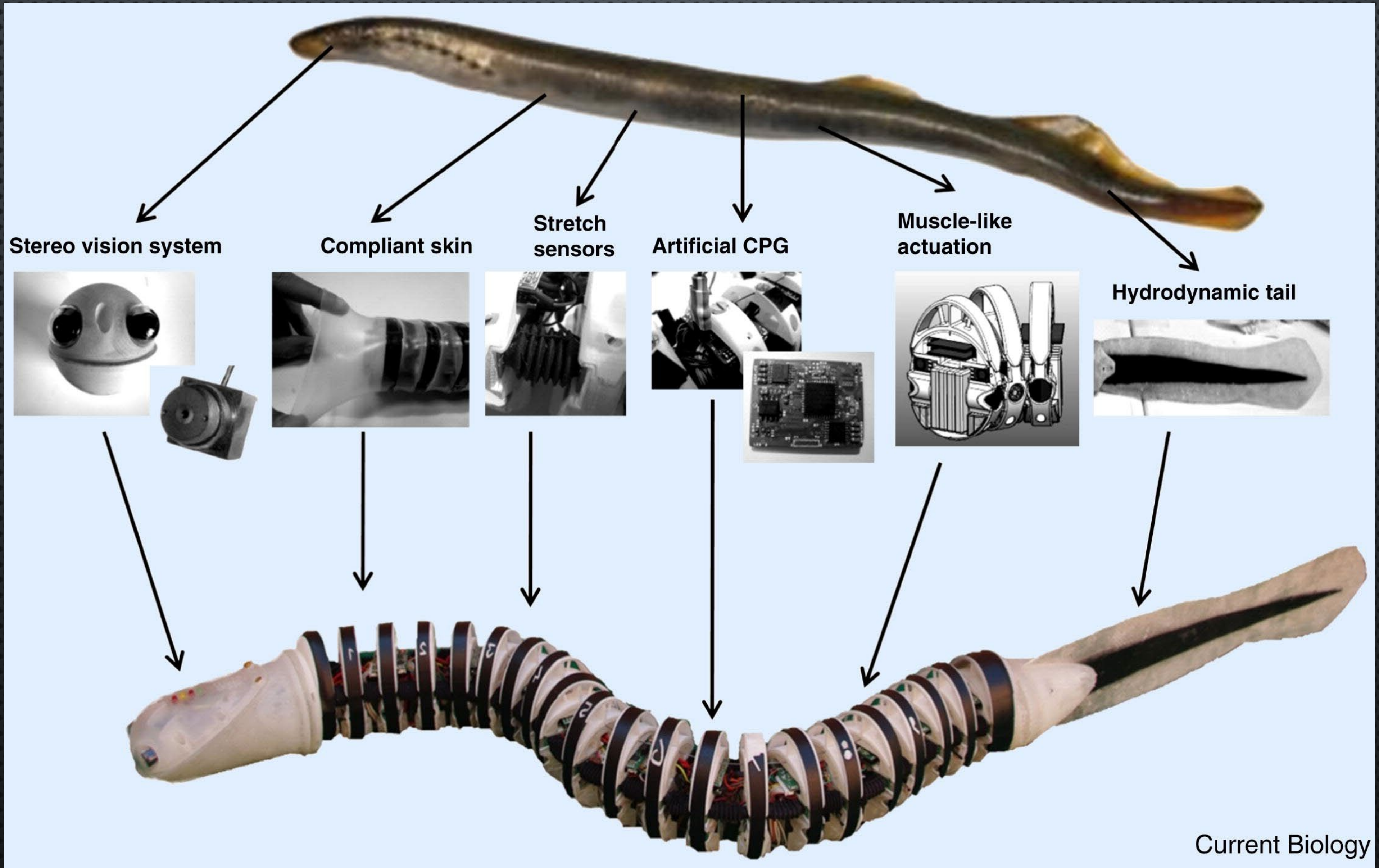
Neurorobotics

- Neurorobotics deals with the study and application of science and technology of embodied, autonomous, brain-inspired algorithms
- A neurorobot is a robot
  - that engages in a behavioural task
  - that is situated in a real-world environment
  - that has a means to interact with other agents via sensing and acting
  - whose behaviour is controlled by a simulated nervous system having a design that reflects, at some level, the brain's architecture and dynamics

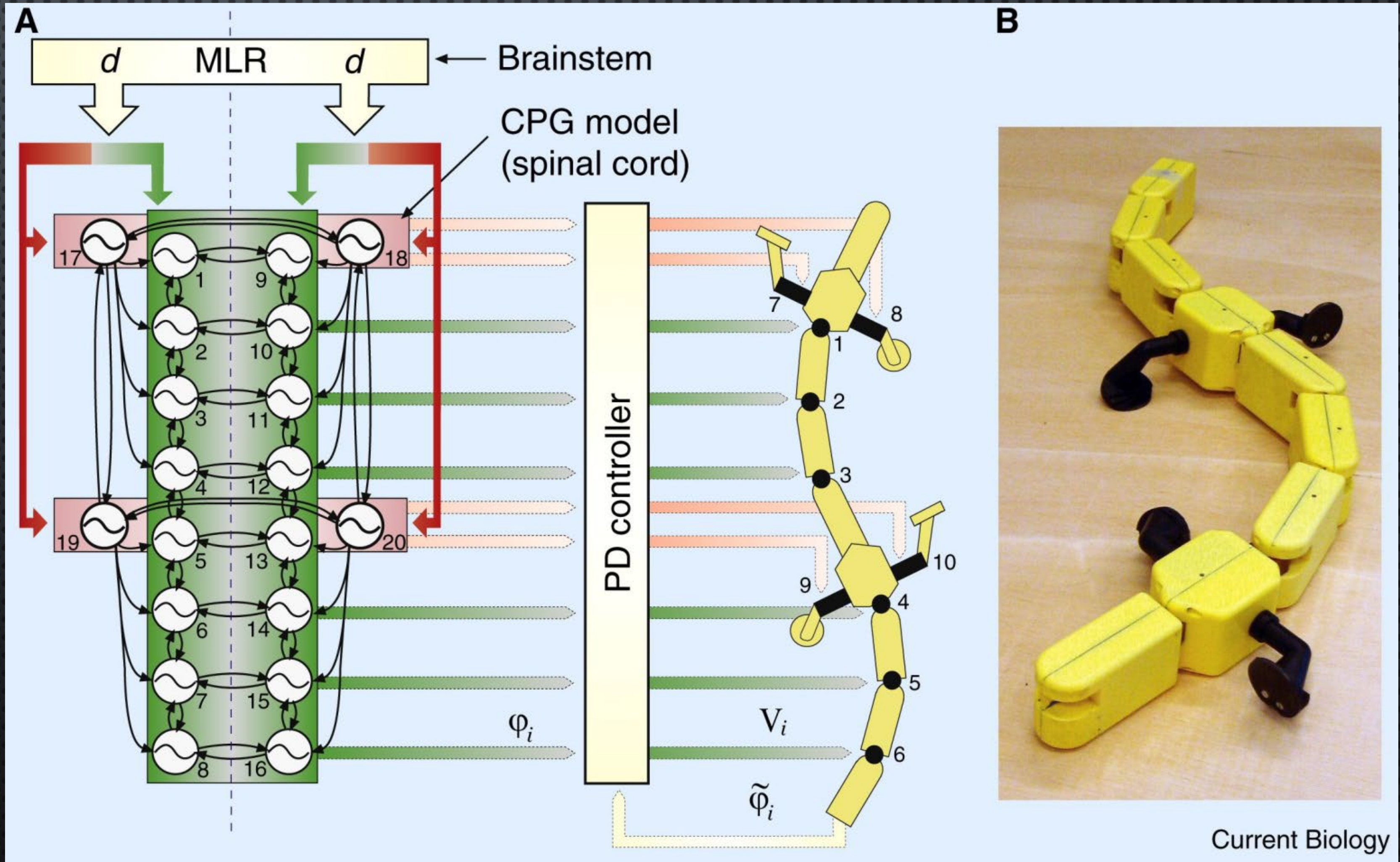
Classes of neurorobot models



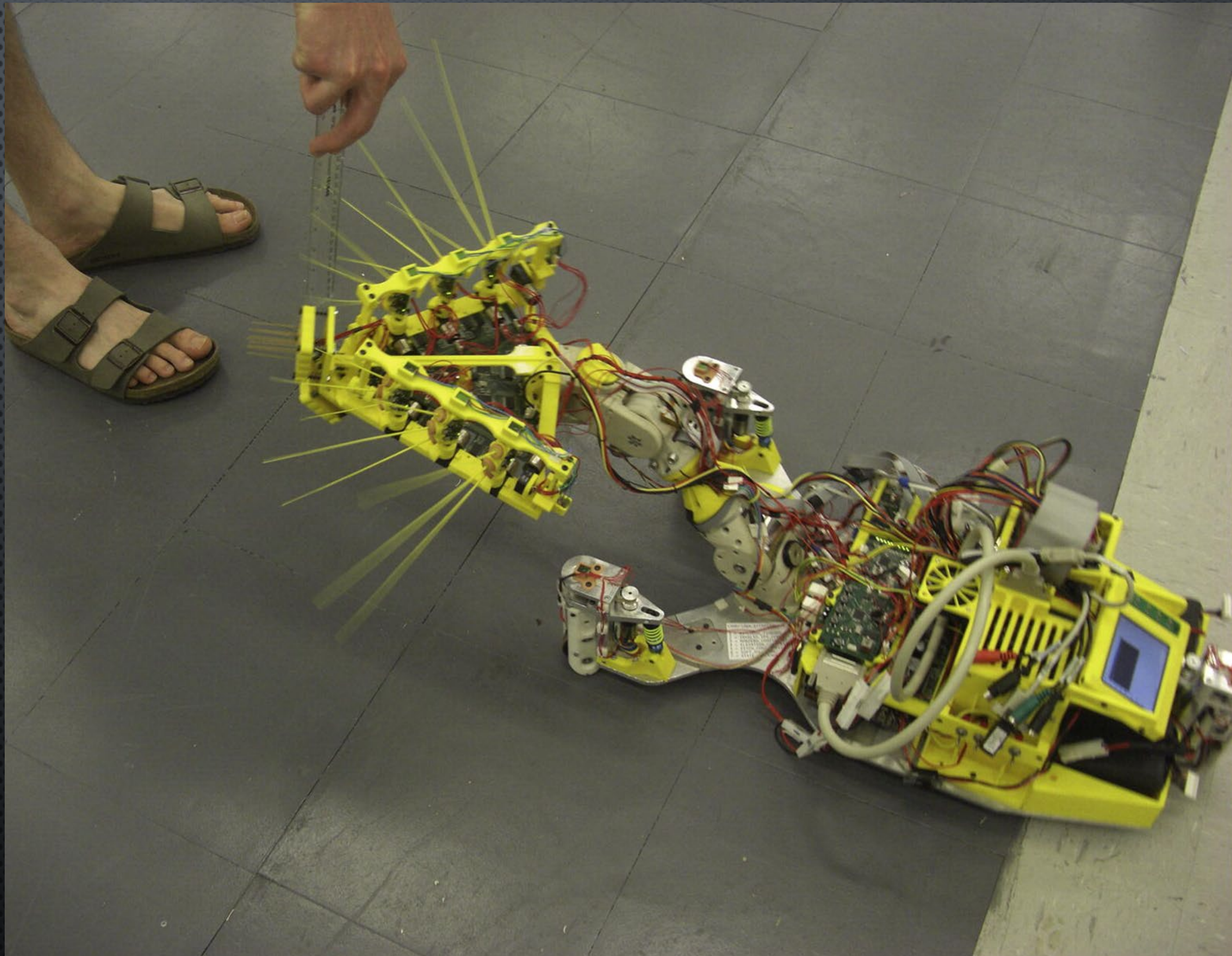
Models for locomotion



Models for locomotion

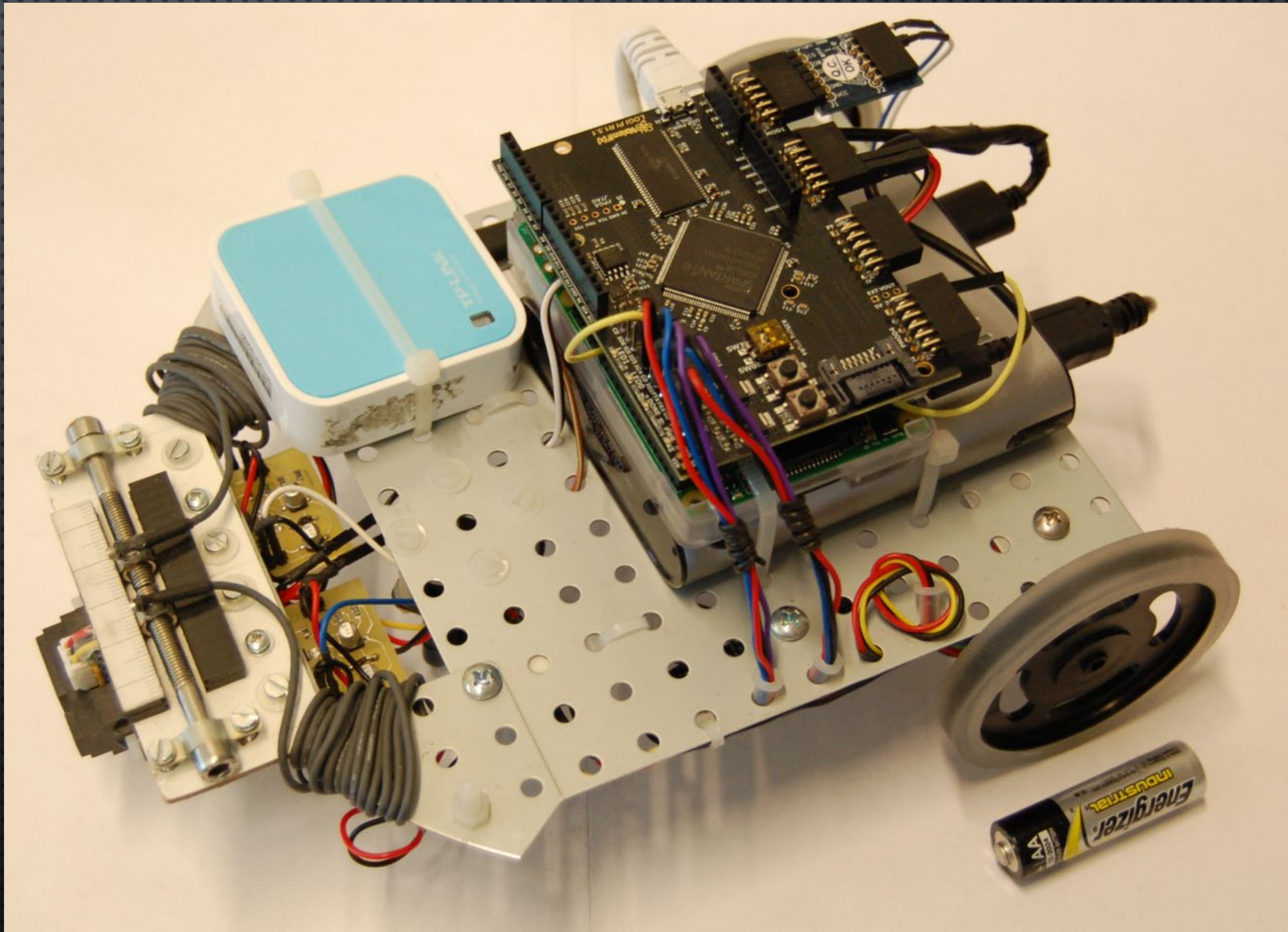


Models for sensorimotor integration



Models for perception

Models for perception



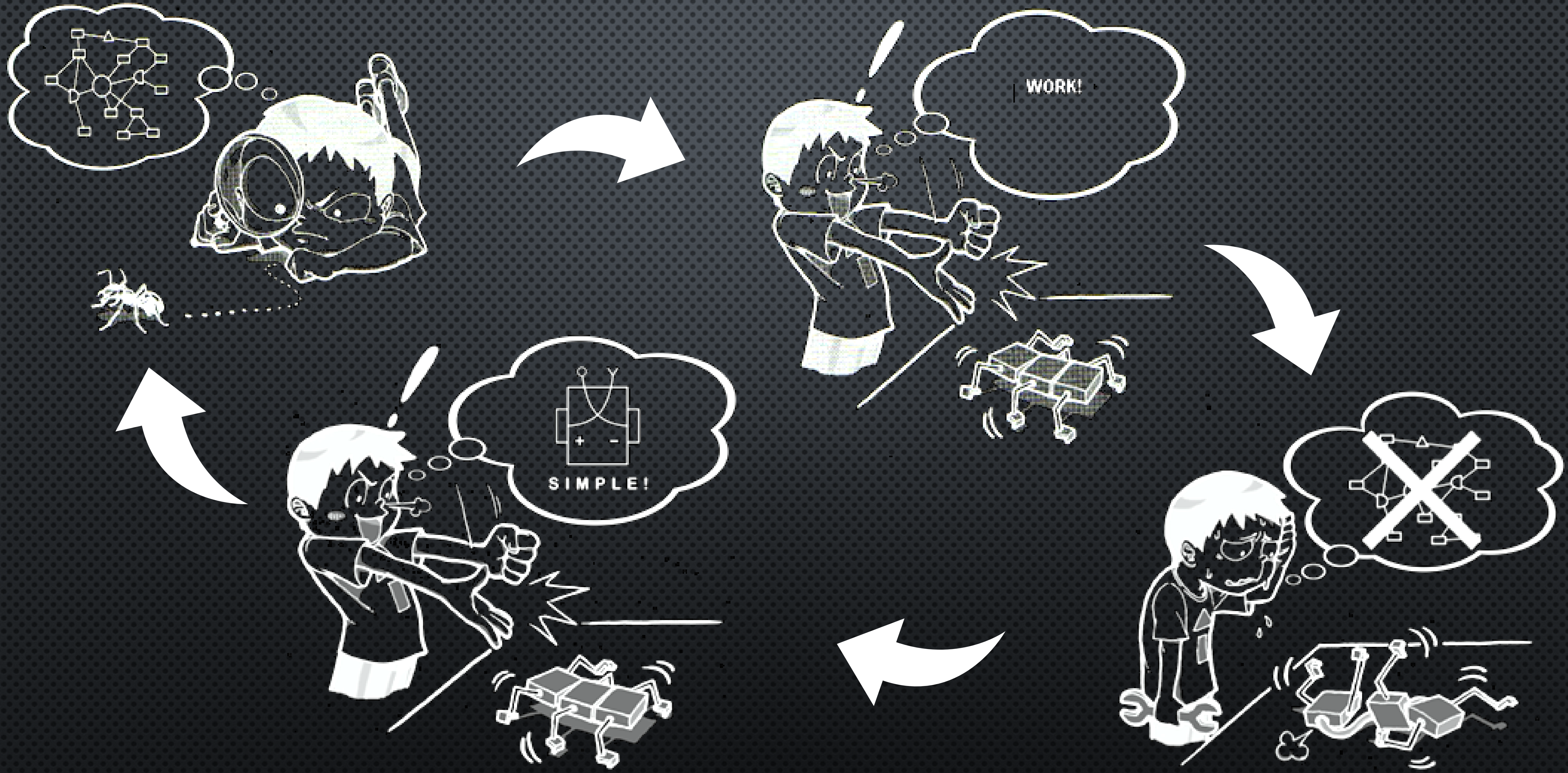
BioRobotics

using biological systems as inspiration for robotic design…

using robotic systems to understand biological design…

Images: Pfeifer, R. & Bongard, J. C. (2007), "How the Body Shapes the Way We Think: A New View of Intelligence", MIT Press, Cambridge, MA

Reality check



Images: Pfeifer, R. & Bongard, J. C. (2007), "How the Body Shapes the Way We Think: A New View of Intelligence", MIT Press, Cambridge, MA

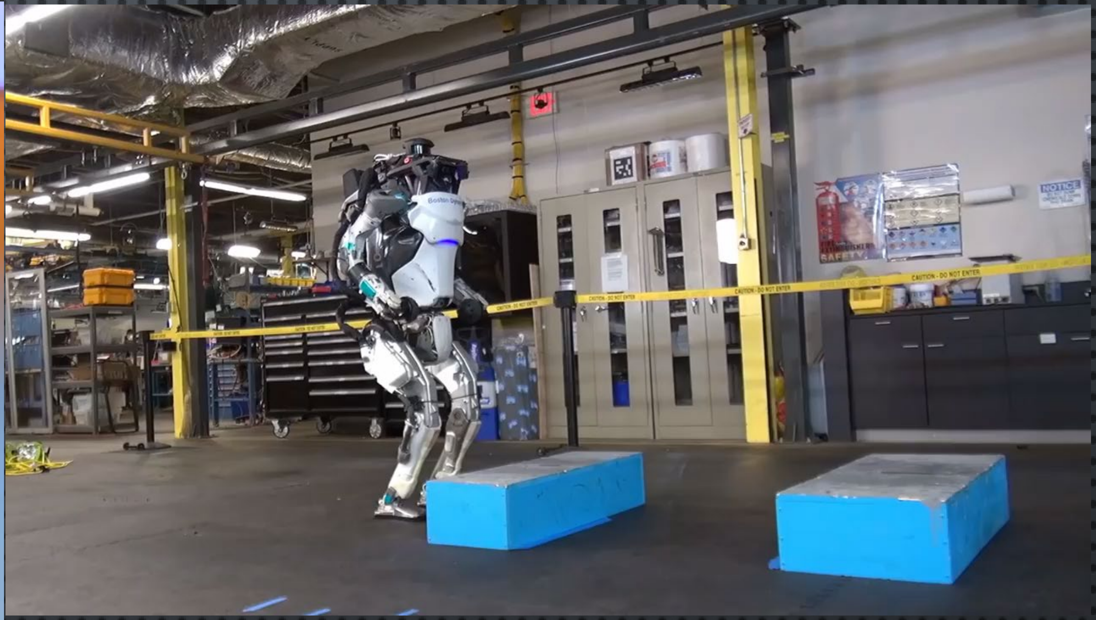
A world full of model organisms!

World's first biologically-inspired robots

Elmer and Elsie (ELectroMEchanical Robot, Light-Sensitive)

William Grey Walter

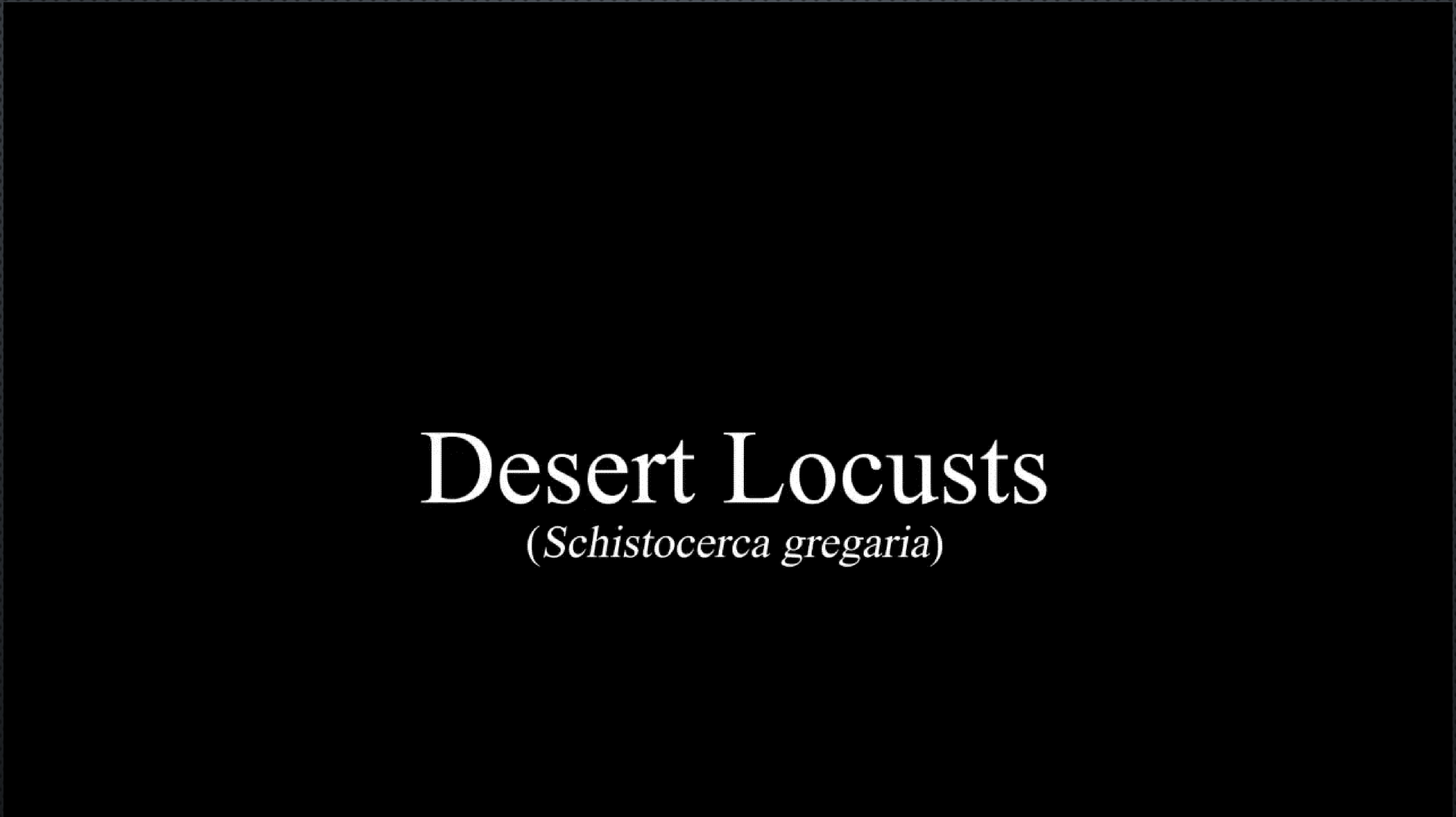
Cool robot videos for motivation…



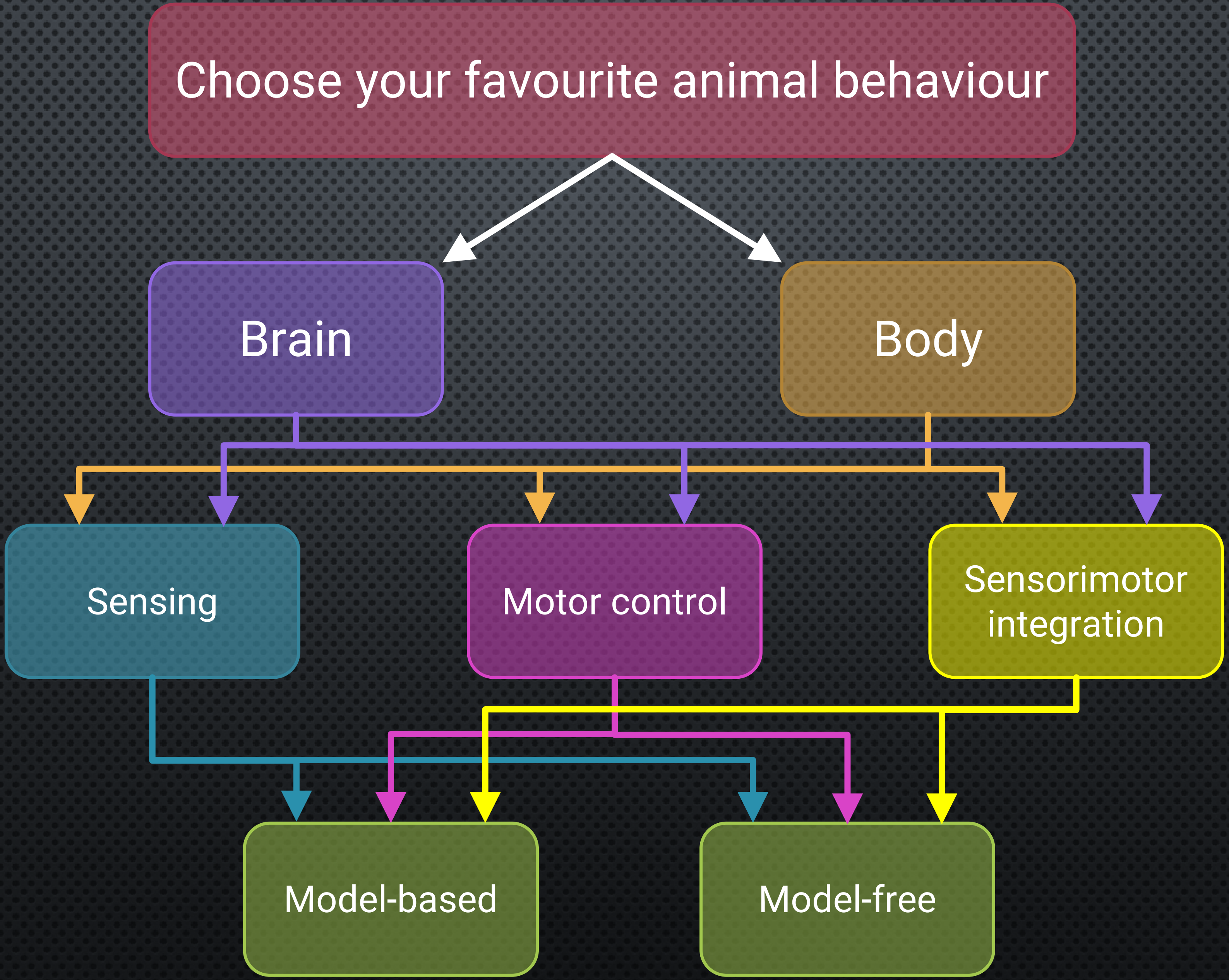
T-800: the ultimate biologically-inspired robot

Dance break...

Reality check…



So how and where do I start?



Meet the robot...

8-microphone array
- Alexa voice recognition
- sound direction estimation

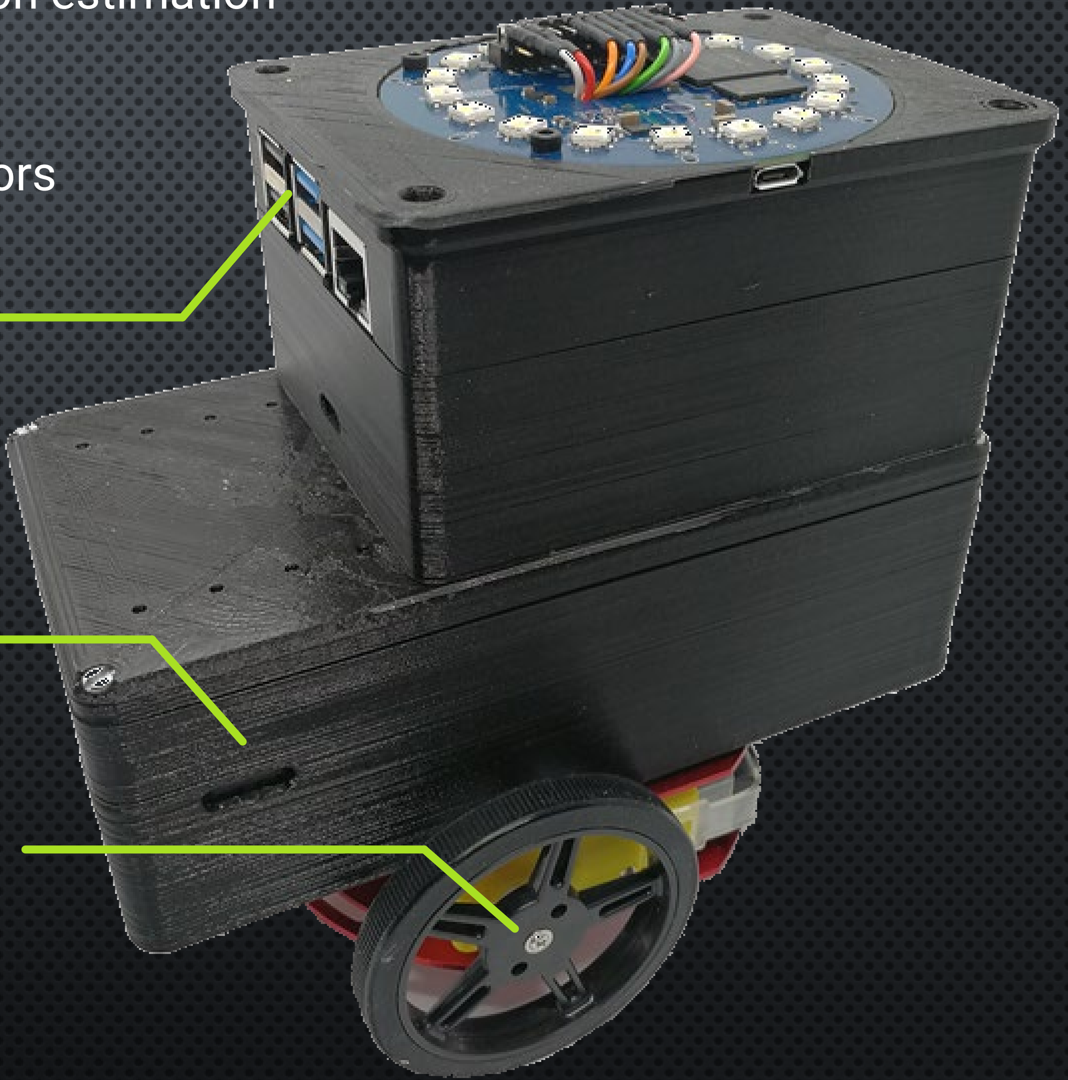
18-LED array
- visual indicators

Raspberry Pi 4

RGB camera

30000 mAh rechargable battery

Differential drive kinematics



Questions?

2 Biological neurons and their functional engineering models

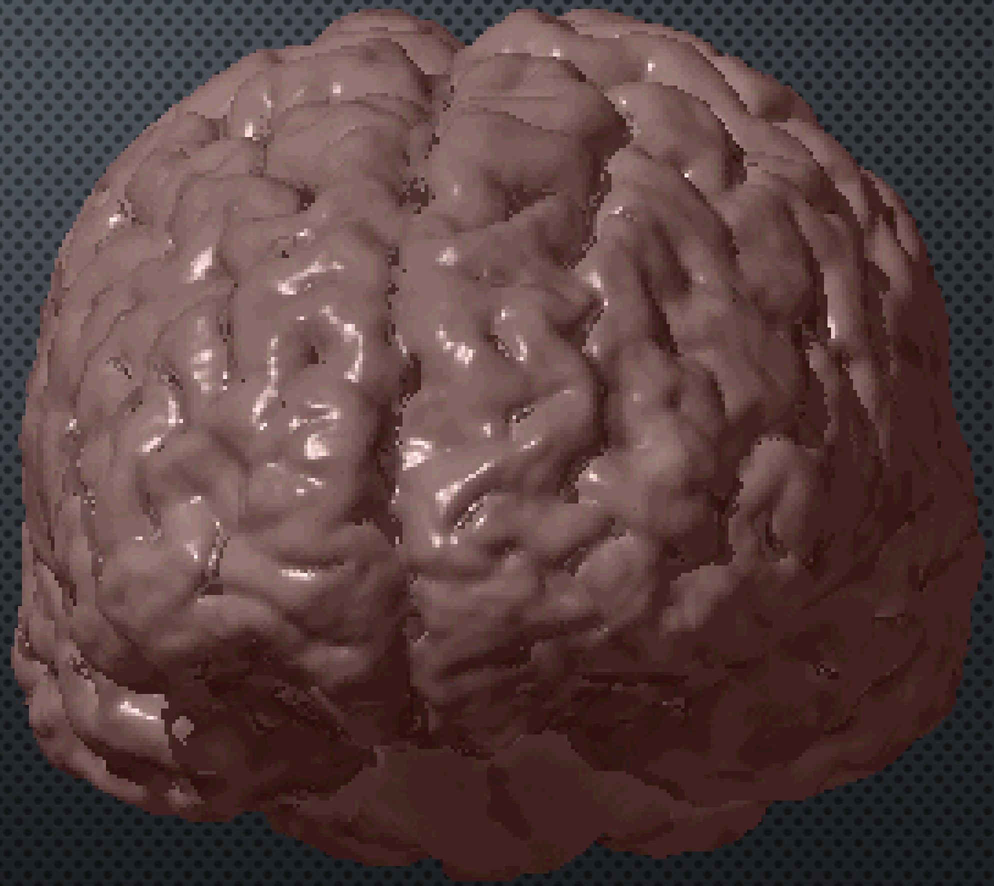
Neurobiology ahead
(get used to it)

The human brain

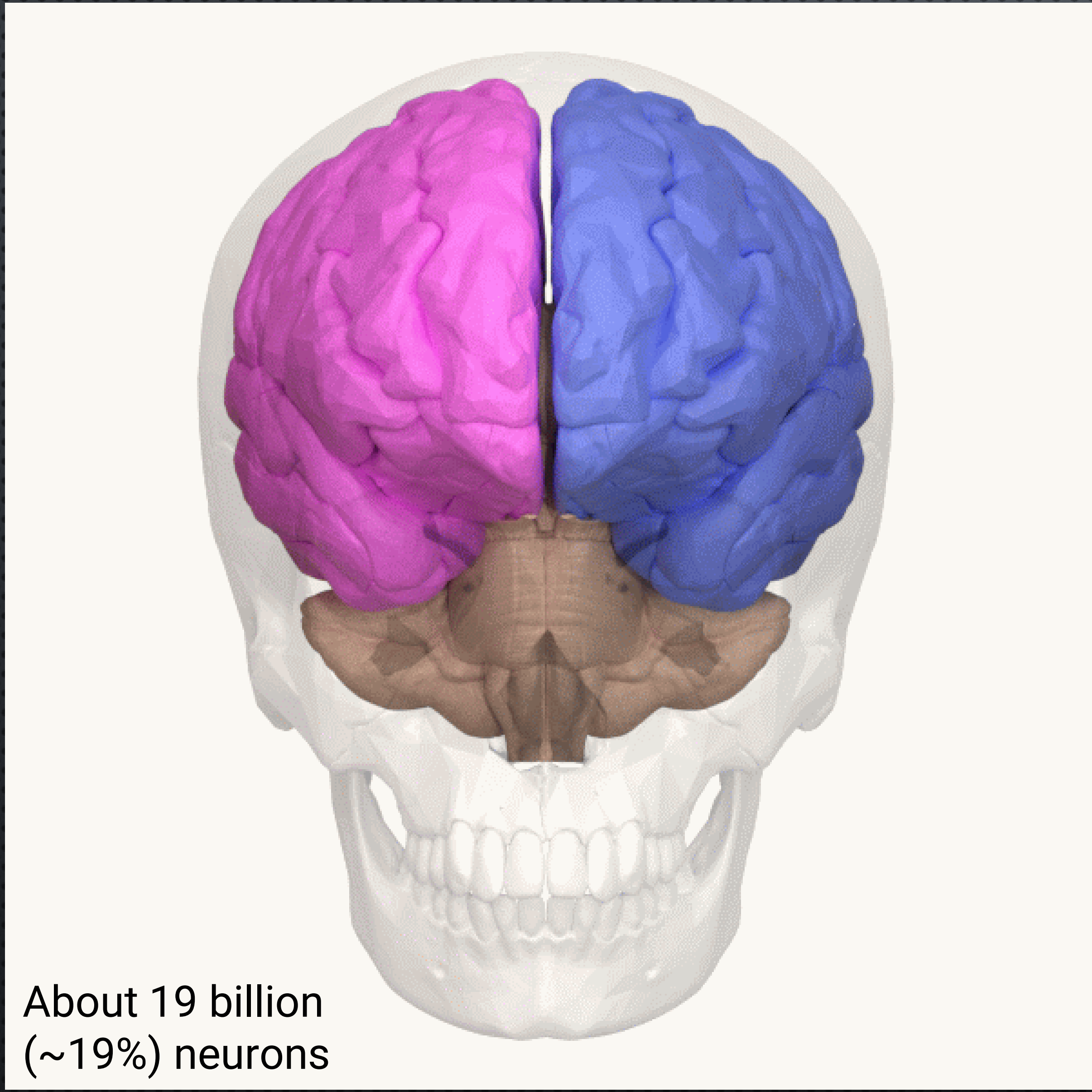
- Represents only 2% of the body weight (1.2 - 1.4 kg) but receives 15% of the cardiac output, 20% of total body oxygen consumption, and 25% of total body glucose utilization

- Composed of neurons, glial cells, neural stem cells, and blood vessels

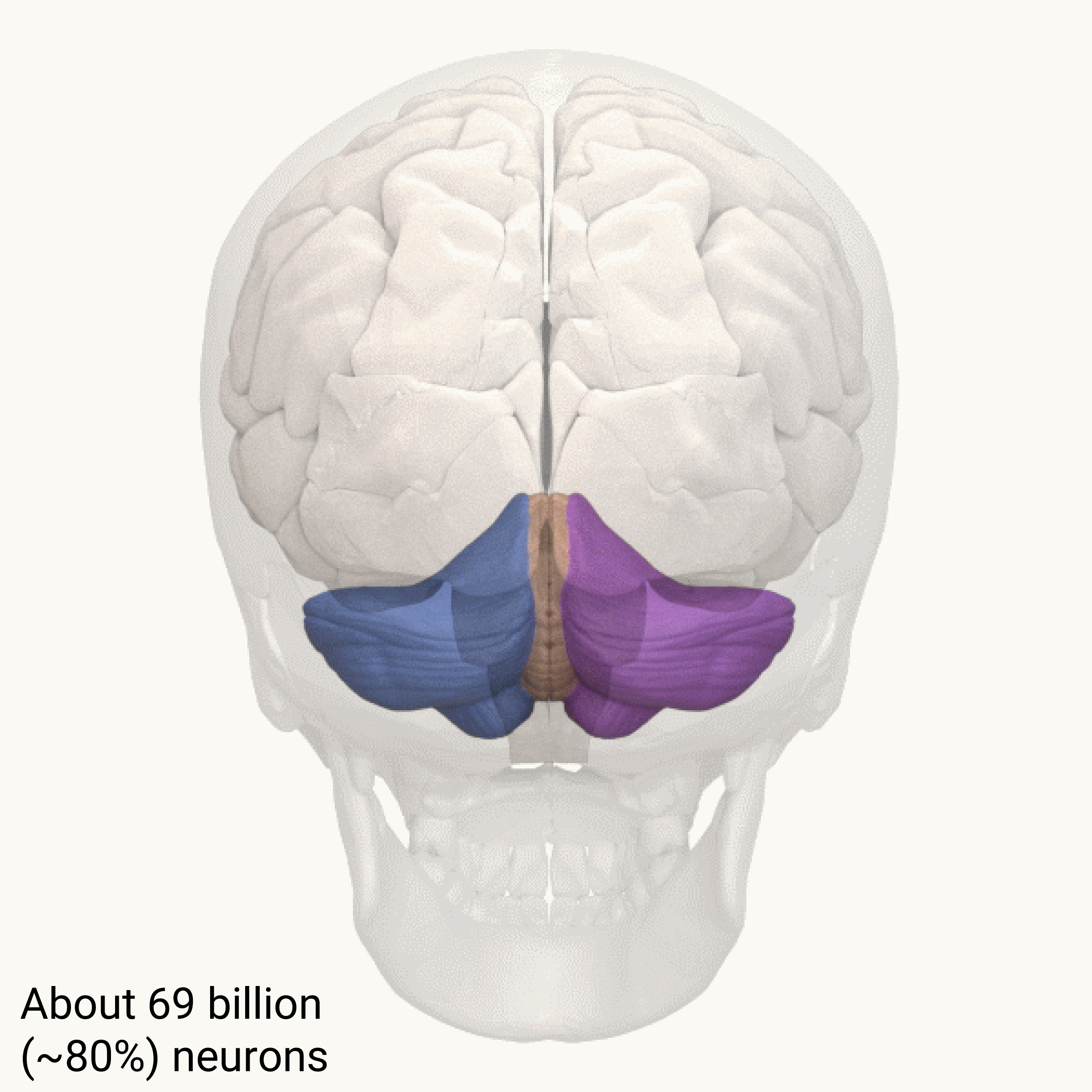
- Contains about 86 ± 8 billion neurons and 85 ± 10 billion non-neuronal cells



What makes the brain so special?

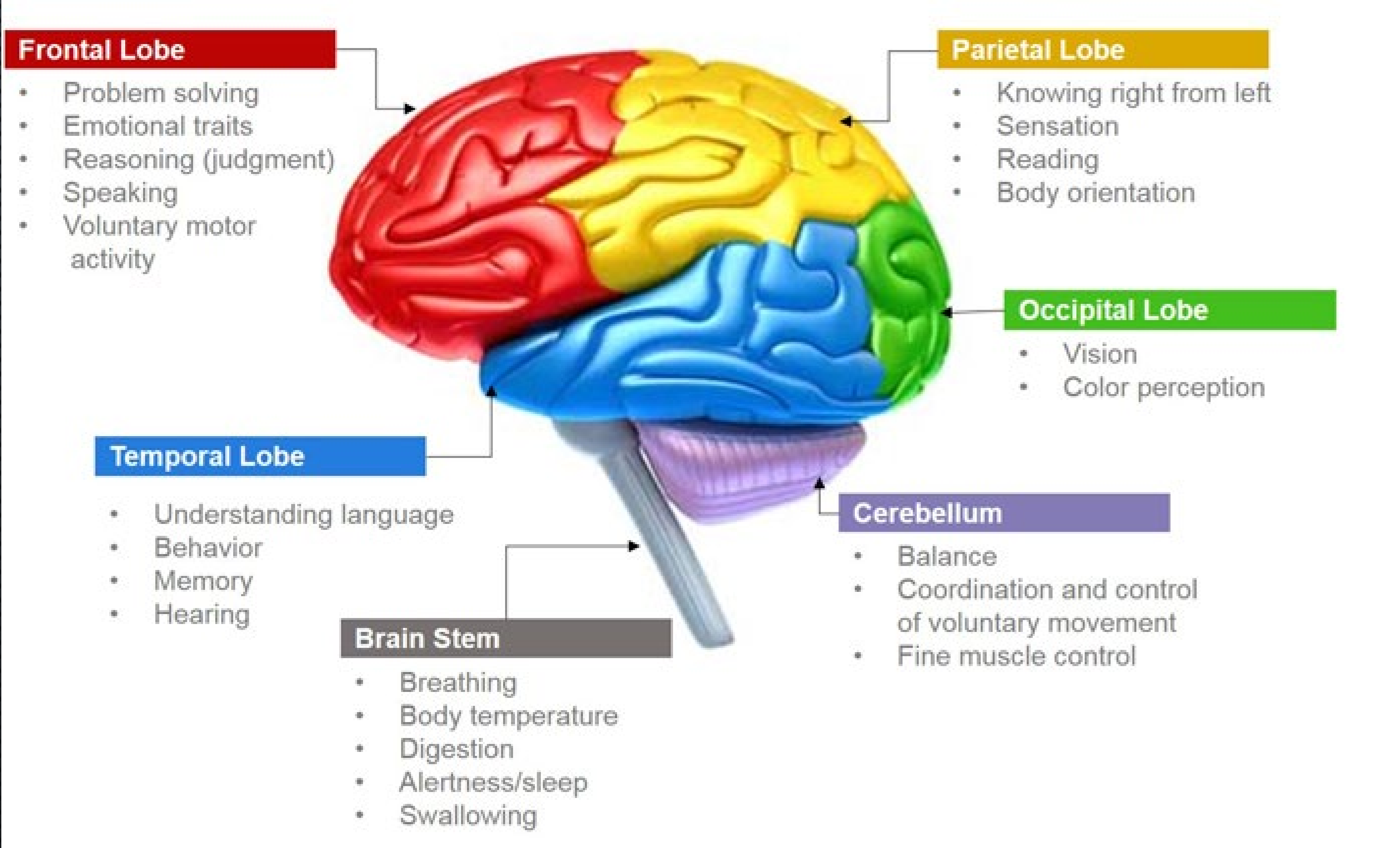


Cerebrum

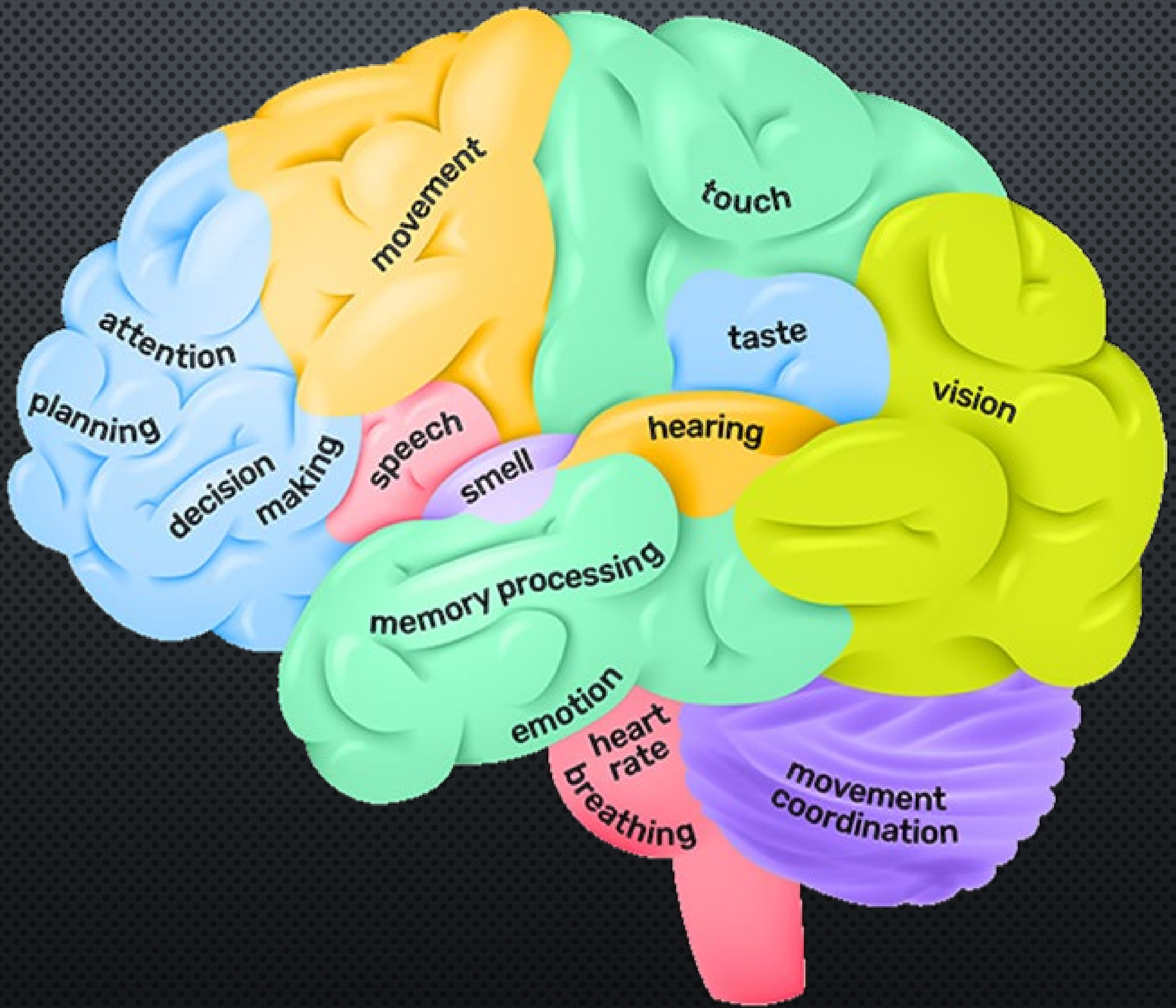


Cerebellum

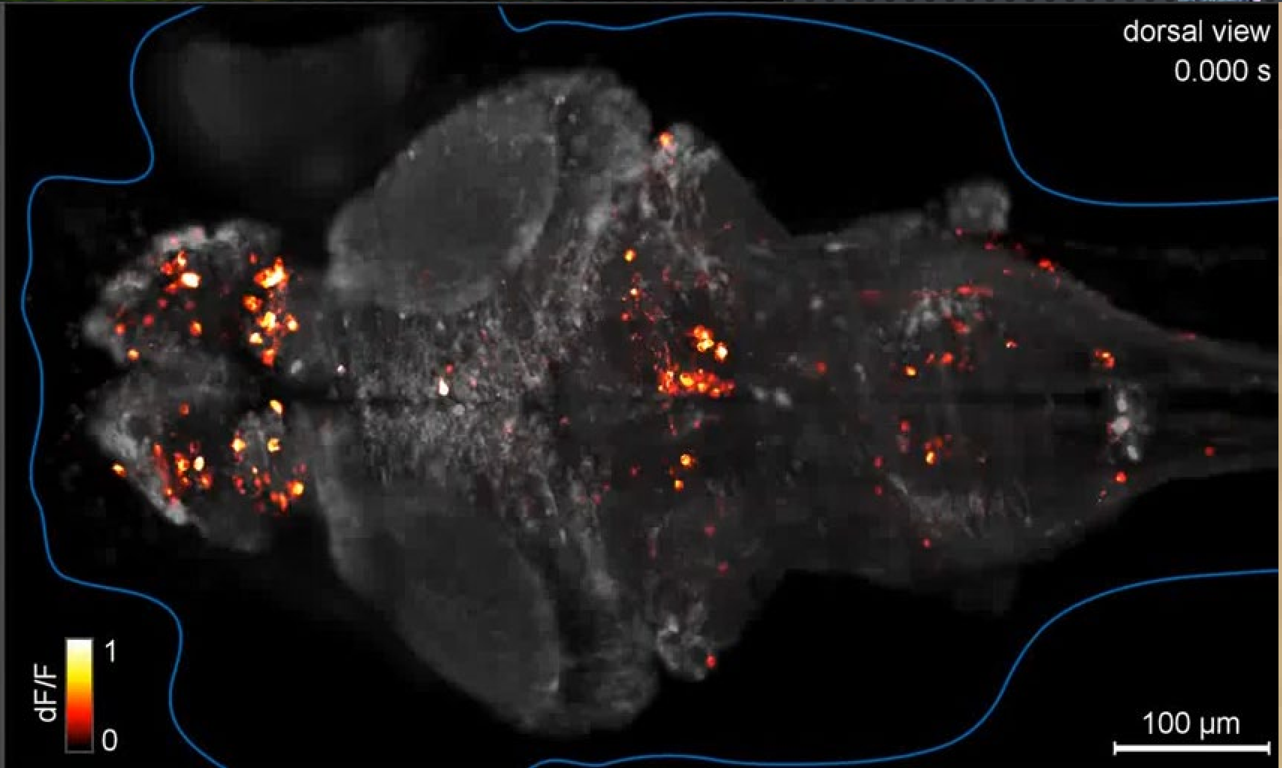
Brain anatomy and function



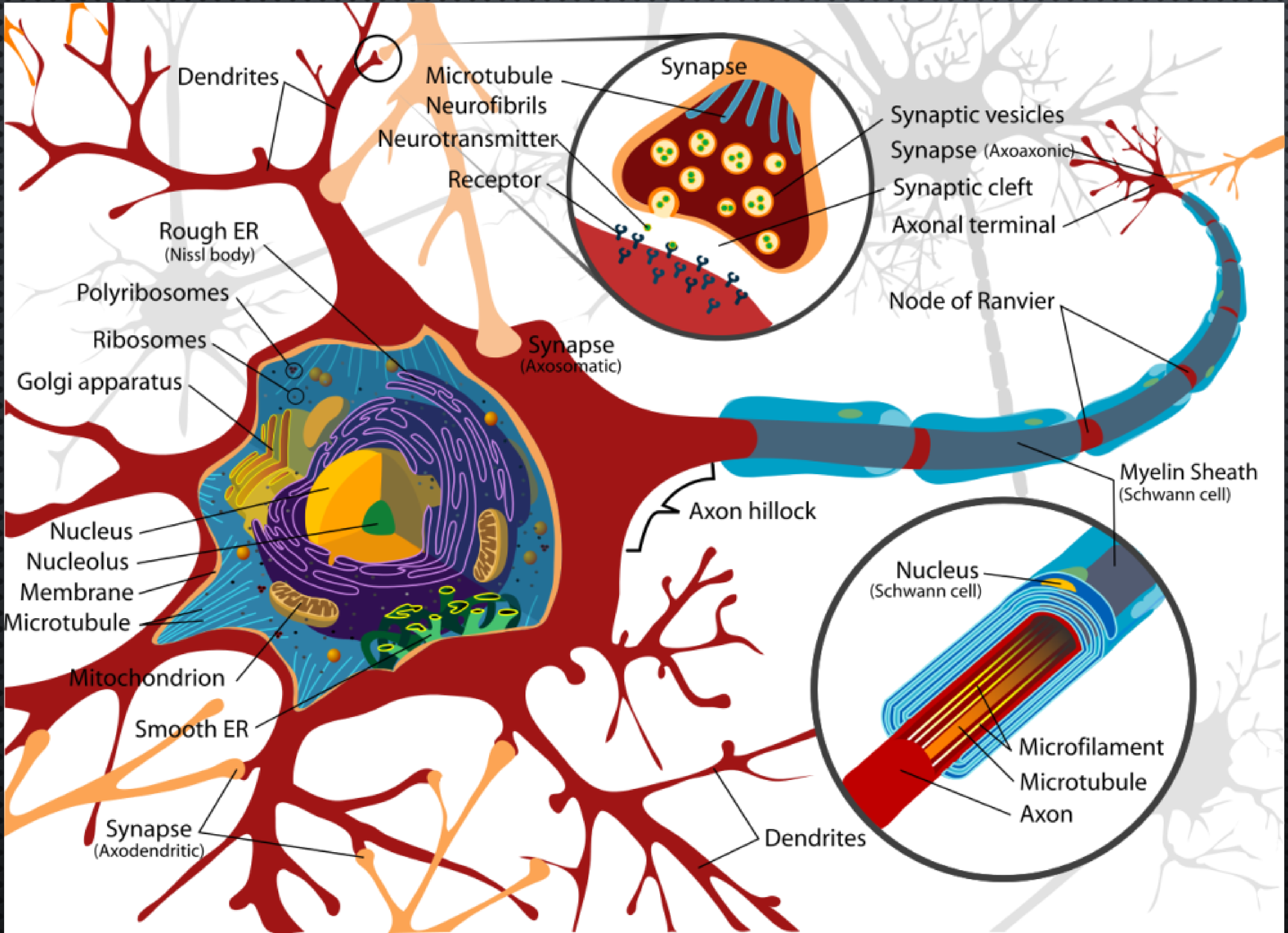
Implications for robotics

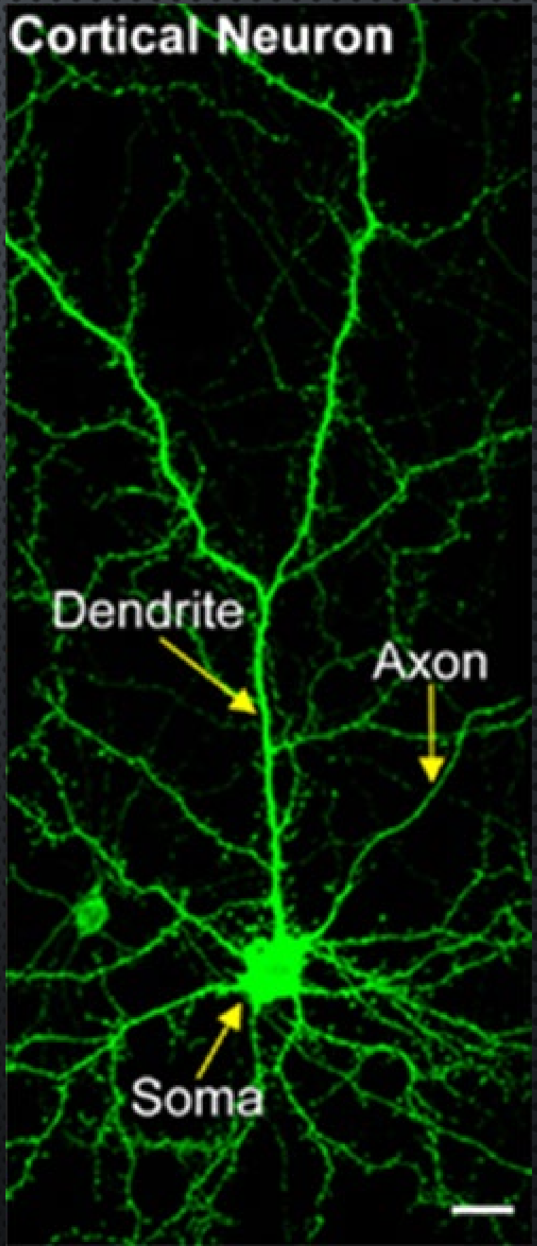


Visualising the brain

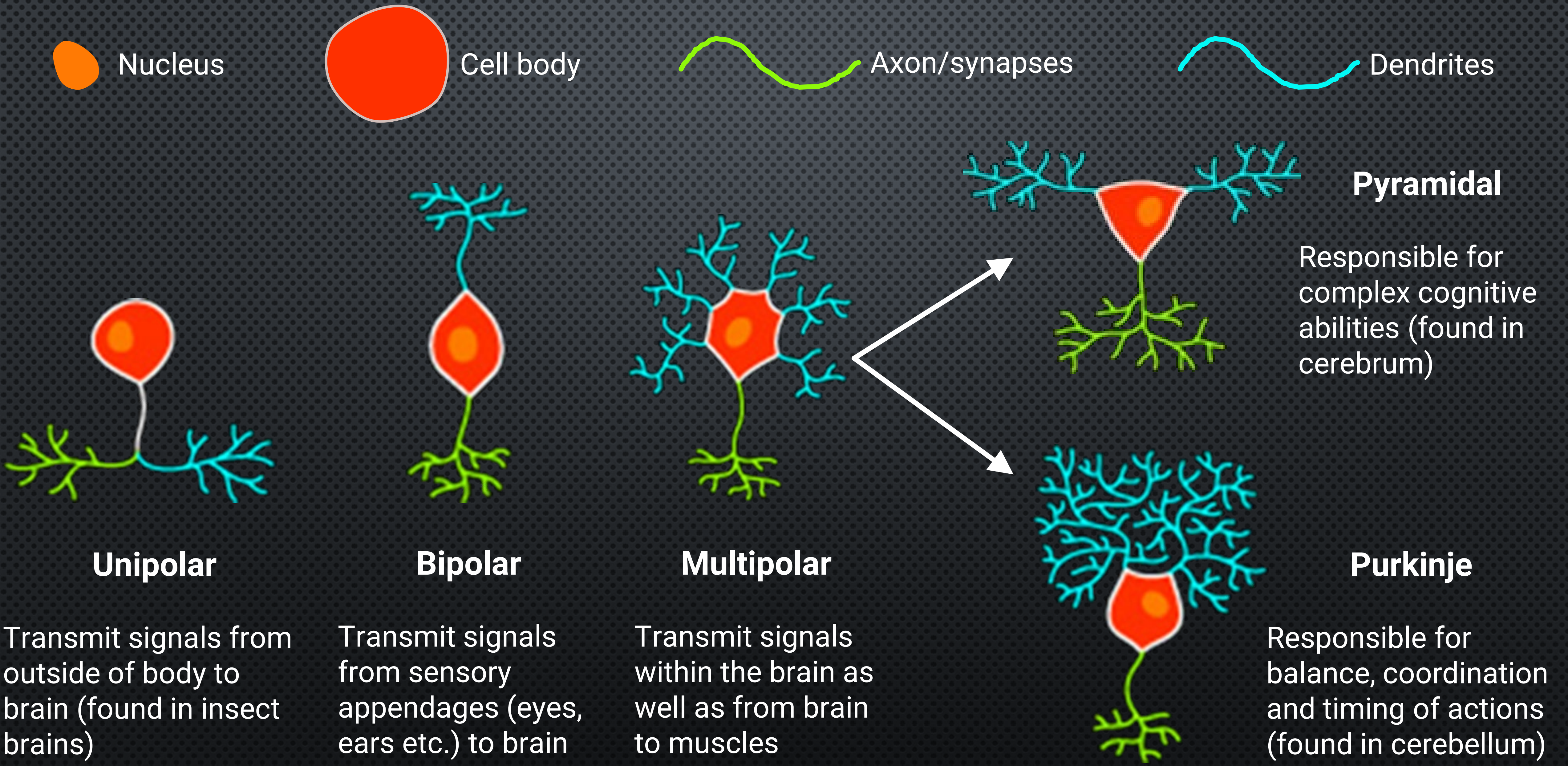


What is a neuron?

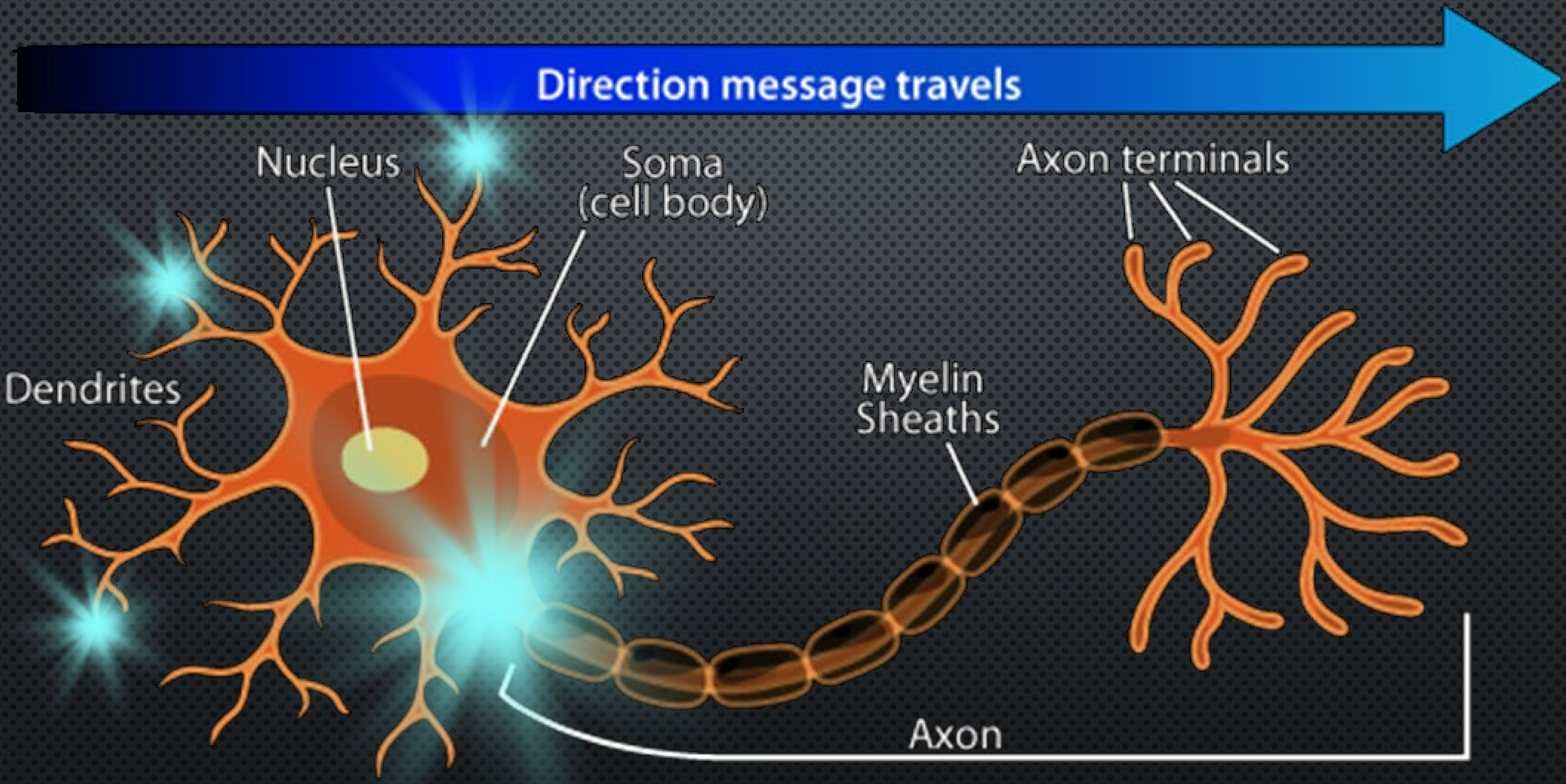




Basic types of neurons

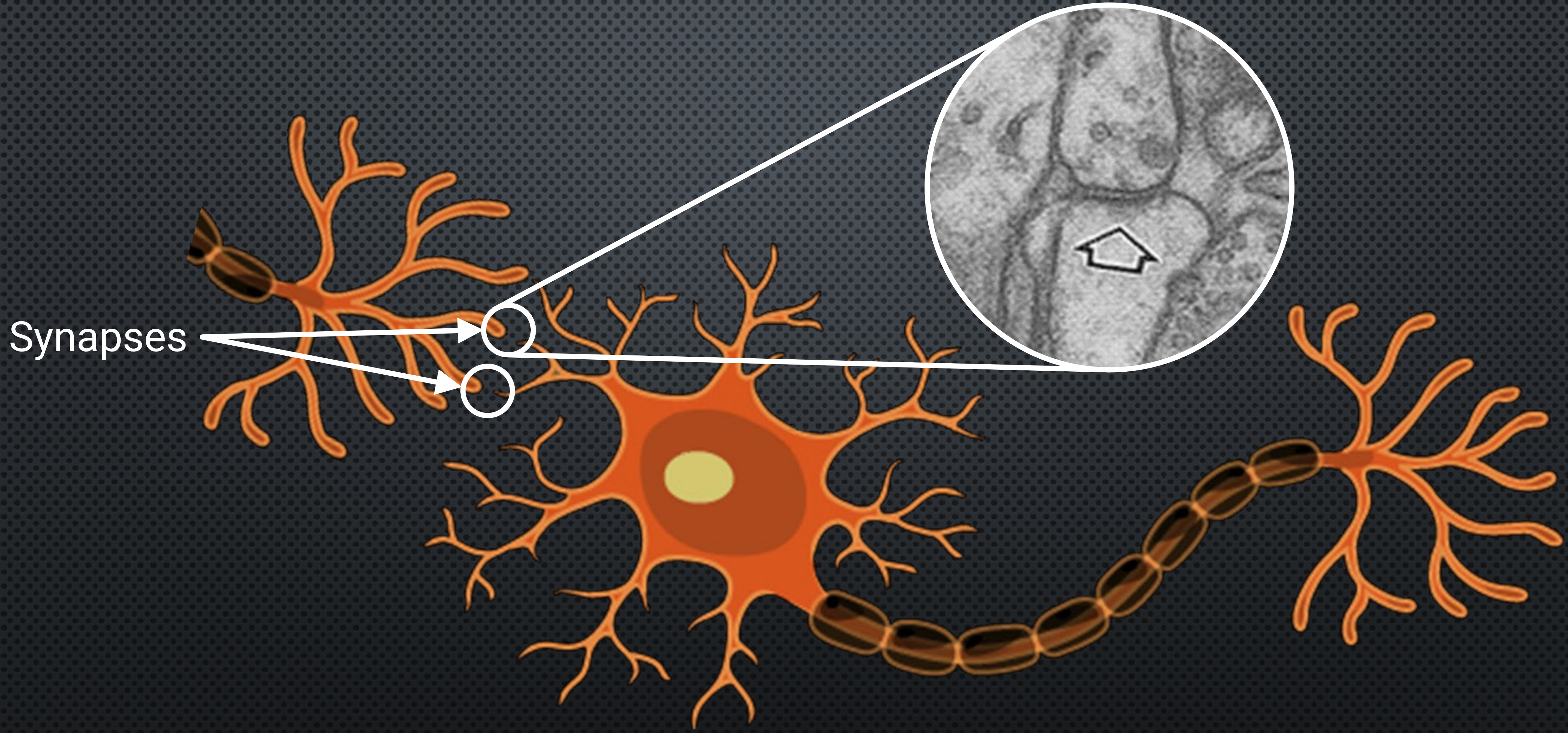


Neurons integrate and transmit information



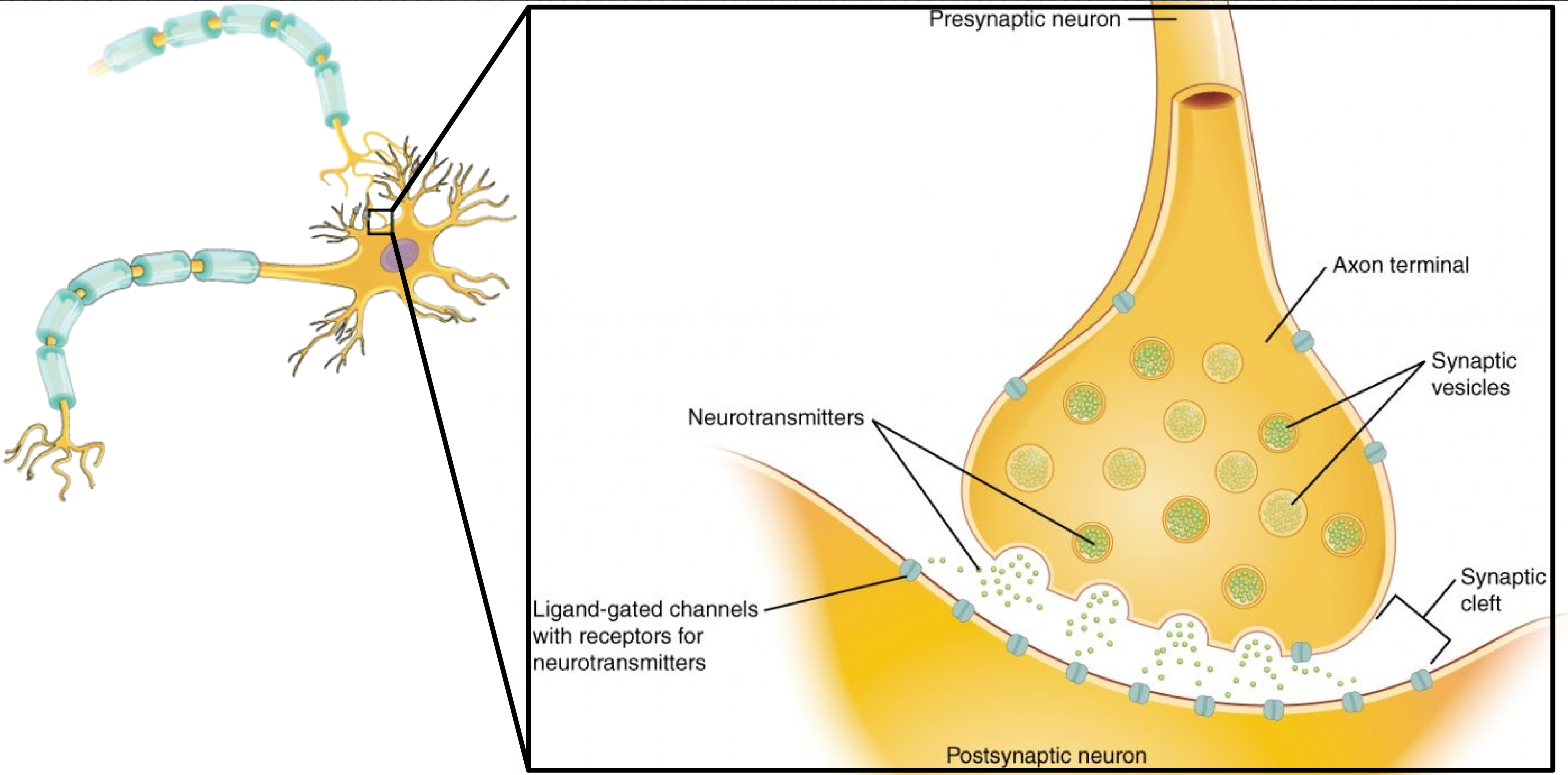
Neurons collect electrical signals (called action potentials) from other neurons via synapses and transmit them to other neurons via the axon

Synapses bridge neurons together

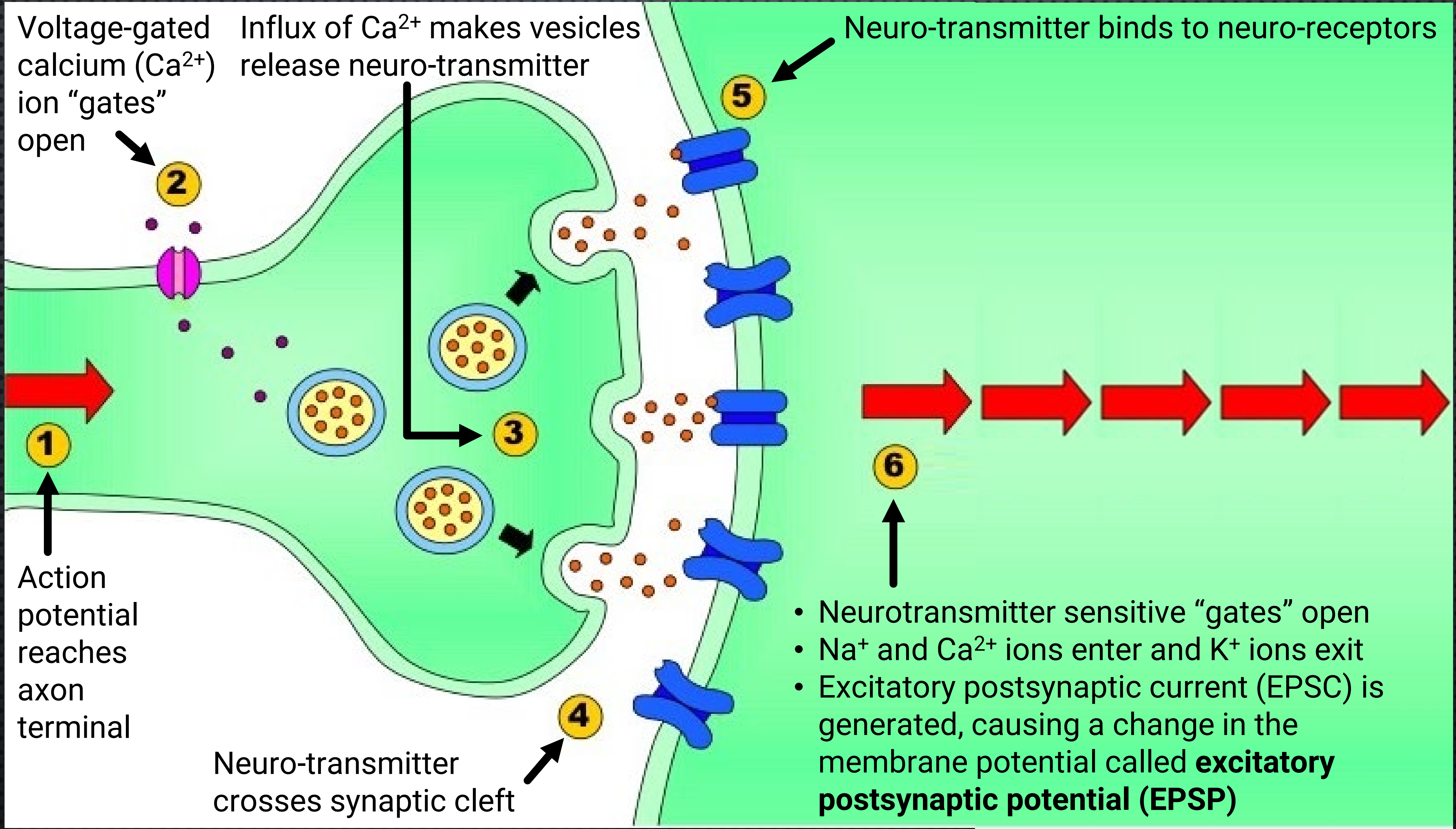


A synapse is a structure that permits a neuron to pass an electrical or chemical signal to another neuron.

Structure of a chemical synapse



Signal transmission in synapses



- Neurotransmitter sensitive "gates" open
- Na^{+} and Ca^{2+} ions enter and K^{+} ions exit
- Excitatory postsynaptic current (EPSC) is generated, causing a change in the membrane potential called **excitatory postsynaptic potential (EPSP)**

Action potential

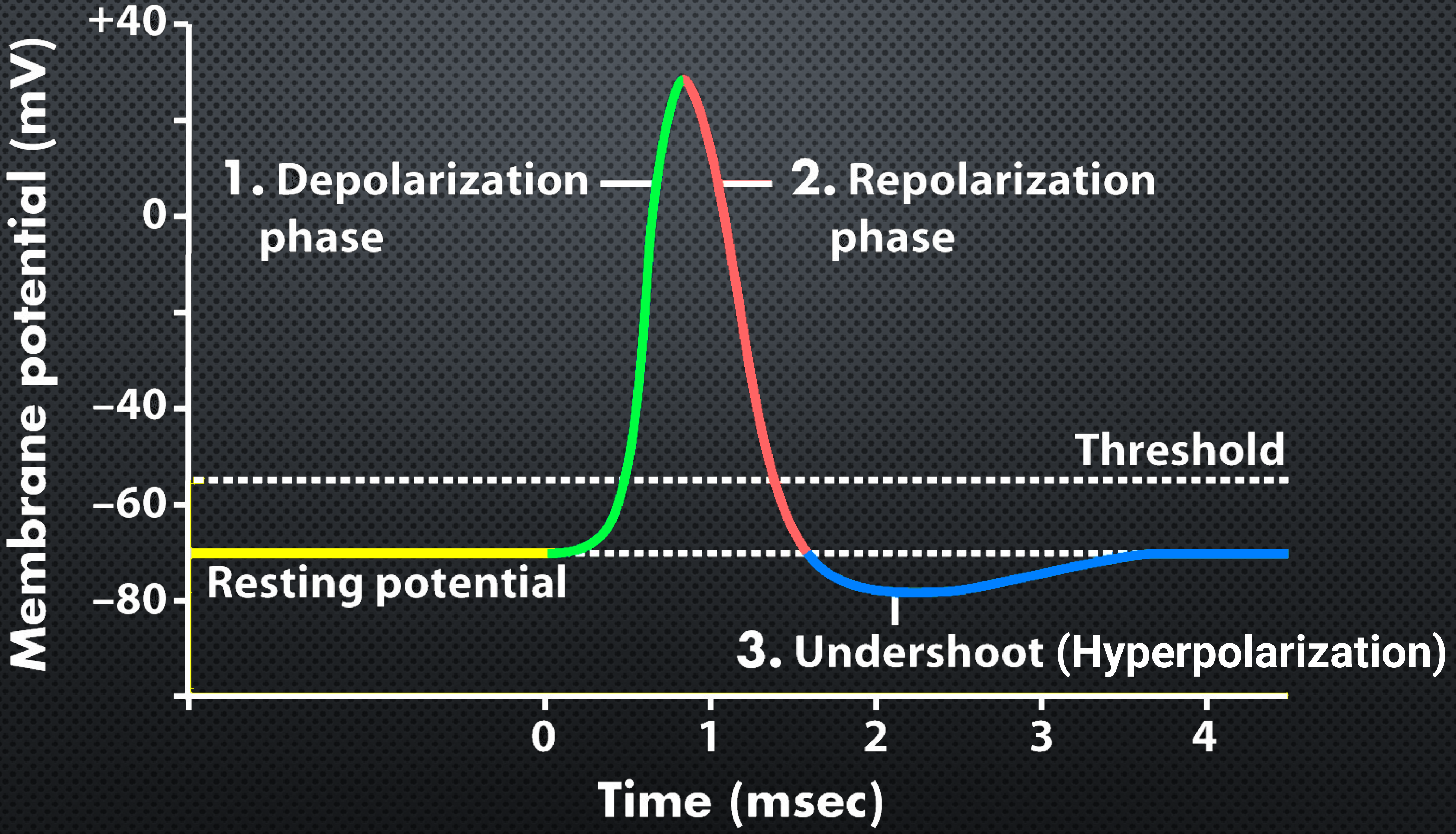
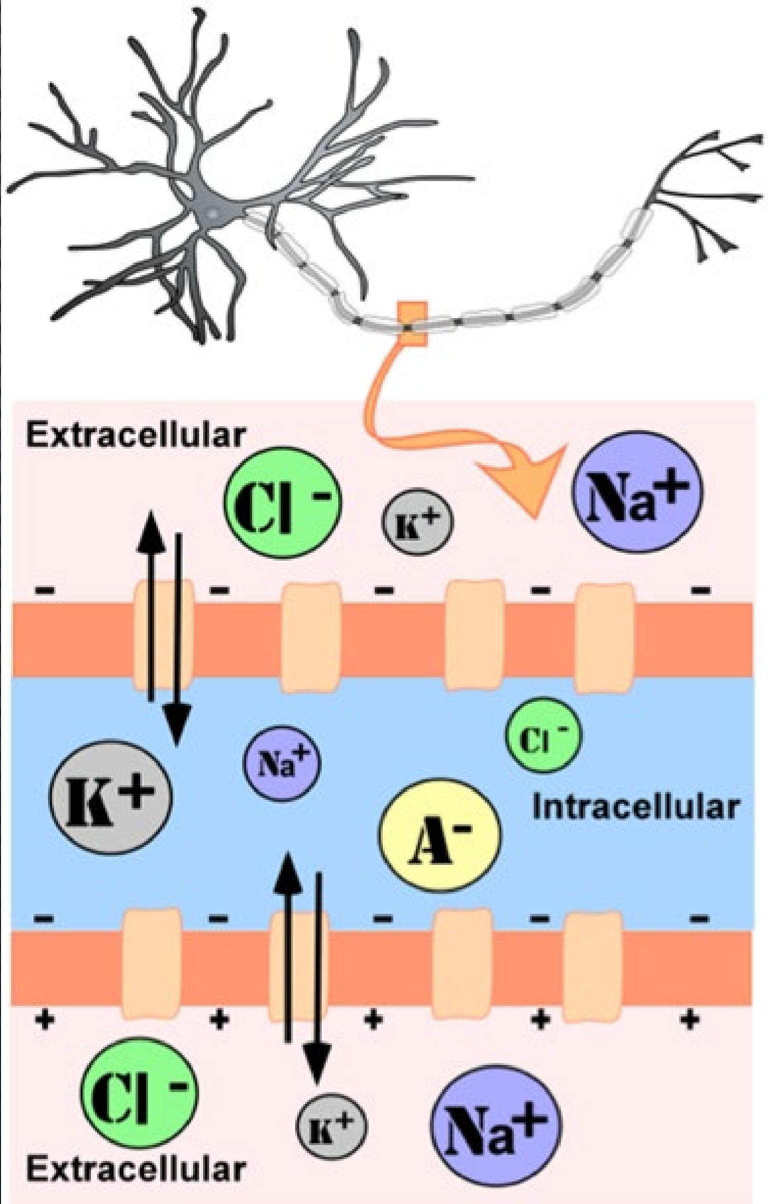
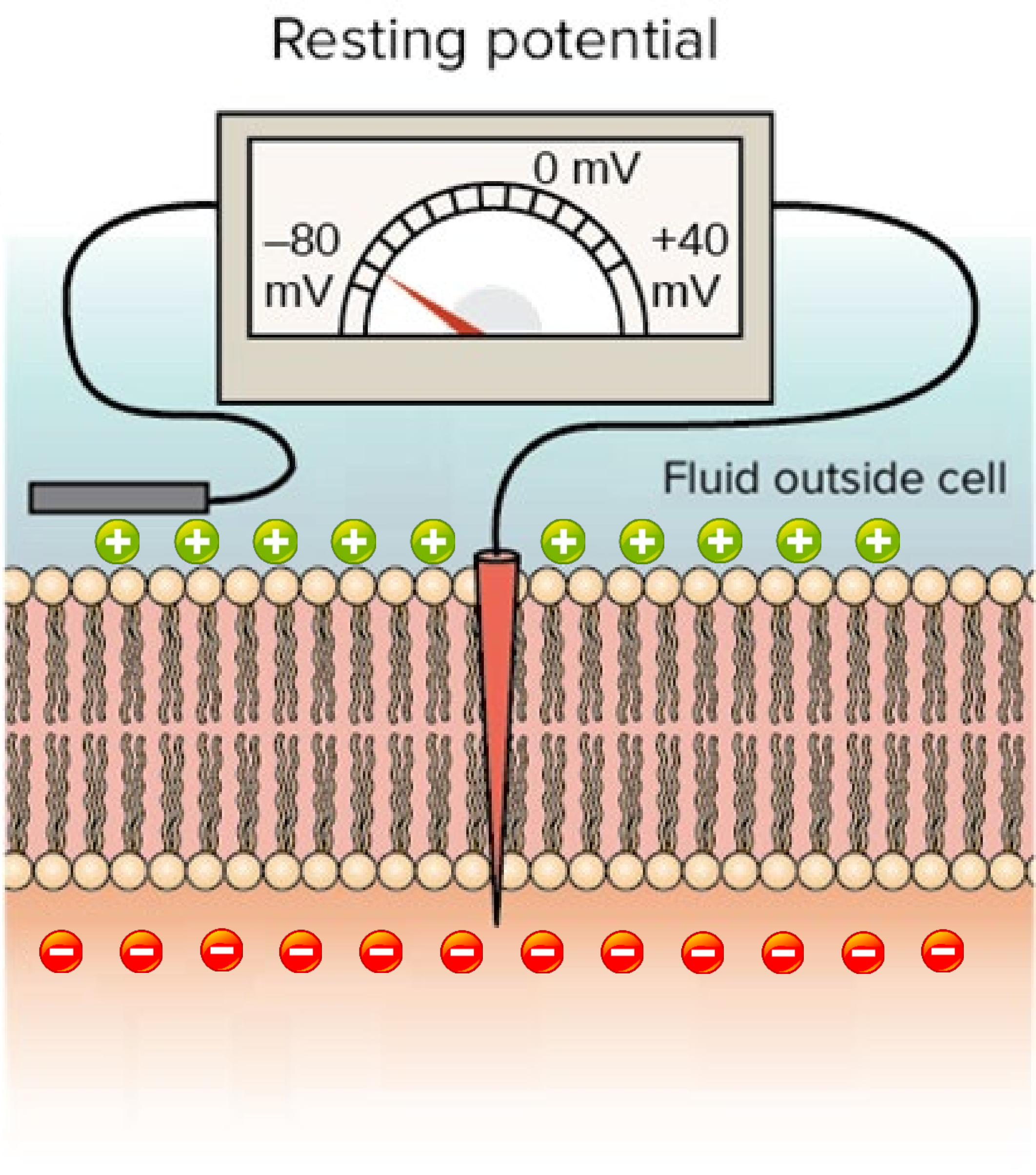


Figure 45-5 Biological Science, 2/e

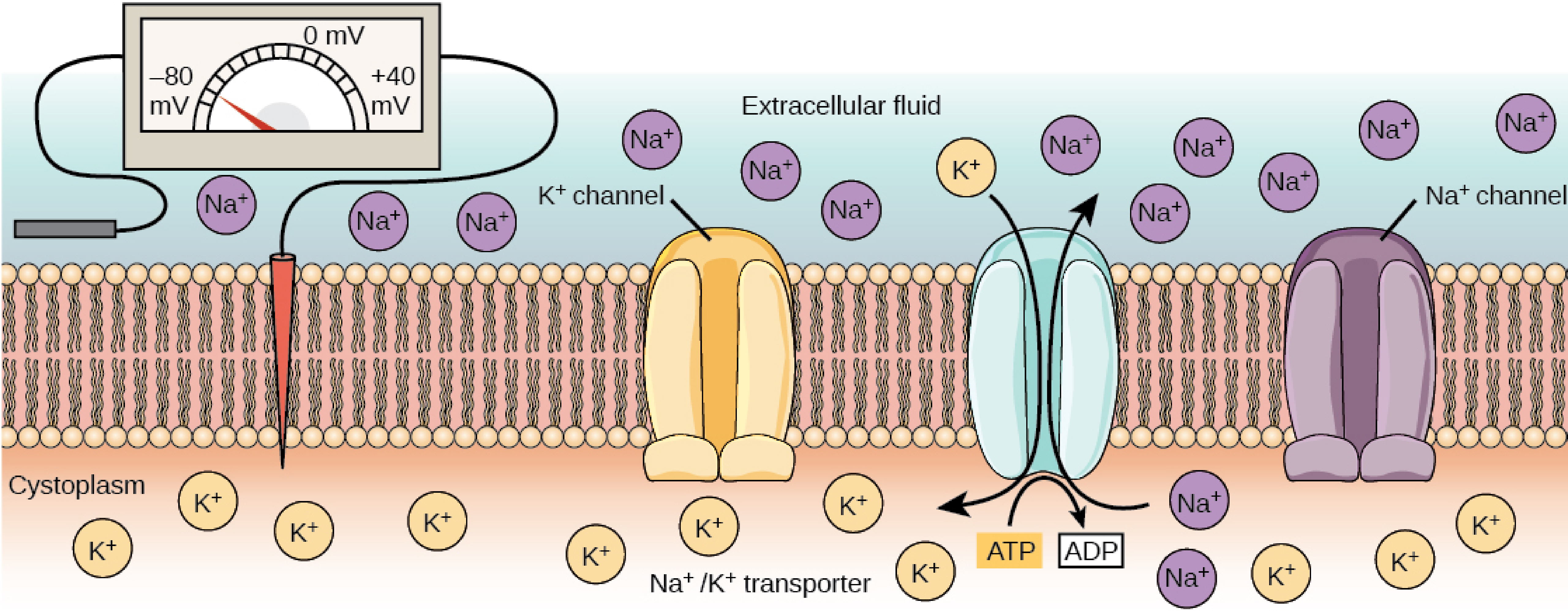
Resting membrane potential: default state of a neuron

- Difference between electric potential inside and outside a neuron, due to different ion concentration
- Potassium (K^+) ions have higher concentration inside
- Sodium (Na^+) ions have higher concentration outside
- Leakage of K^+ to outside leaves excess negative charge inside and excess positive charge outside
- Ion leakage sets resting membrane potential to around −65 mV to -70 mV



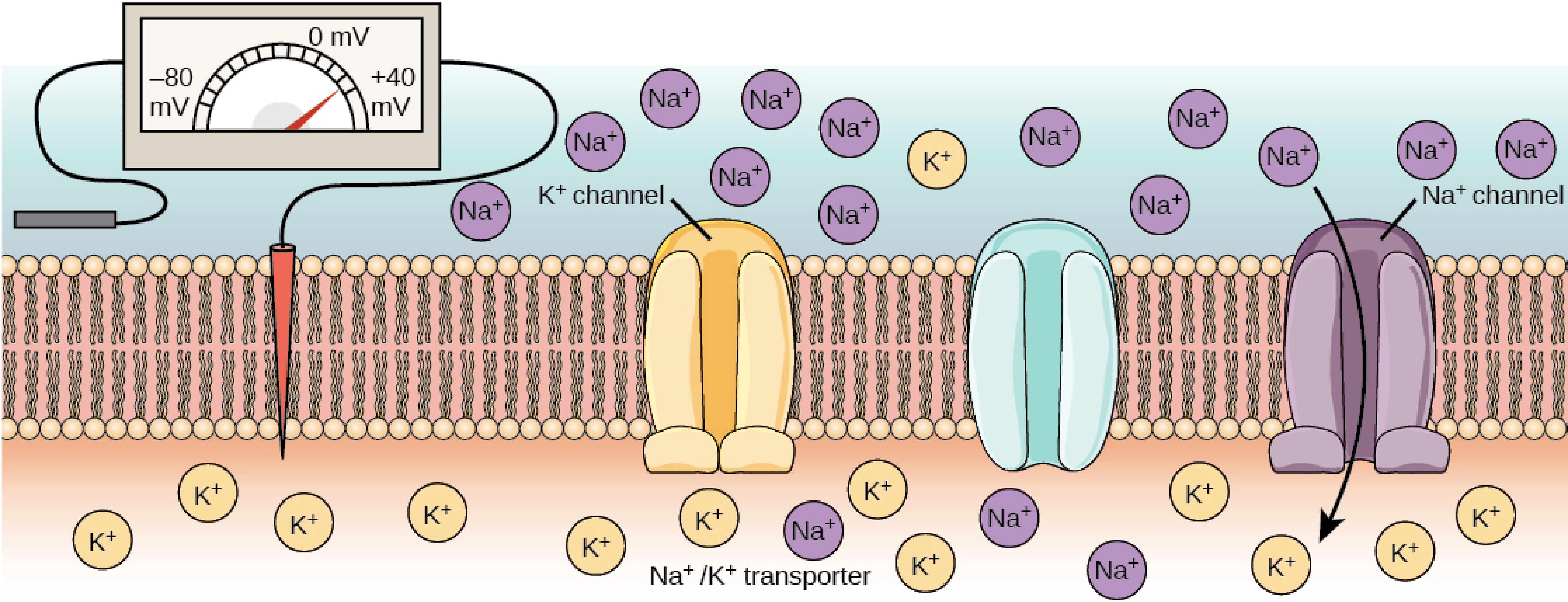


Resting membrane potential: default state of a neuron



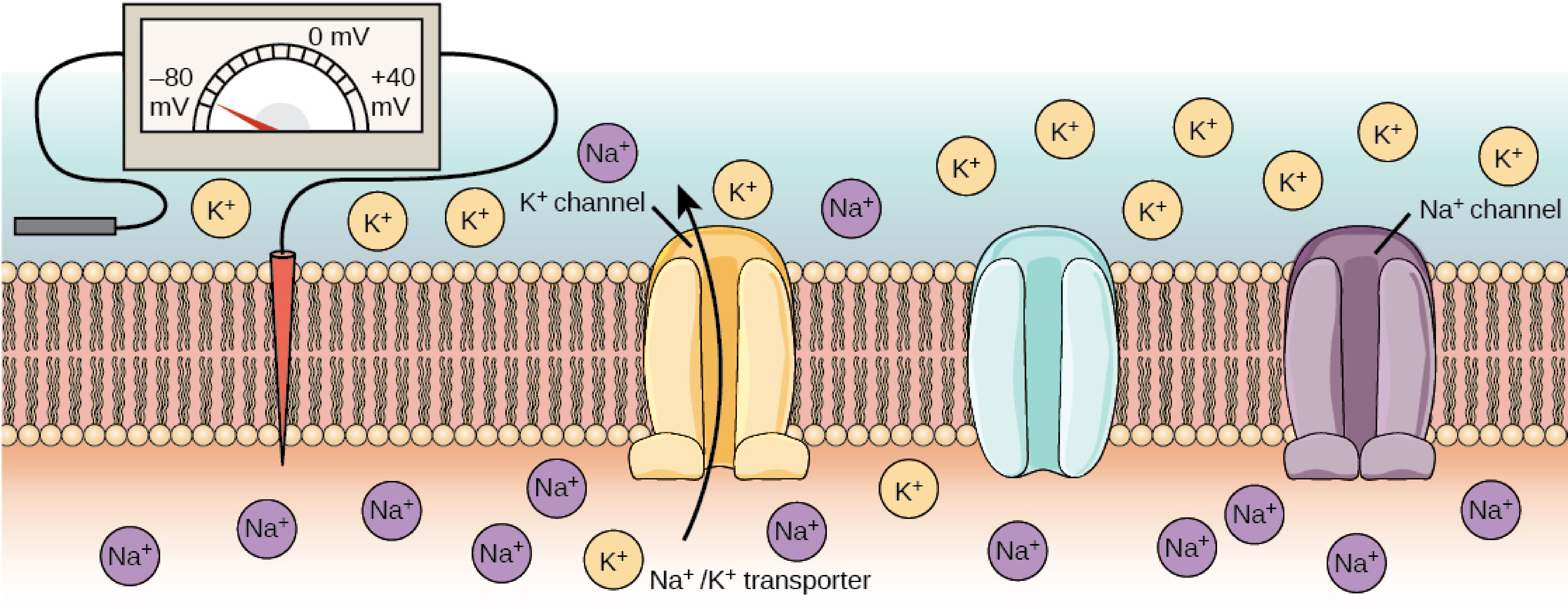
At the resting potential, all voltage-gated Na^+ channels and most voltage-gated K^+ channels are closed. The Na^+/K^+ transporter pumps K^+ ions into the cell and Na^+ ions out.

De-polarisation



In response to a depolarization, some Na^+ channels open, allowing Na^+ ions to enter the cell. The membrane starts to depolarize (the charge across the membrane lessens). If the threshold of excitation is reached, all the Na^+ channels open.

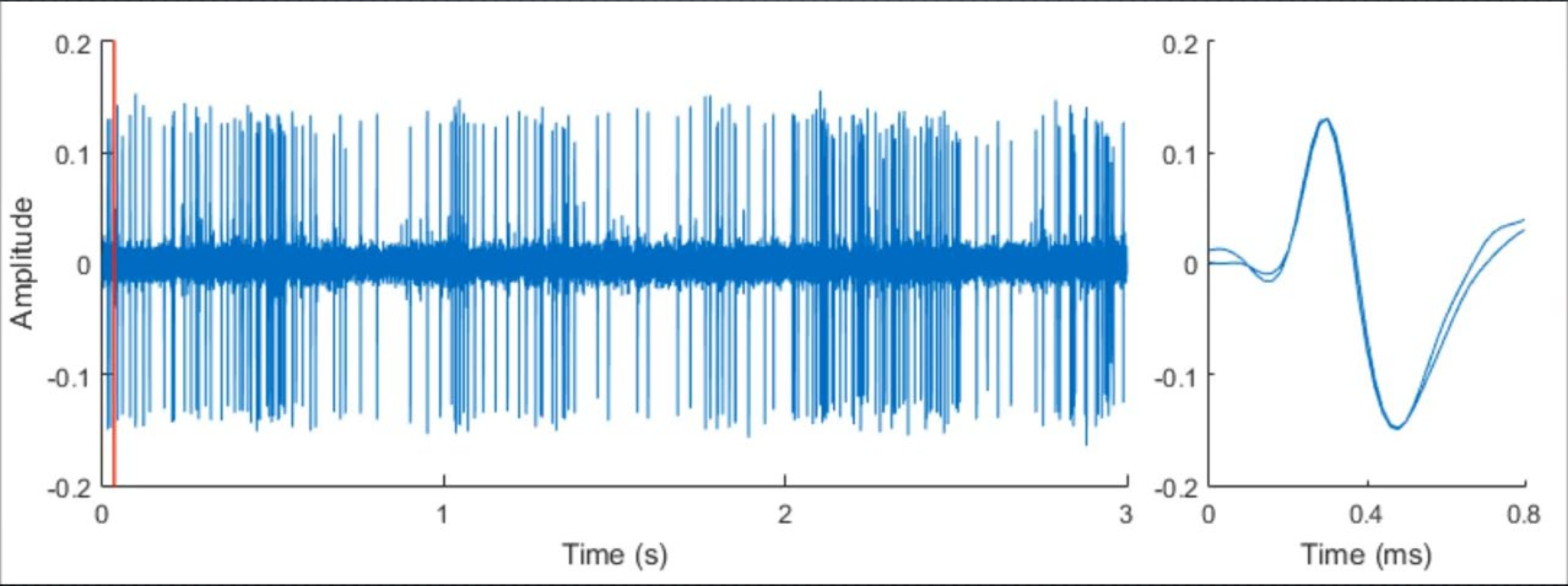
Hyper-polarisation



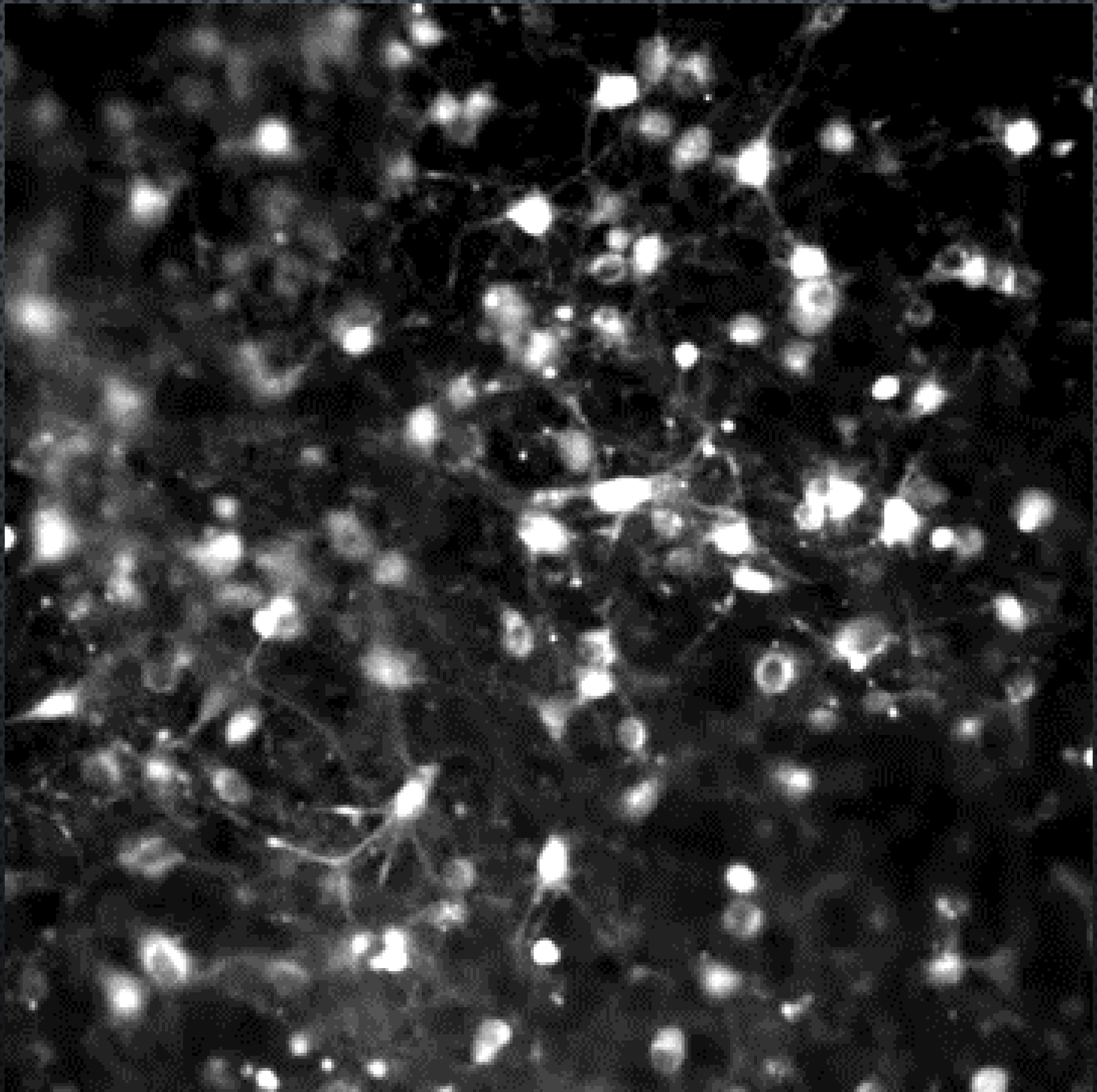
At the peak action potential, Na^+ channels close while K^+ channels open. K^+ leaves the cell, and the membrane eventually becomes hyperpolarized.

How is action potential actually measured?

Sights and sounds of neuronal recordings

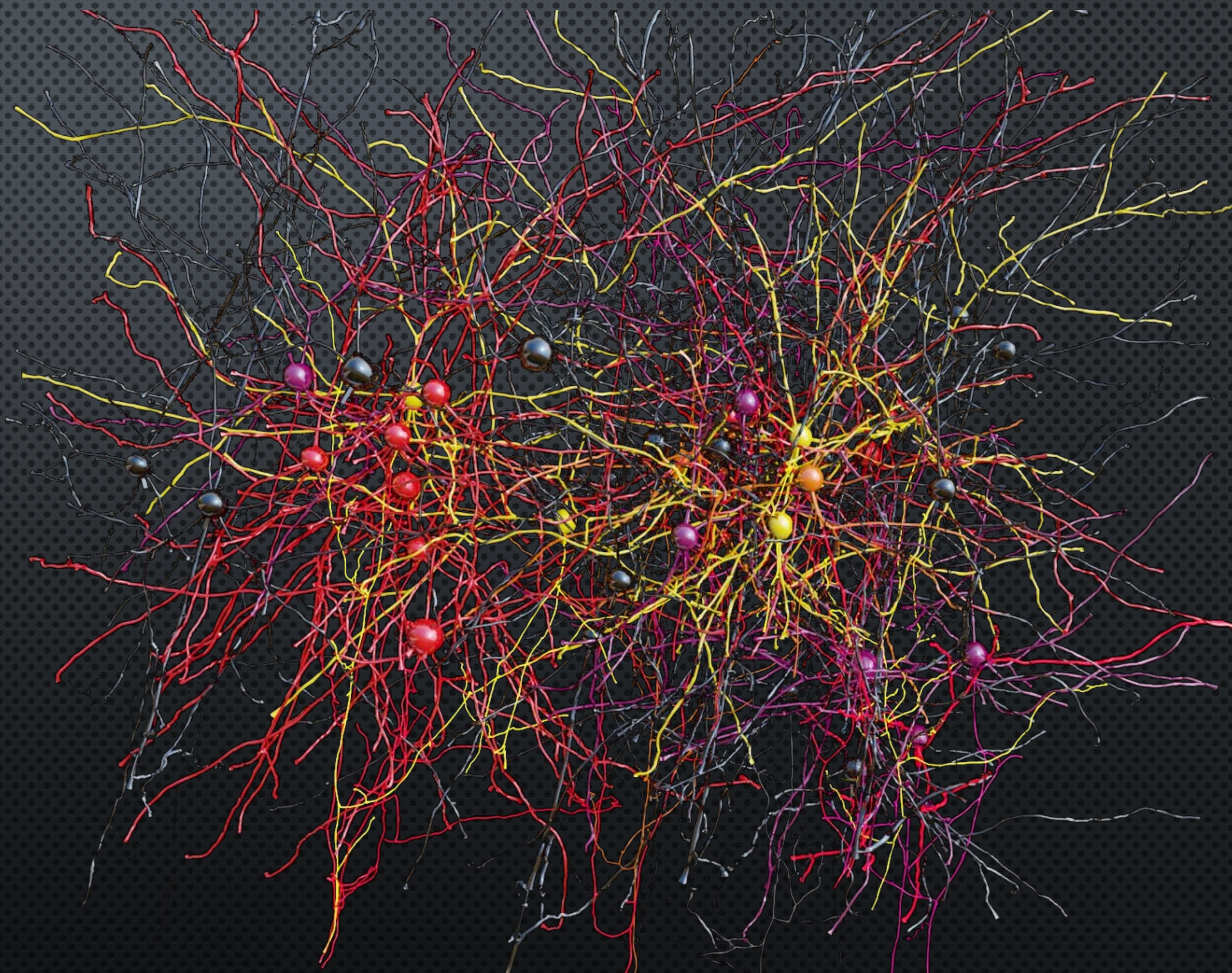


Neuronal firing

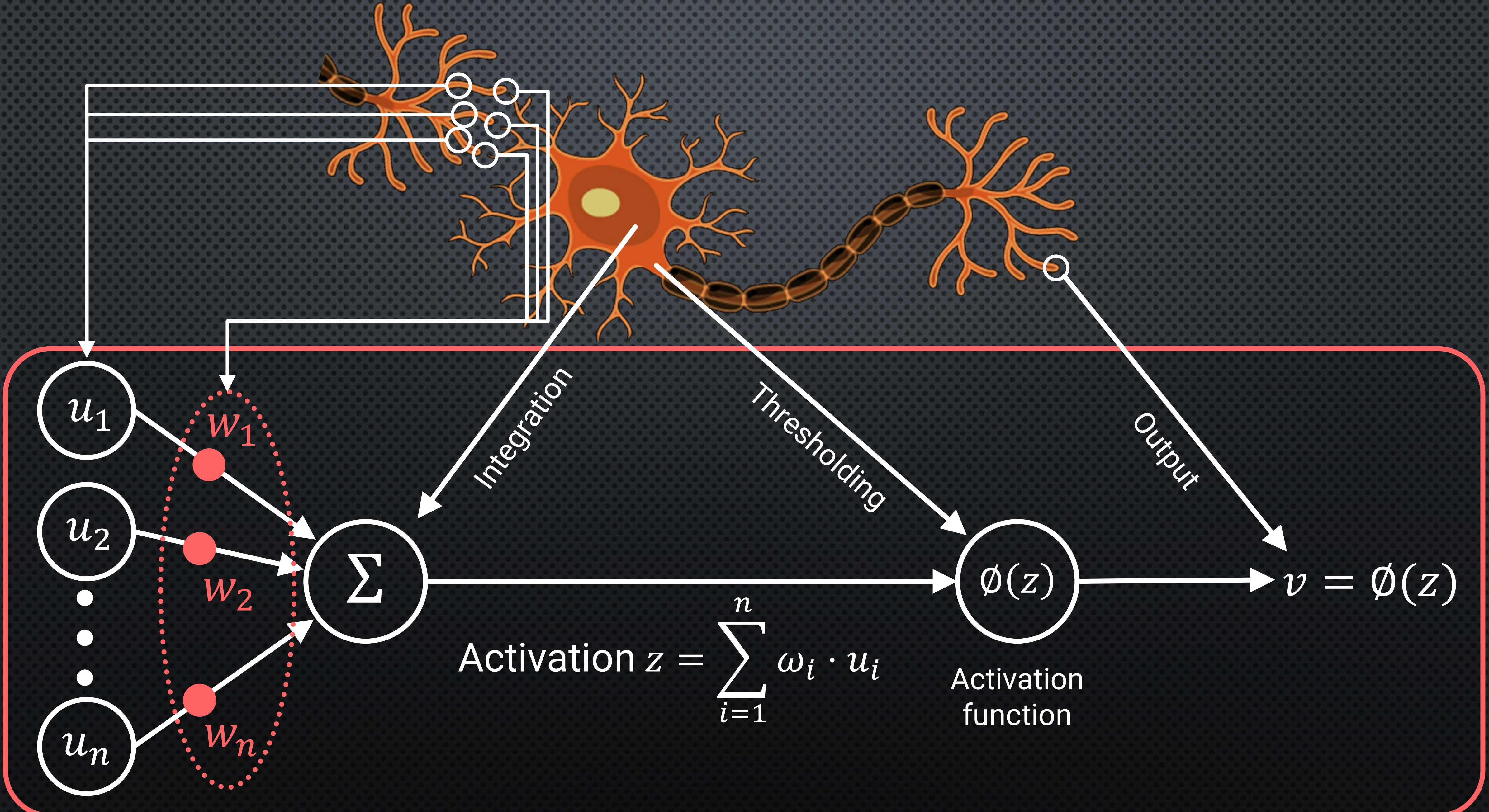


Neural connectivity is very complex!

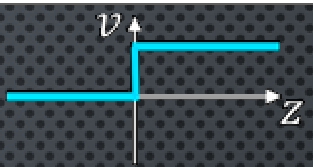
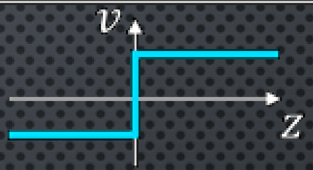
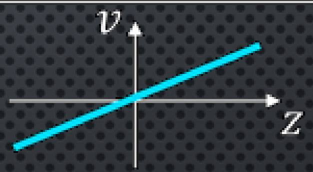
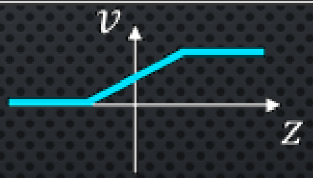
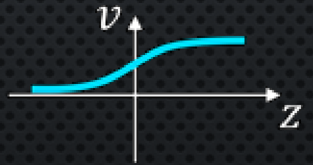
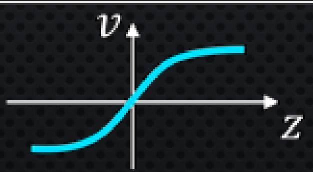
- A typical neuron may receive inputs from on the order of 10,000 neurons
- Pyramidal cells in the hippocampus can receive on the order of 50,000 inputs
- There are an estimated 1000 trillion synapses in the human brain
- Although neural connectivity is highly dense, the brain is not a fully connected network



Perceptrons: from real to artificial neurons

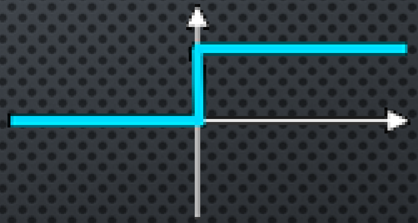


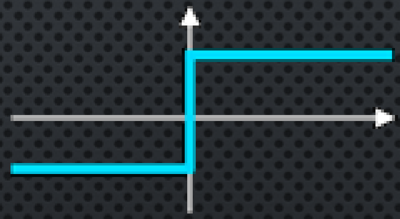
Common activation functions

Activation function	Equation	Example	1D Graph
Unit step (Heaviside) Threshold Logic Unit (TLU)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer NN	

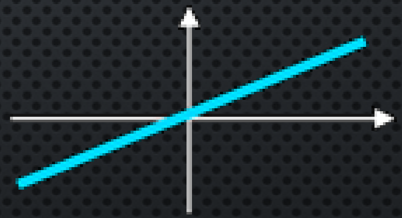
Which activation function to choose?

- Activation function decides what output the perceptron produces
- Choose an AF based on what you want your output to look like. For example, if your output:
  - can only be 0 or 1 (-1 or 1), choose TLU or sign
    - Good for binary classification problems but bad for everything else

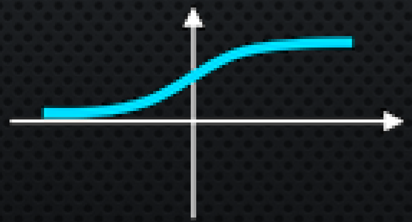


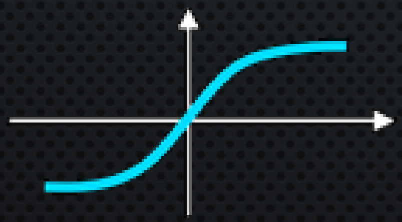


  - can be any value between $-\infty$ to $+\infty$, choose linear
    - Almost always a bad choice so try to avoid using it

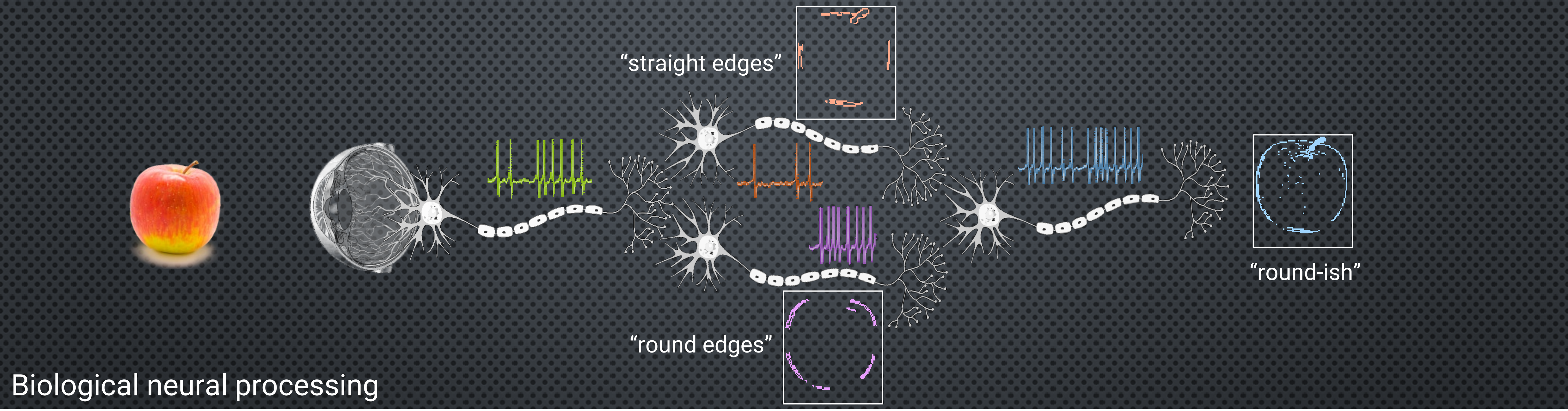


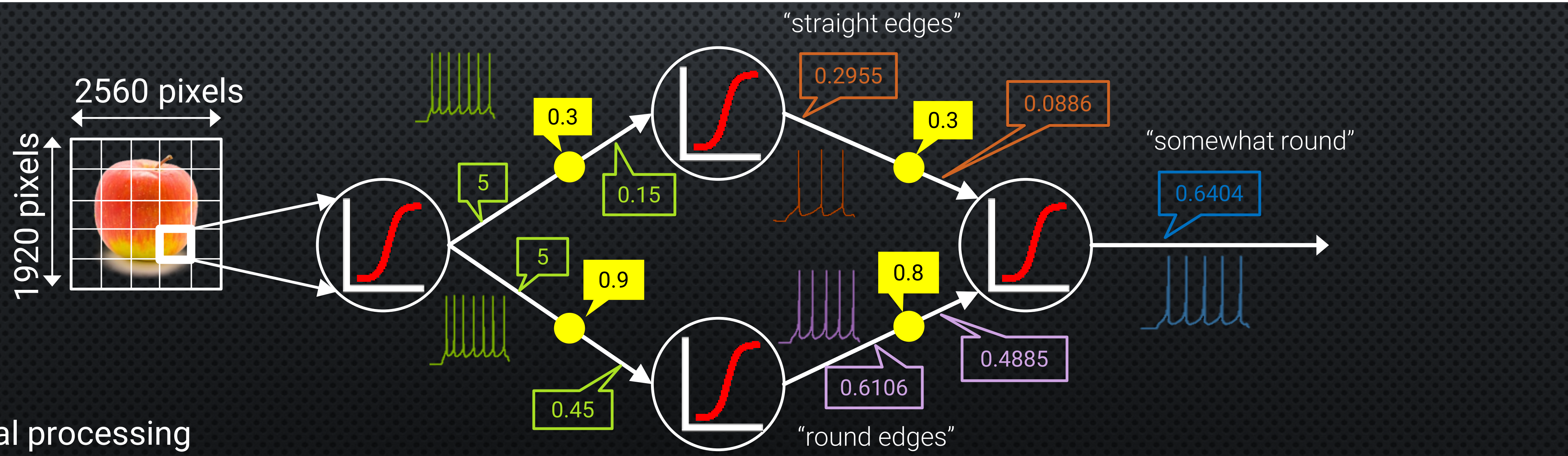
  - can be any value between 0 or 1 (-1 or 1), choose logistic or hyperbolic tangent
    - Almost always the best choice for most ML problems



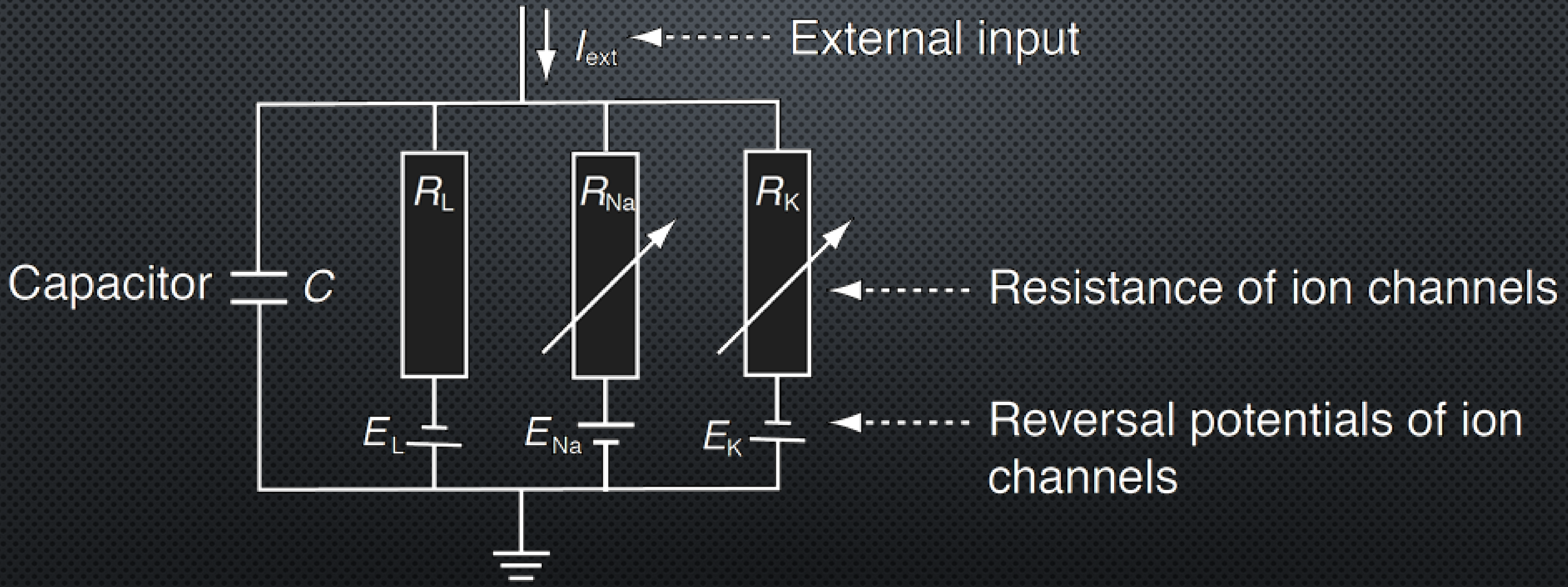


Biologically-inspired computation



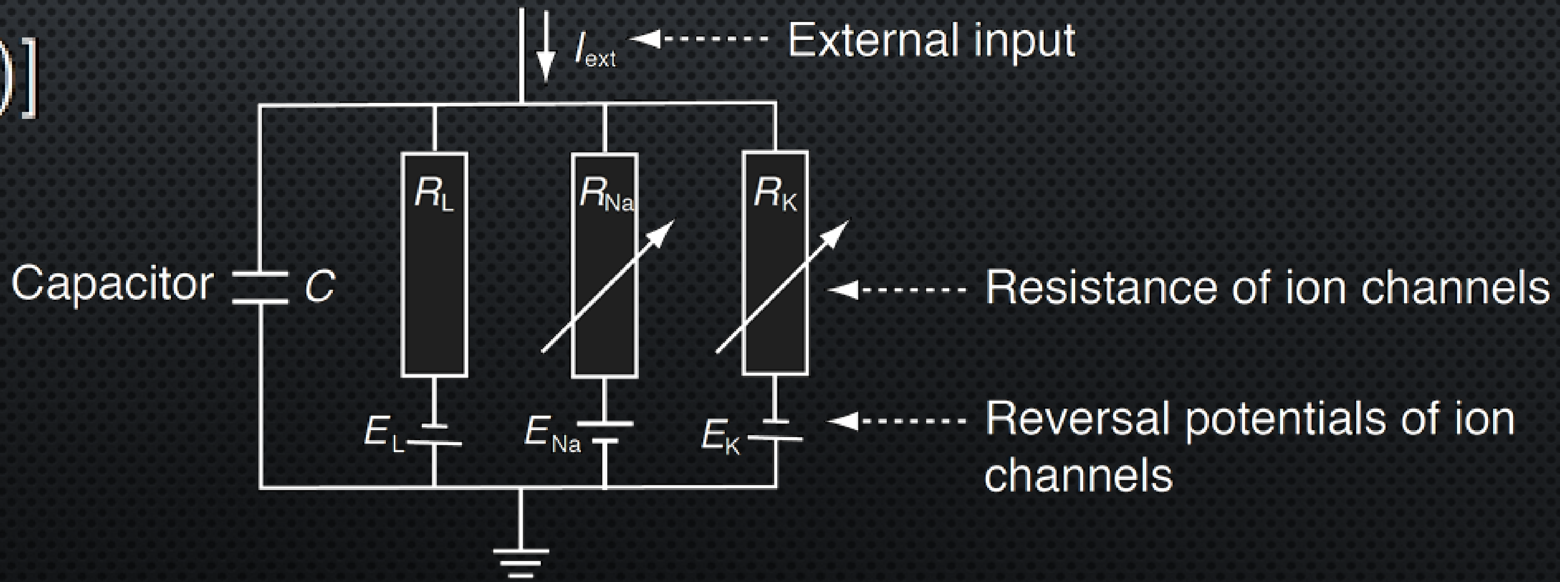


A biologically realistic neuron model: Hodgkin-Huxley model

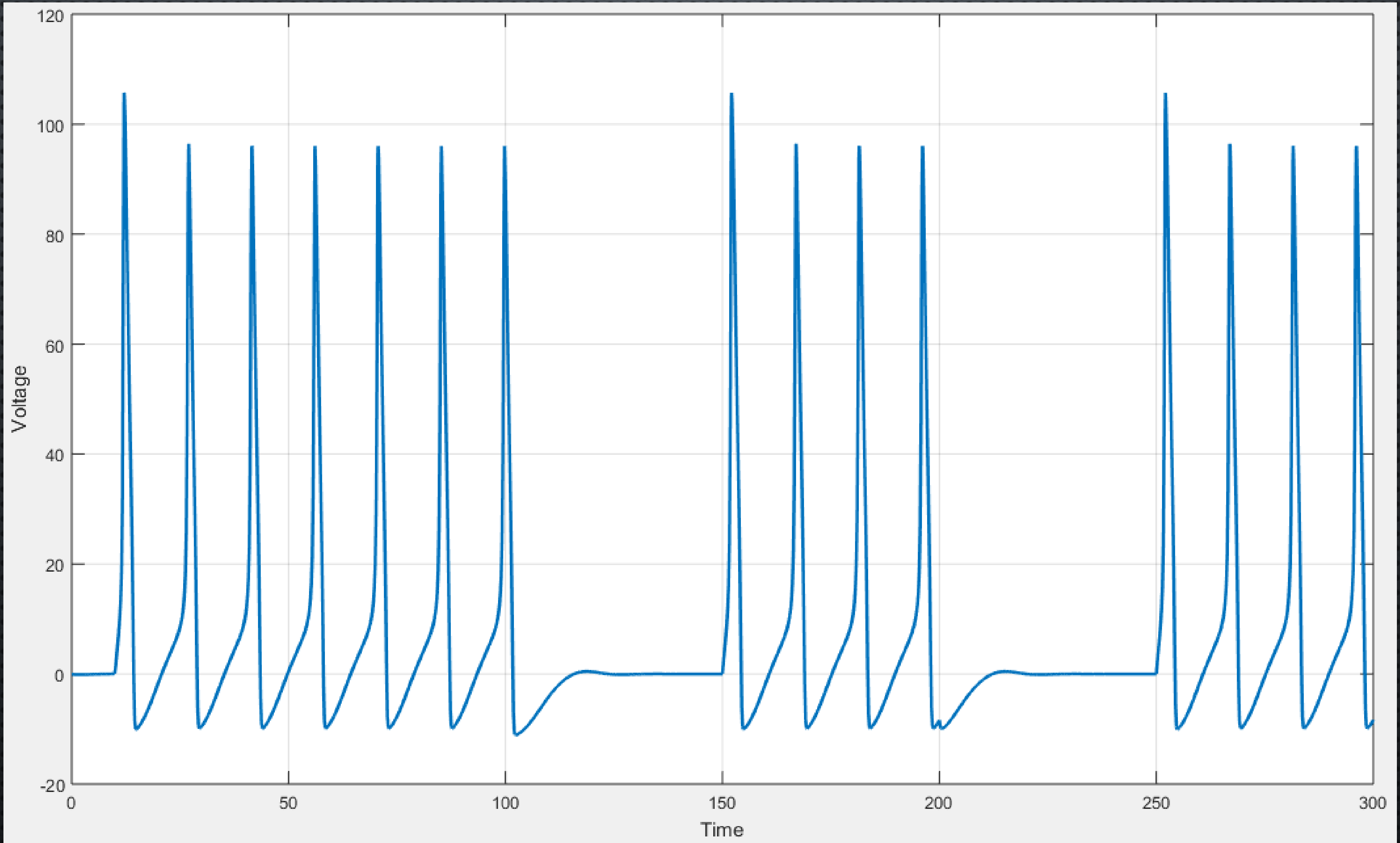


A biologically realistic neuron model: Hodgkin-Huxley model

$$
\begin{aligned}
C\frac{\mathrm{d}V}{\mathrm{d}t} &= -g_{\mathrm{K}}n^4(V-E_{\mathrm{K}}) - g_{\mathrm{Na}}m^3h(V-E_{\mathrm{Na}}) - g_{\mathrm{L}}(V-E_{\mathrm{L}}) + I(t) \\
\tau_{\mathrm{n}}(V)\frac{\mathrm{d}n}{\mathrm{d}t} &= -[n-n_0(V)] \\
\tau_{\mathrm{m}}(V)\frac{\mathrm{d}m}{\mathrm{d}t} &= -[m-m_0(V)] \\
\tau_{\mathrm{h}}(V)\frac{\mathrm{d}h}{\mathrm{d}t} &= -[h-h_0(V)]
\end{aligned}
$$



A biologically realistic neuron model: Hodgkin-Huxley model

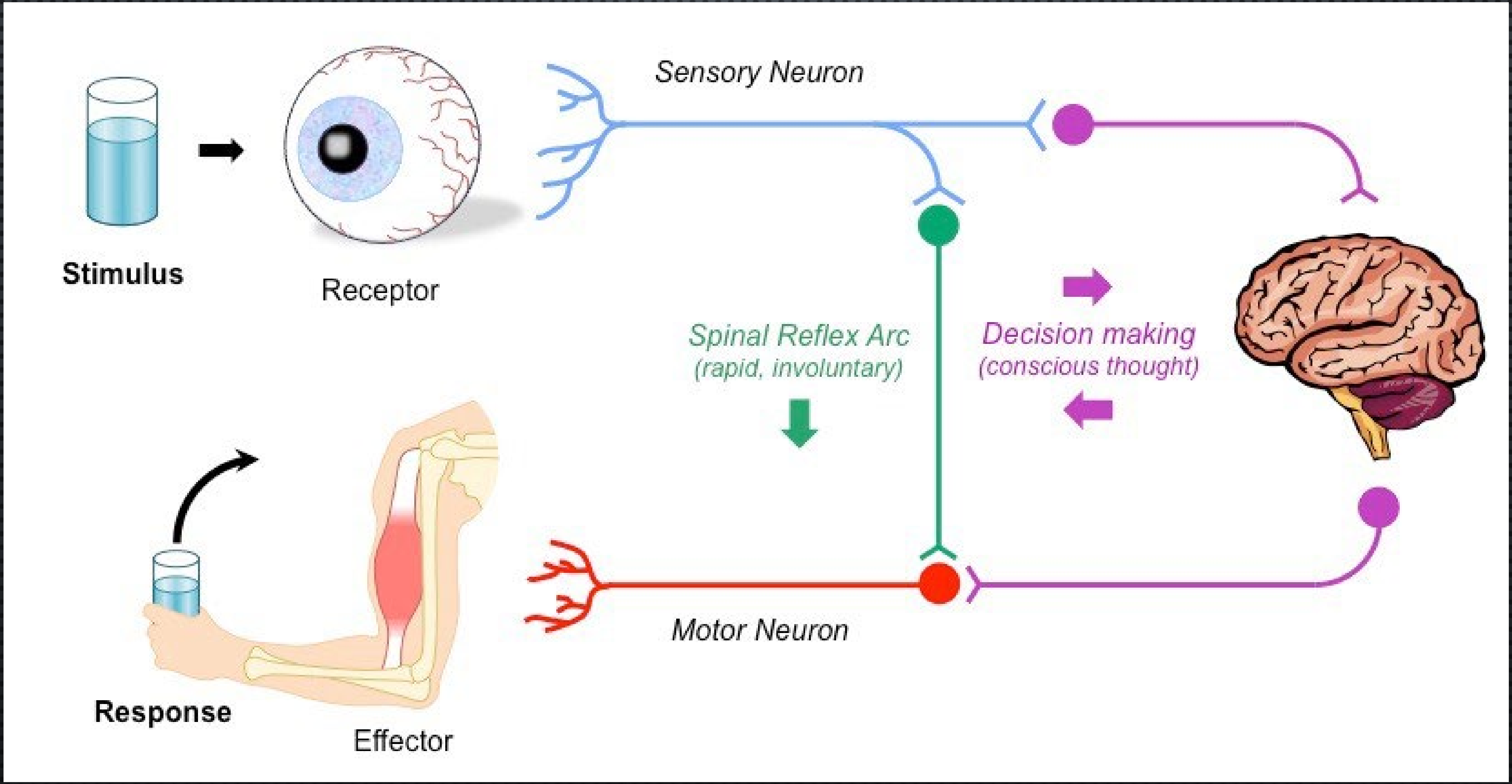


Matlab exercises

- Download “Assignment 1.pdf” from Itslearning and follow the instructions in it.

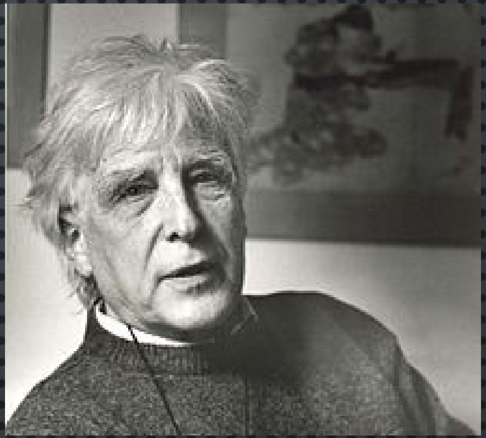
3 Simple artificial neural brains for sensing and control

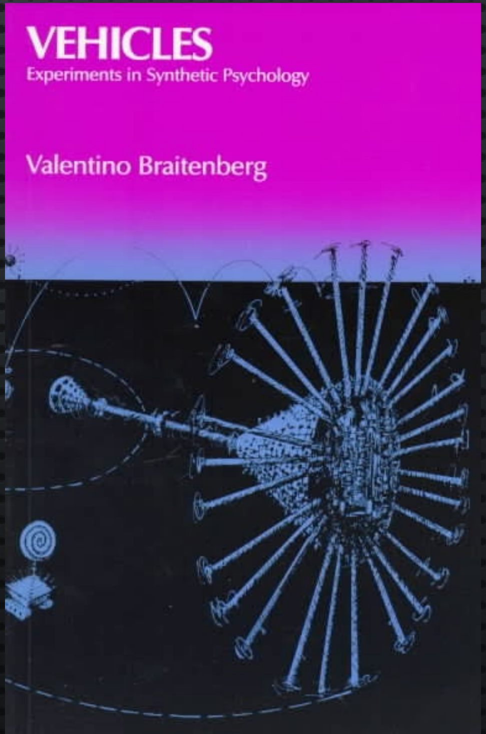
A simple template for an artificial brain



Braitenberg vehicles: a simple neurorobot

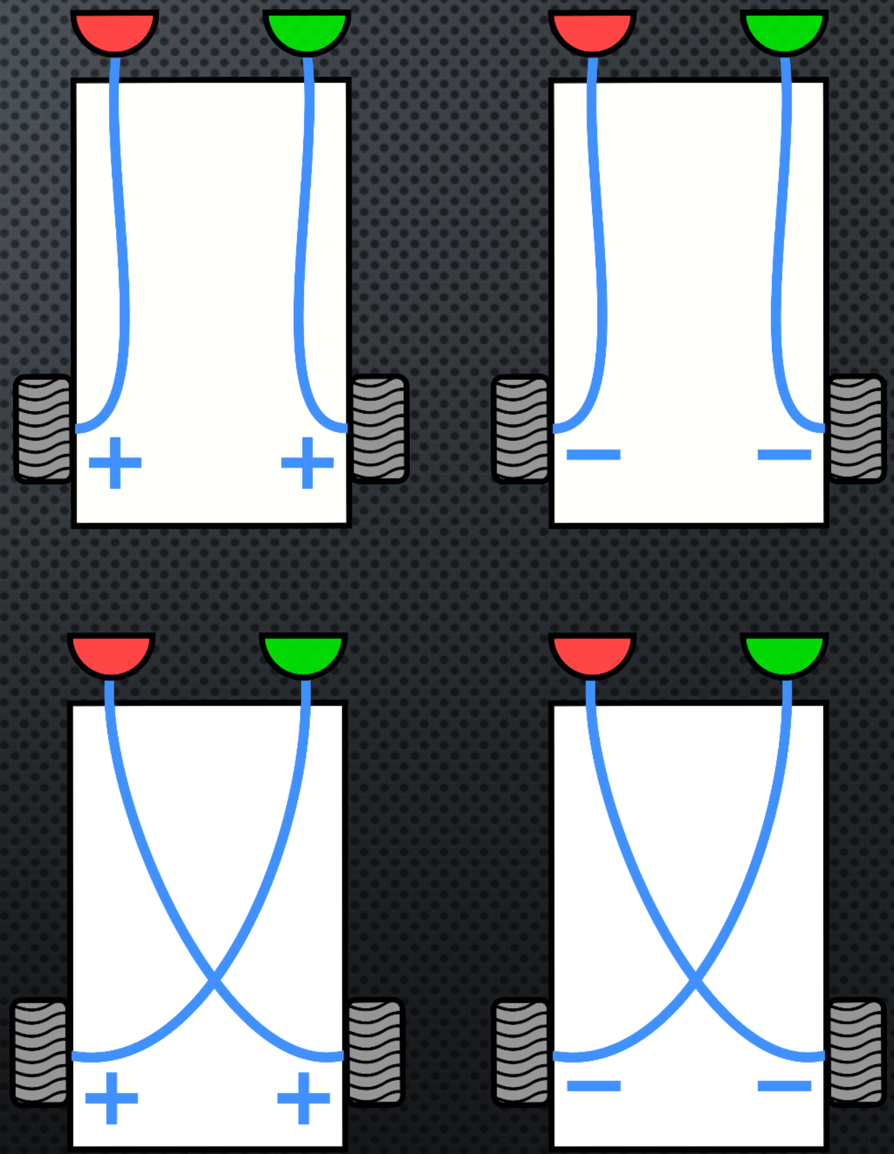
- Proposed by Valentino Braitenberg in 1984 book “Vehicles: Experiments in synthetic psychology”
- Exploring relation between structures and functions of the brain
- Hypothetical analogue vehicles (a combination of sensors, actuators and their interconnections)
- Vehicles displayed behaviours akin to aggression, love, fear, and exploration





Braitenberg vehicles: a simple neurorobot

- A sensor is directly connected to an actuator (e.g. light sensor → wheel motor)
- Sensorimotor connections can be ipsilateral or contralateral and excitatory or inhibitory
- Depending on how sensors and wheels are connected, the vehicle exhibits different movement behaviours
- In a complex environment with several sources of stimulus, vehicles exhibit complex and dynamic behaviour
- Functioning of the vehicle is purely mechanical, without any information processing or other apparently cognitive processes.



Some basic Braitenberg vehicles

$$v_{left} = k \times \frac{1}{s_{right}}$$
$$v_{right} = k \times \frac{1}{s_{left}}$$

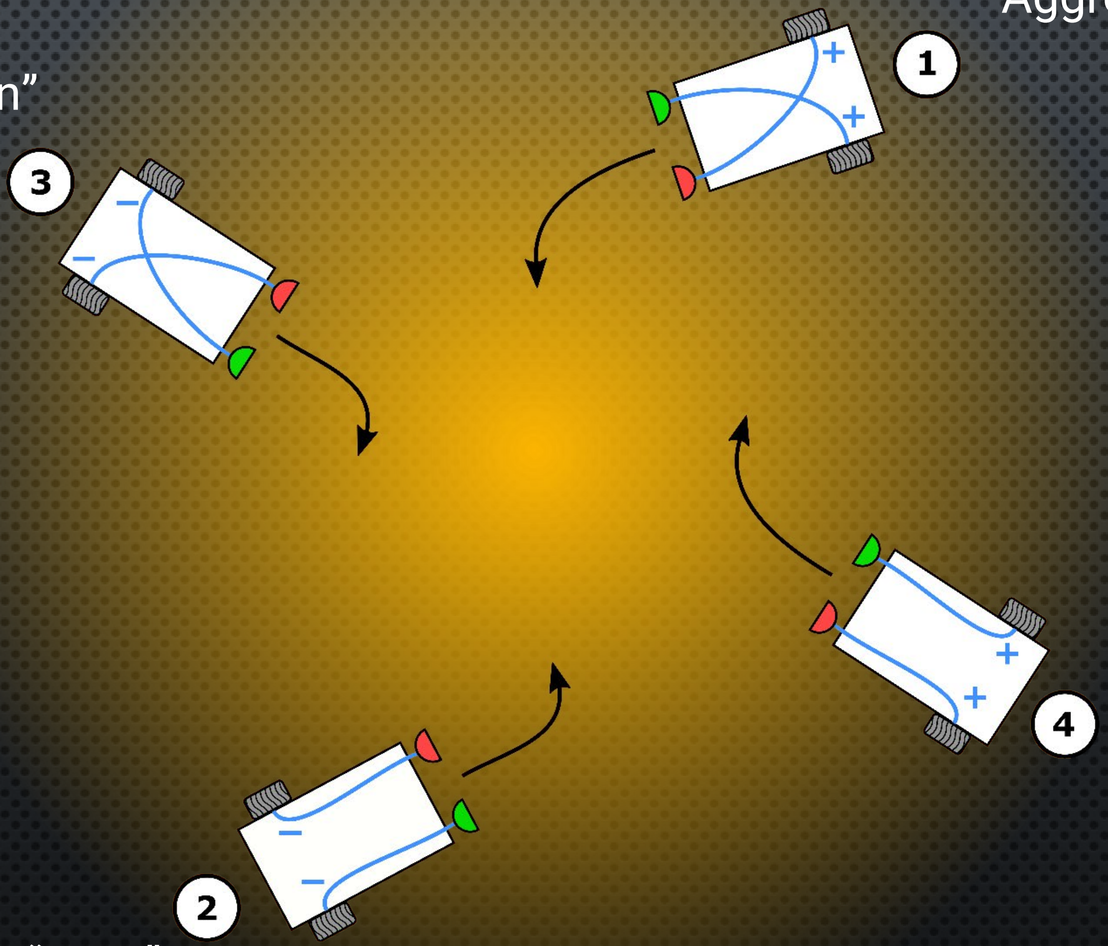
“Exploration”

OR

$$v_{left} = k \times (s_{max} - s_{right})$$
$$v_{right} = k \times (s_{max} - s_{left})$$

“Aggression”

$$v_{left} = k \times s_{right}$$
$$v_{right} = k \times s_{left}$$



$$v_{left} = \frac{1}{k} \times \frac{1}{s_{left}}$$
$$v_{right} = \frac{1}{k} \times \frac{1}{s_{right}}$$

OR

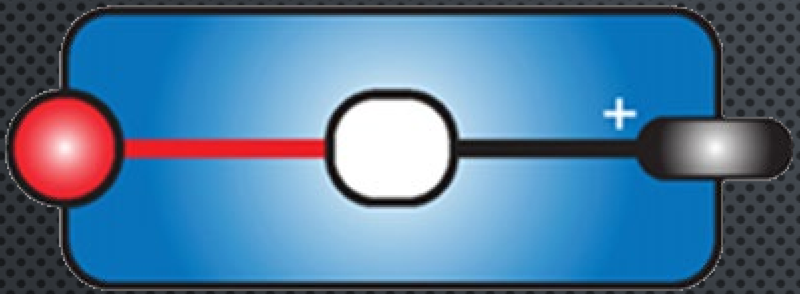
$$v_{left} = k \times (s_{max} - s_{left})$$
$$v_{right} = k \times (s_{max} - s_{right})$$

“Love”

“Fear”

$$v_{left} = k \times s_{left}$$
$$v_{right} = k \times s_{right}$$

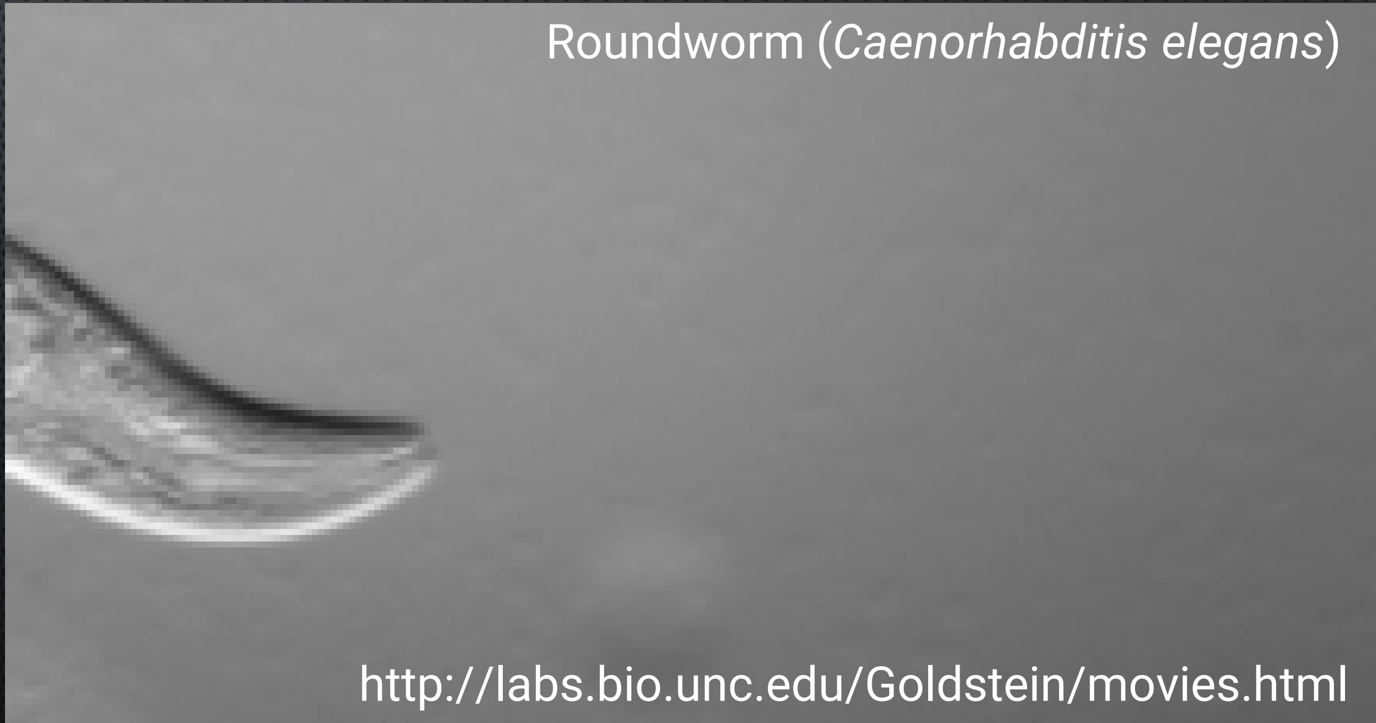
Braitenberg vehicles as animal models



Alex Gomez-Marin et al., 2010

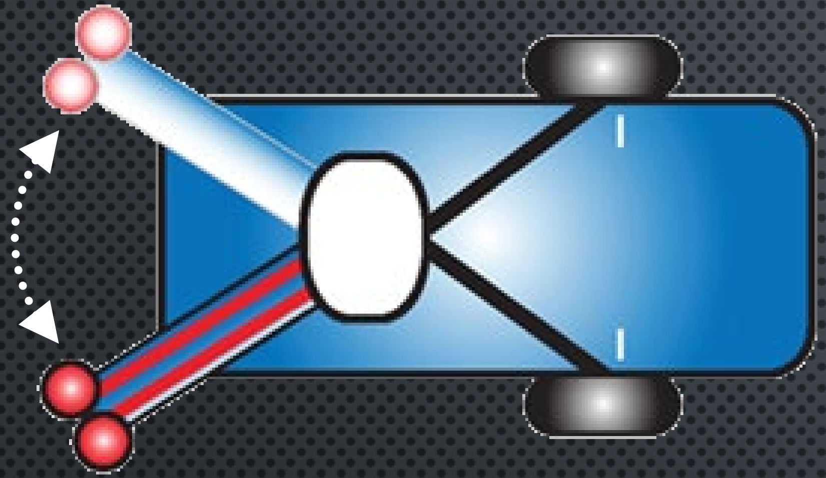
Escherichia coli

Roundworm (*Caenorhabditis elegans*)



http://labs.bio.unc.edu/Goldstein/movies.html

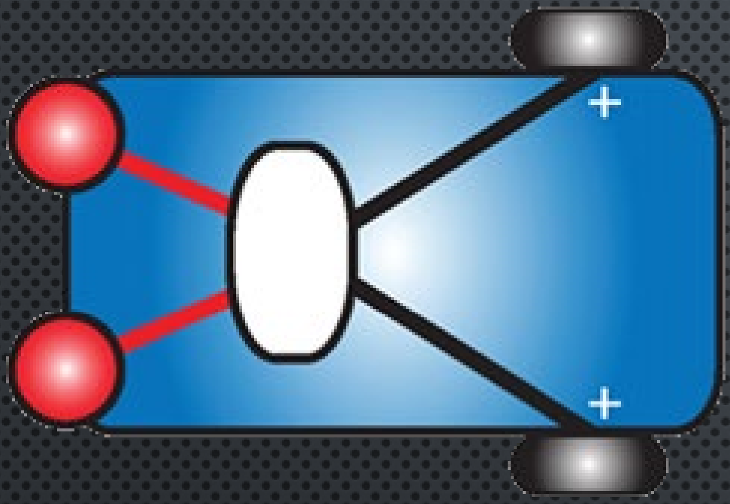
Braitenberg vehicles as animal models



Alex Gomez-Marin et al., 2010

Drosophila melanogaster or common fruit fly, larval stage

Braitenberg vehicles as animal models

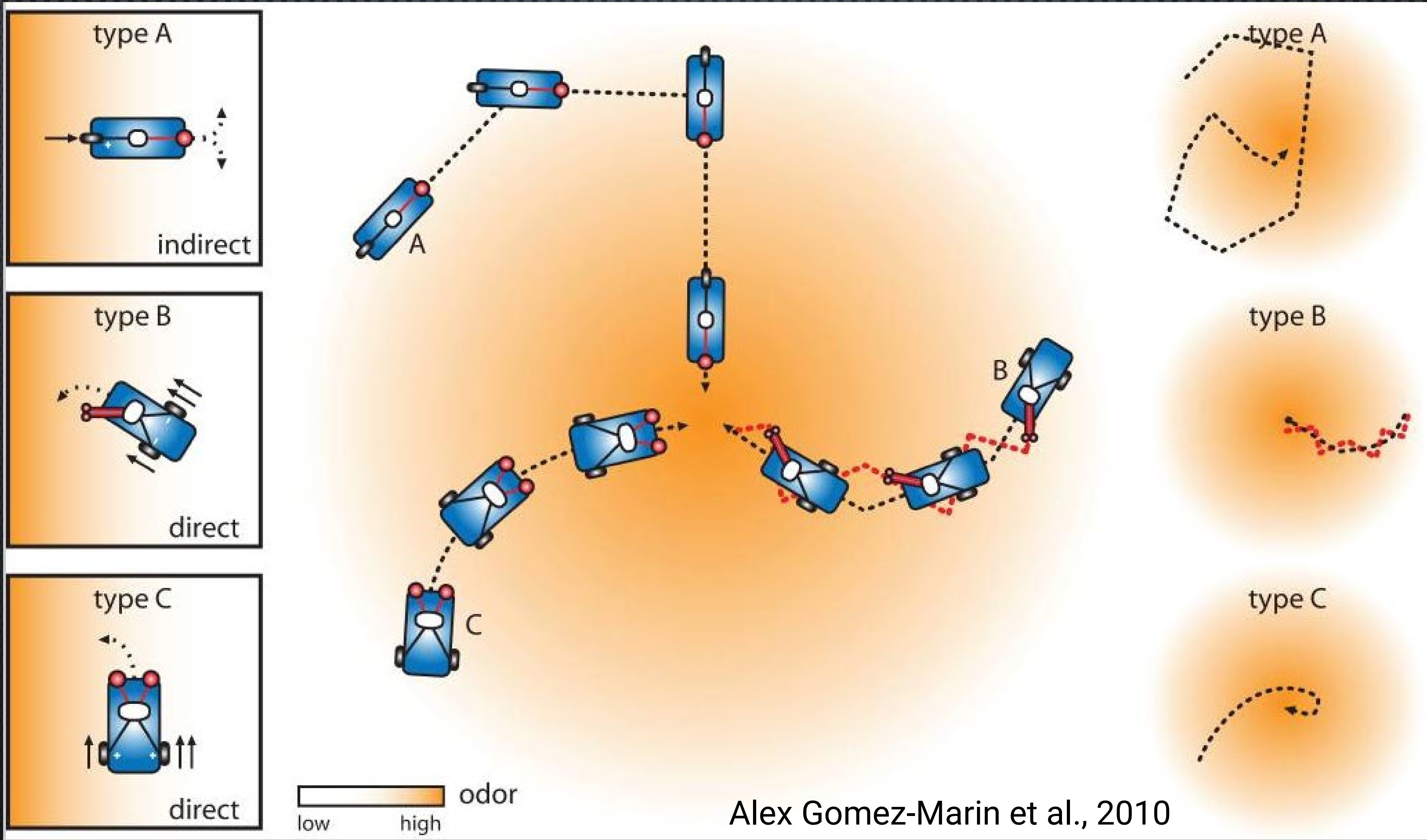


Alex Gomez-Marin et al., 2010

Sensor | Brain | Motor | Motor circuit | Sensory circuit

Drosophila melanogaster or common fruit fly, adult stage

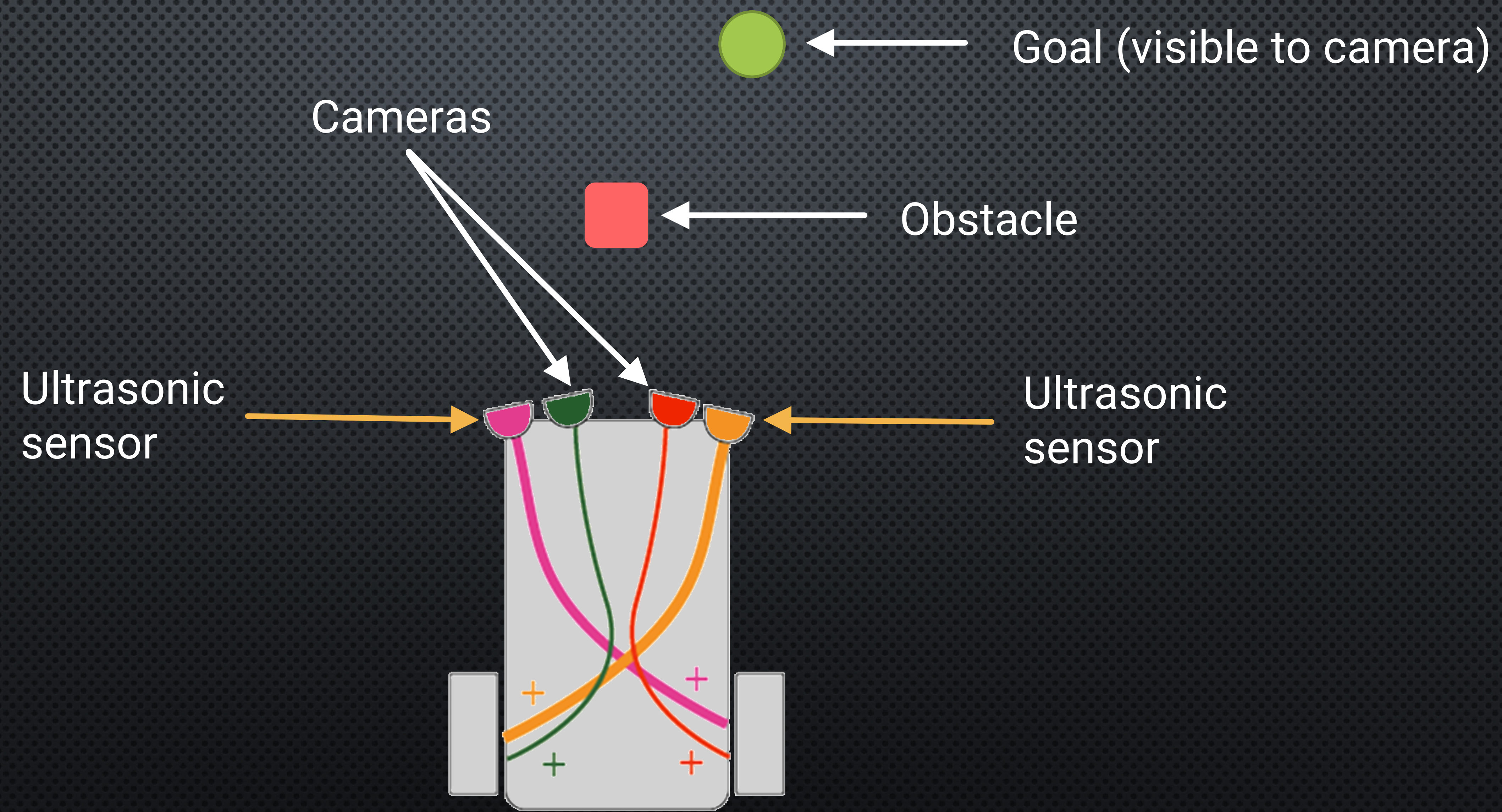
Braitenberg vehicles as animal models



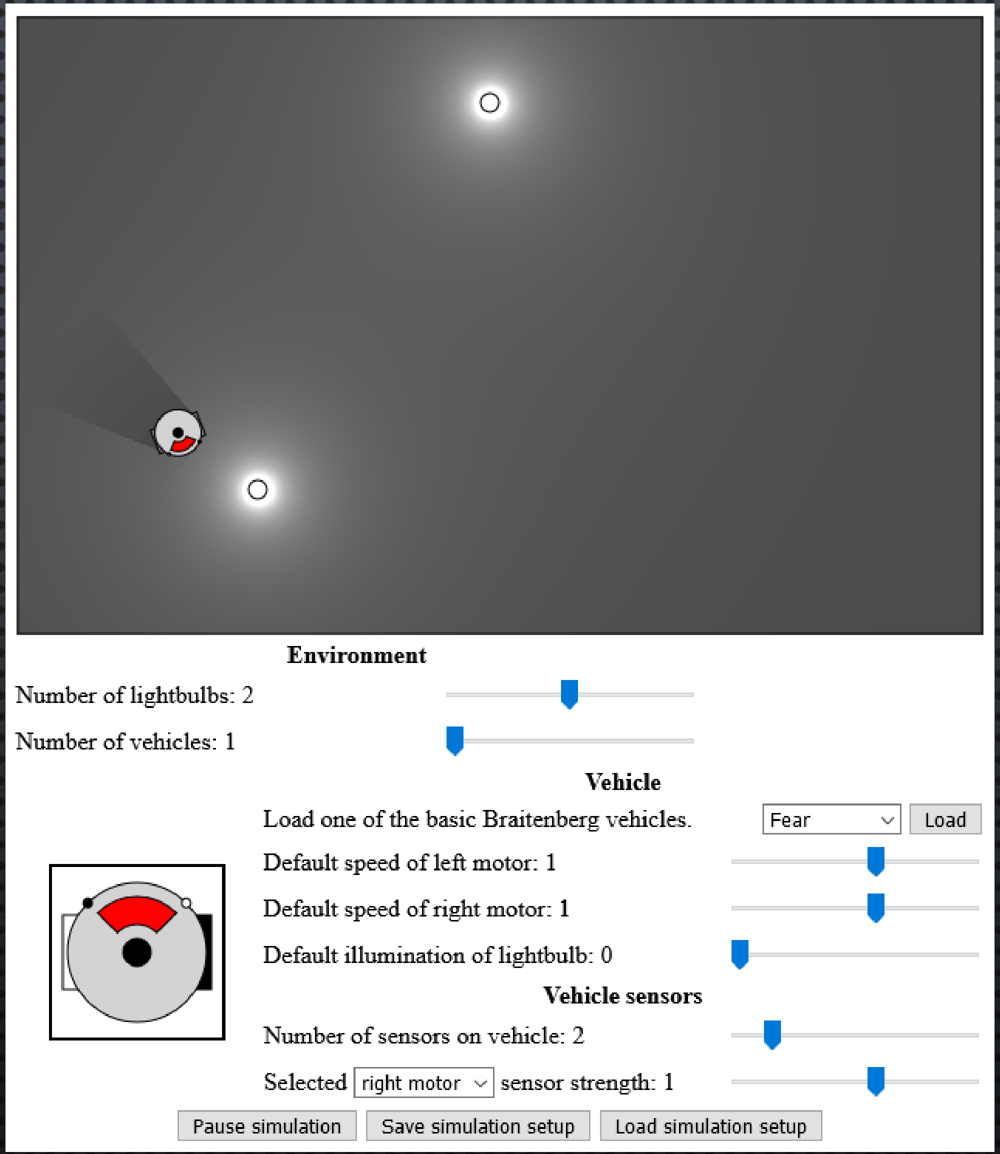
Alex Gomez-Marin et al., 2010

Some examples

How will this vehicle behave?



Play time: Braitenberg vehicle playground



http://www.harmendeweerd.nl/braitenberg-vehicles/

A case for Braitenberg controllers

The artificial solution

https://www.youtube.com/watch?v=umRdt3zGgpU

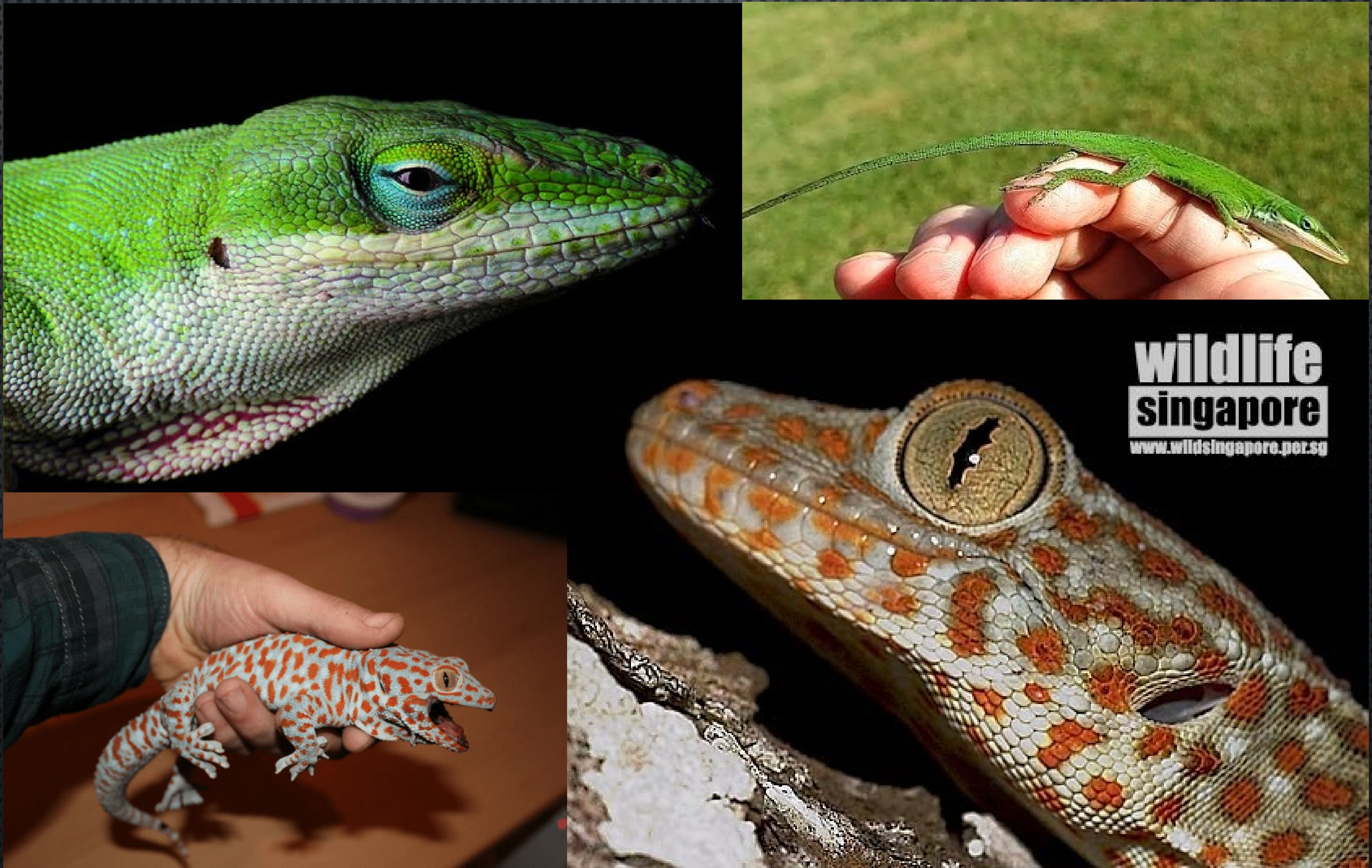
The natural solution

https://www.youtube.com/watch?v=p-_RHRAzUHM

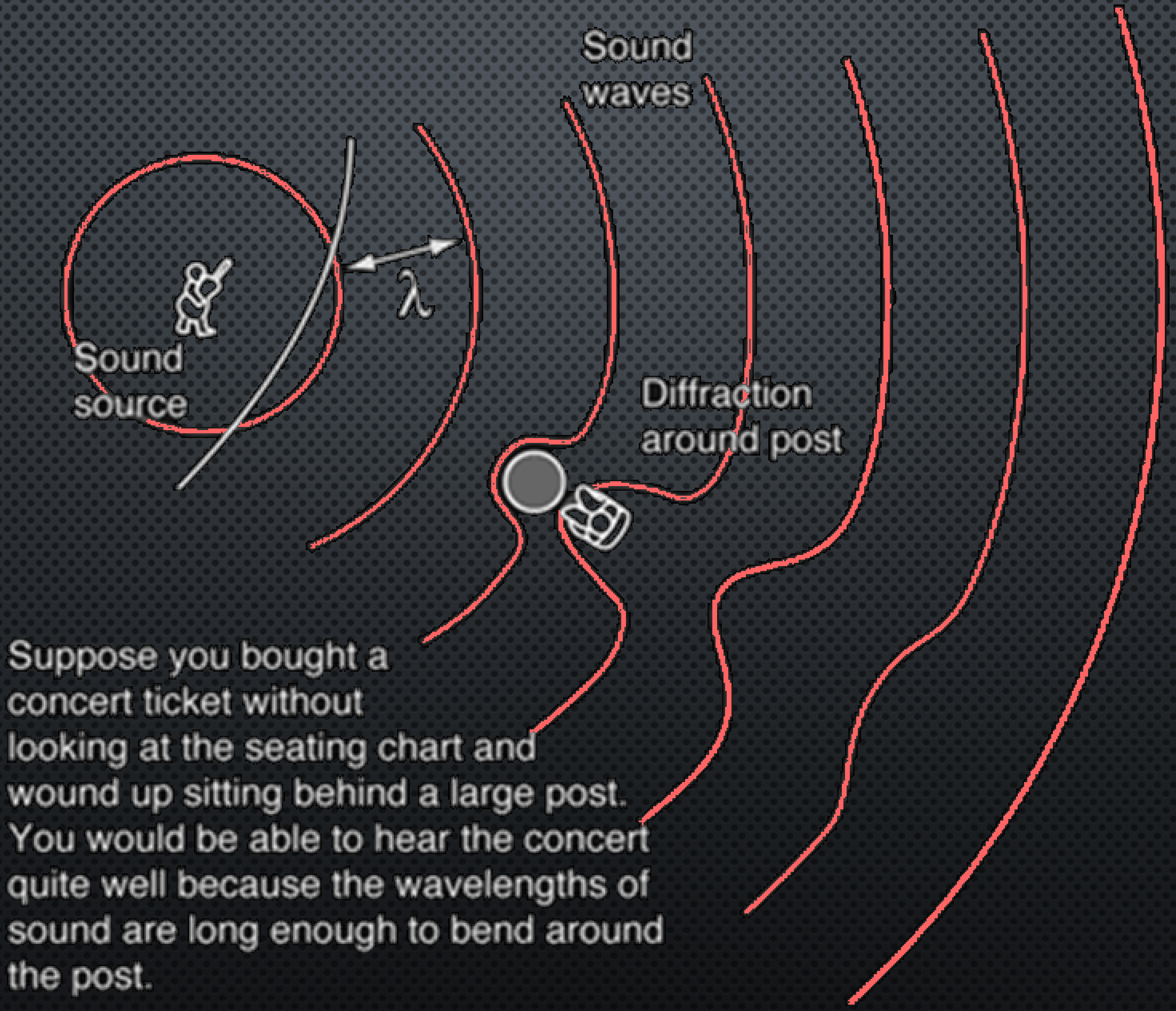
Case study 1

A lizard-inspired sound localising robot

Lizard audition

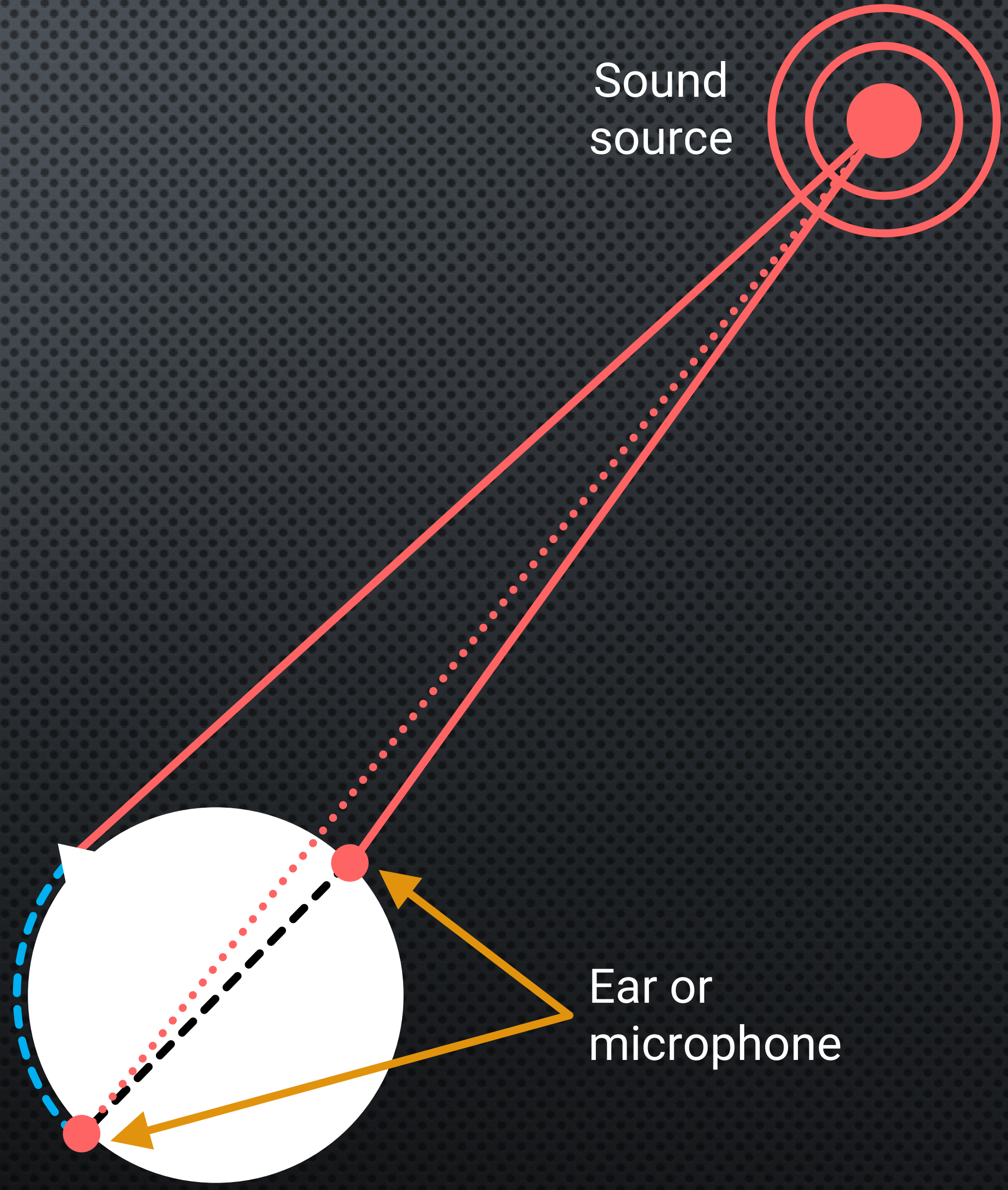


Ecological niche: physics of sound propagation



Basics of acoustics: diffraction

- Two ways to determine sound direction
  - Inter-aural level difference (ILD)
  - Inter-aural time difference (ITD)
- ILD = difference in sensed loudness (in dB) of sound at each ear
- ITD = difference in time of arrival (in μs) of sound at each ear
- If sound frequency is high (wavelength > distance between ears), ILD is significant
- If sound frequency is low (wavelength < distance between ears) ITD is significant

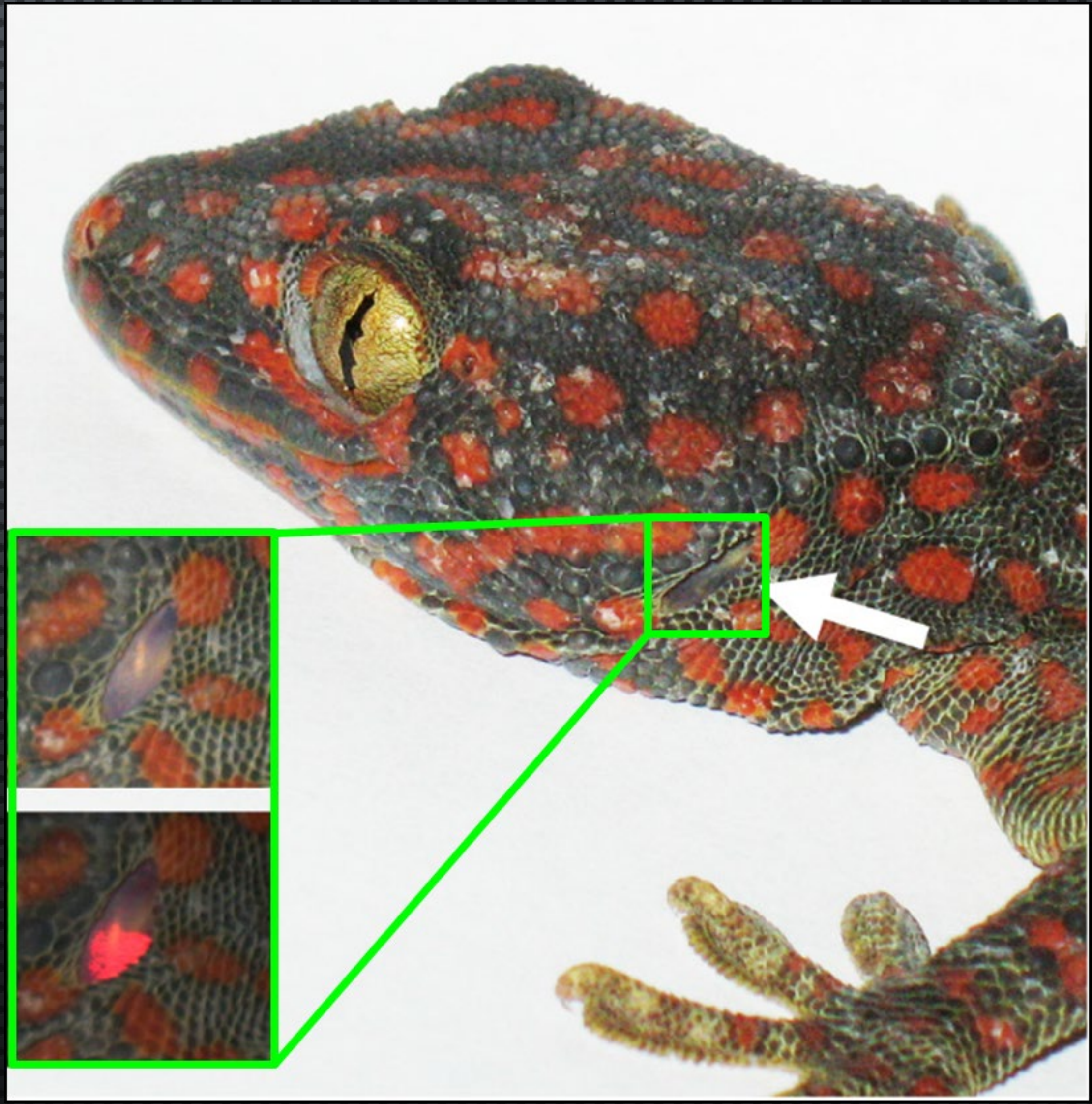


Characteristics of lizard hearing

- Small size (10—20 mm) w.r.t sound wavelengths (85–340 mm) → sound diffracts around the head

- Diffraction leads to small inter-aural level difference (ILD) (typically 1-2 dBs → too small to measure without special, big, heavy and expensive equipment)

- Tiny (µs scale) inter-aural time difference (ITD) information available to determine sound direction



Shaikh et al., 2016

Auditory directional cues

Image: reptilepark.com.au

Auditory directional cues

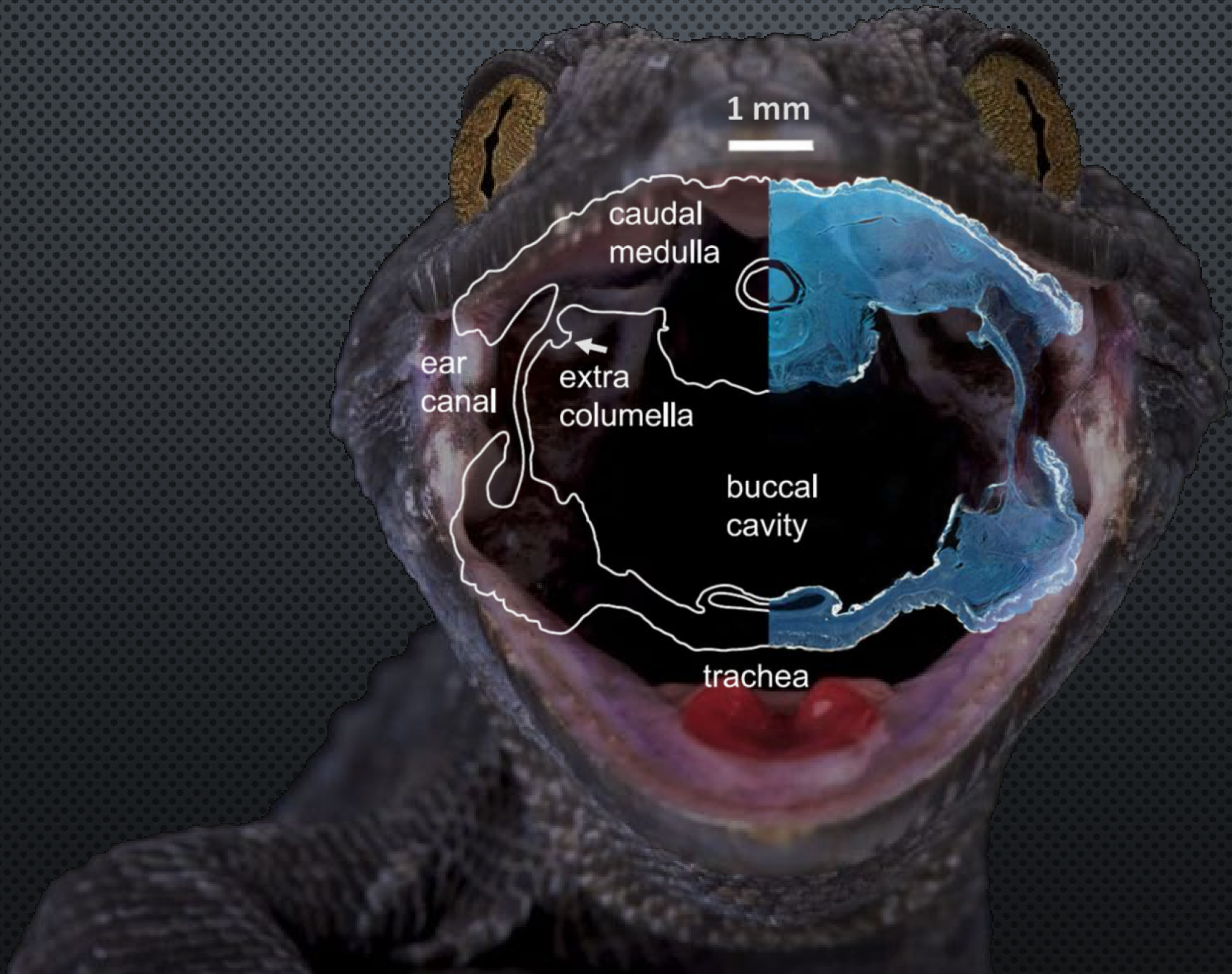


Image: reptilepark.com.au

Auditory directional cues

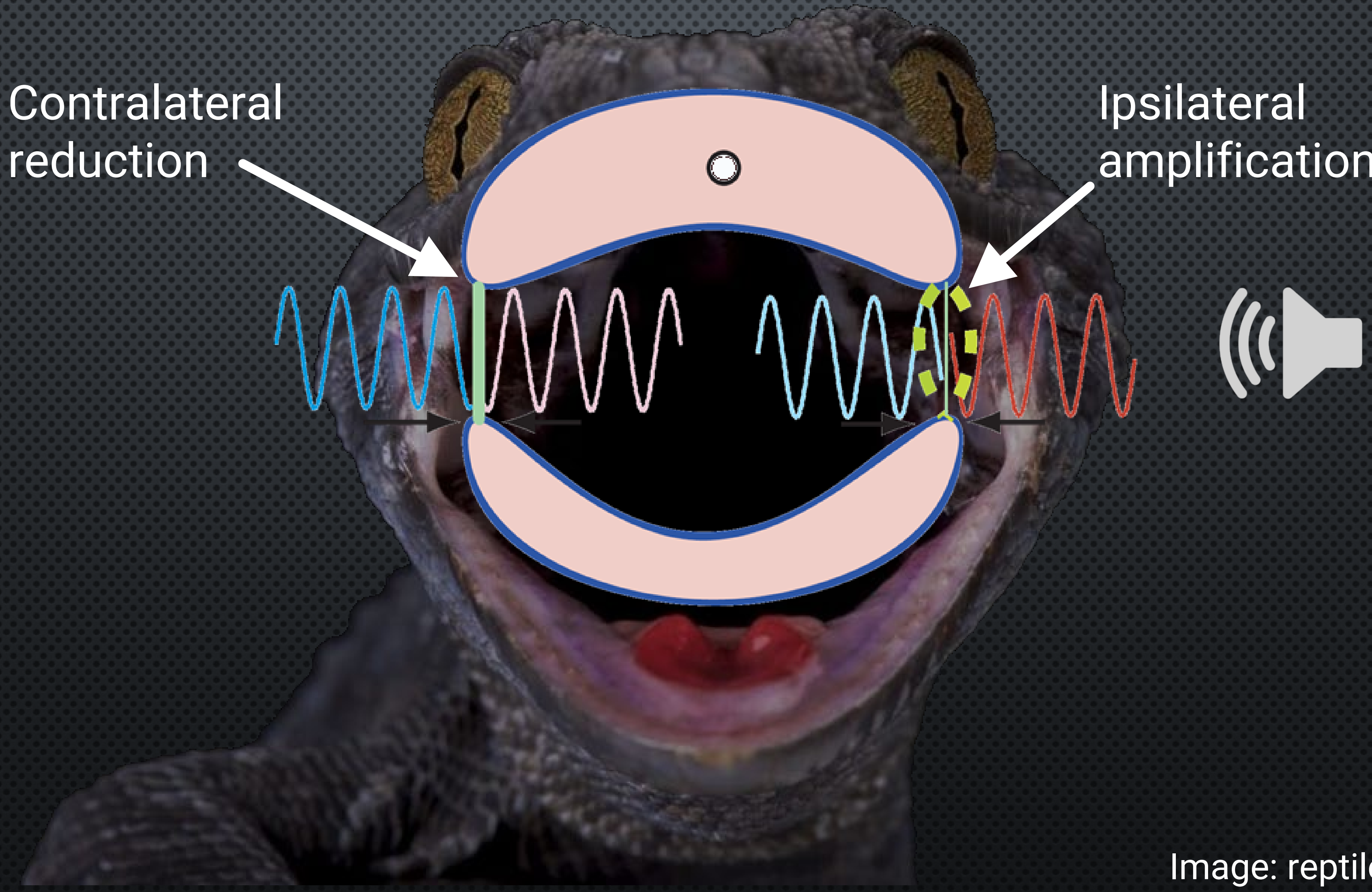
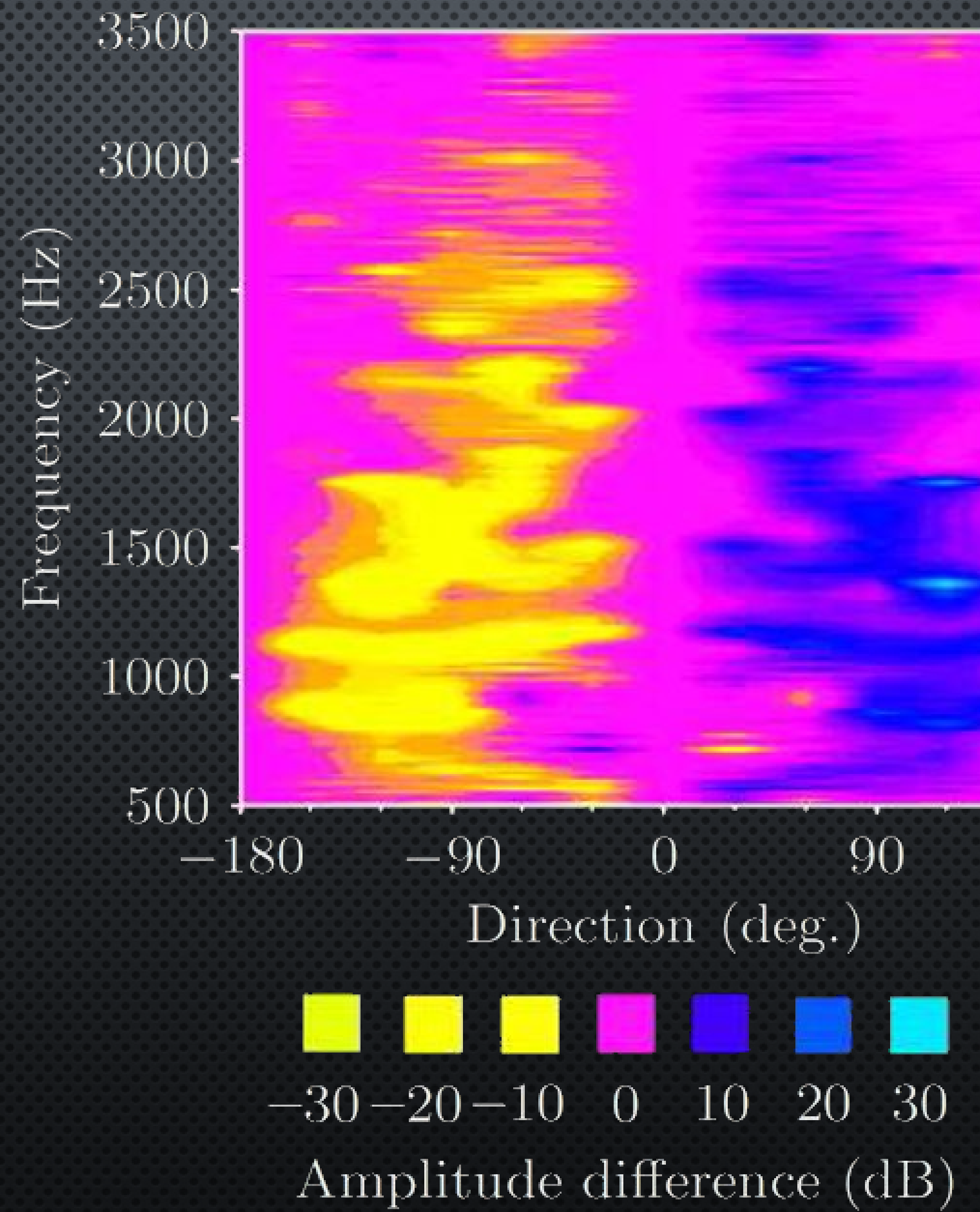
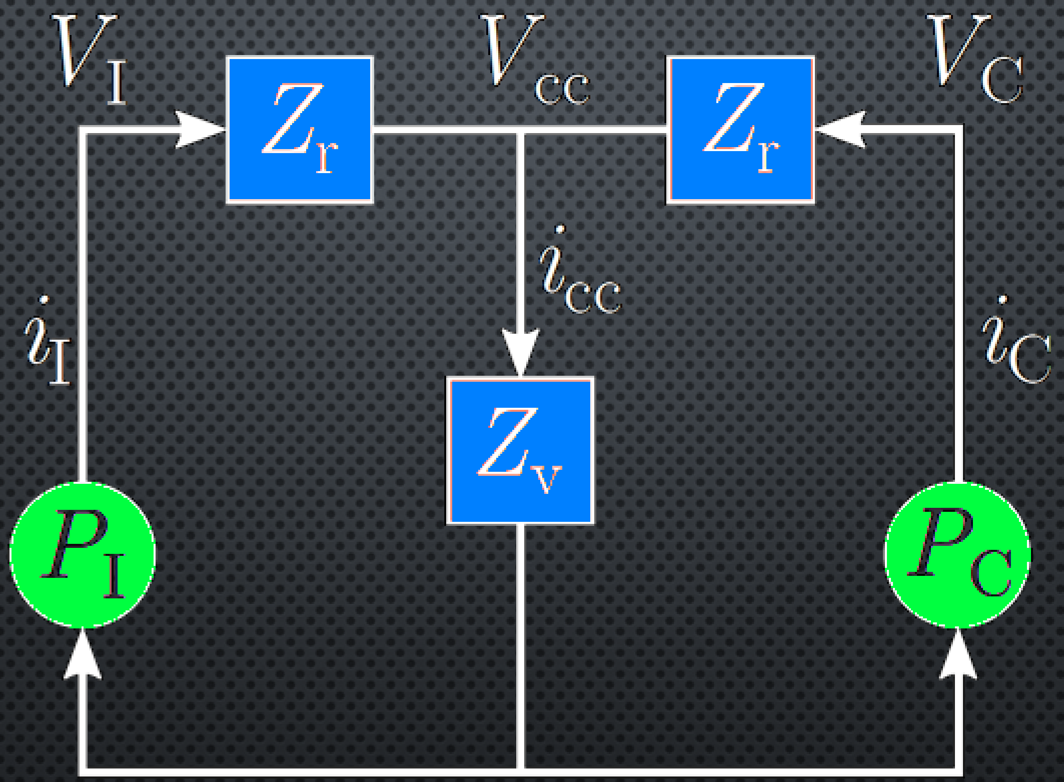


Image: reptilepark.com.au

Sound direction informtion

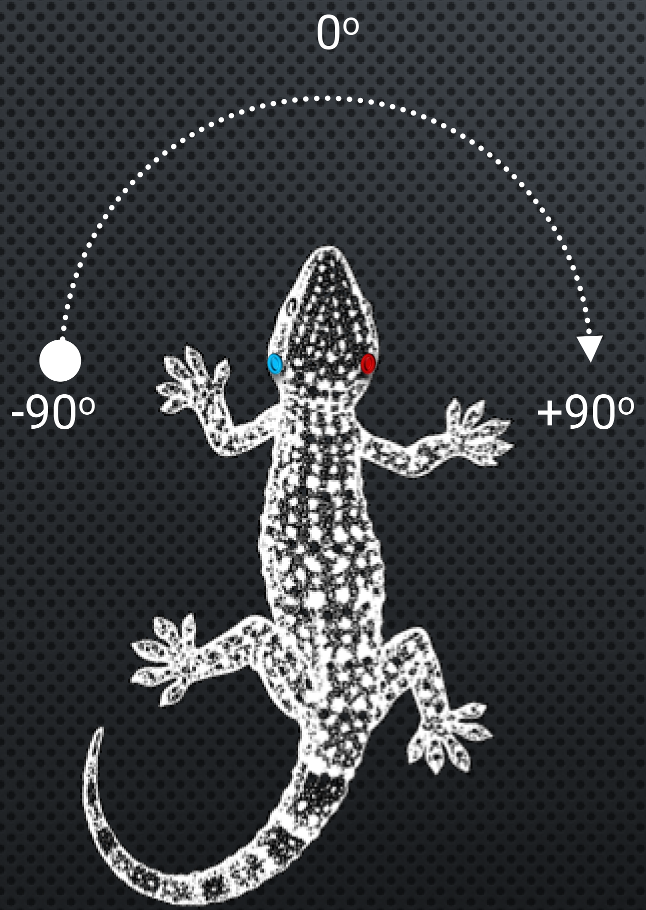


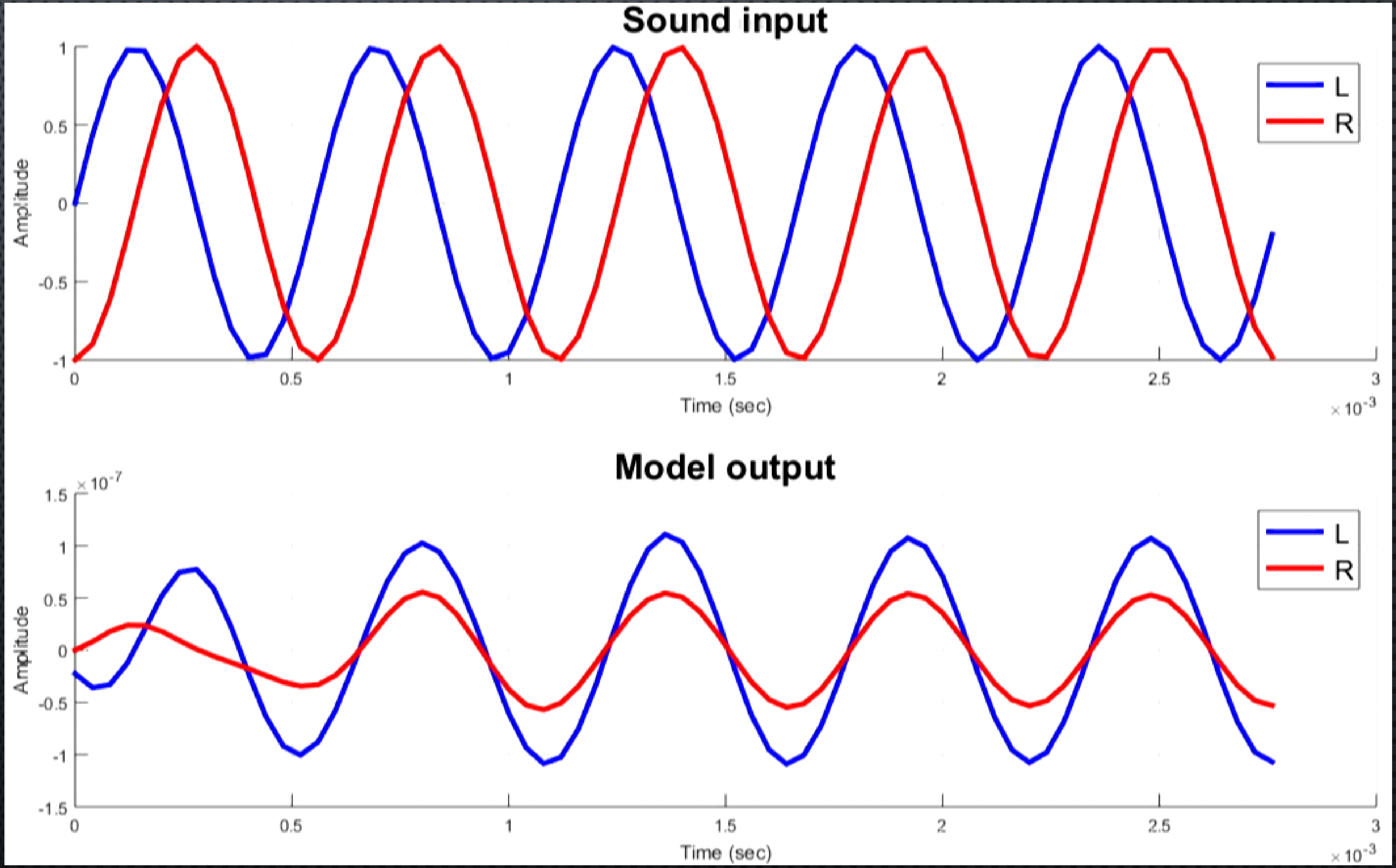
Sensor model



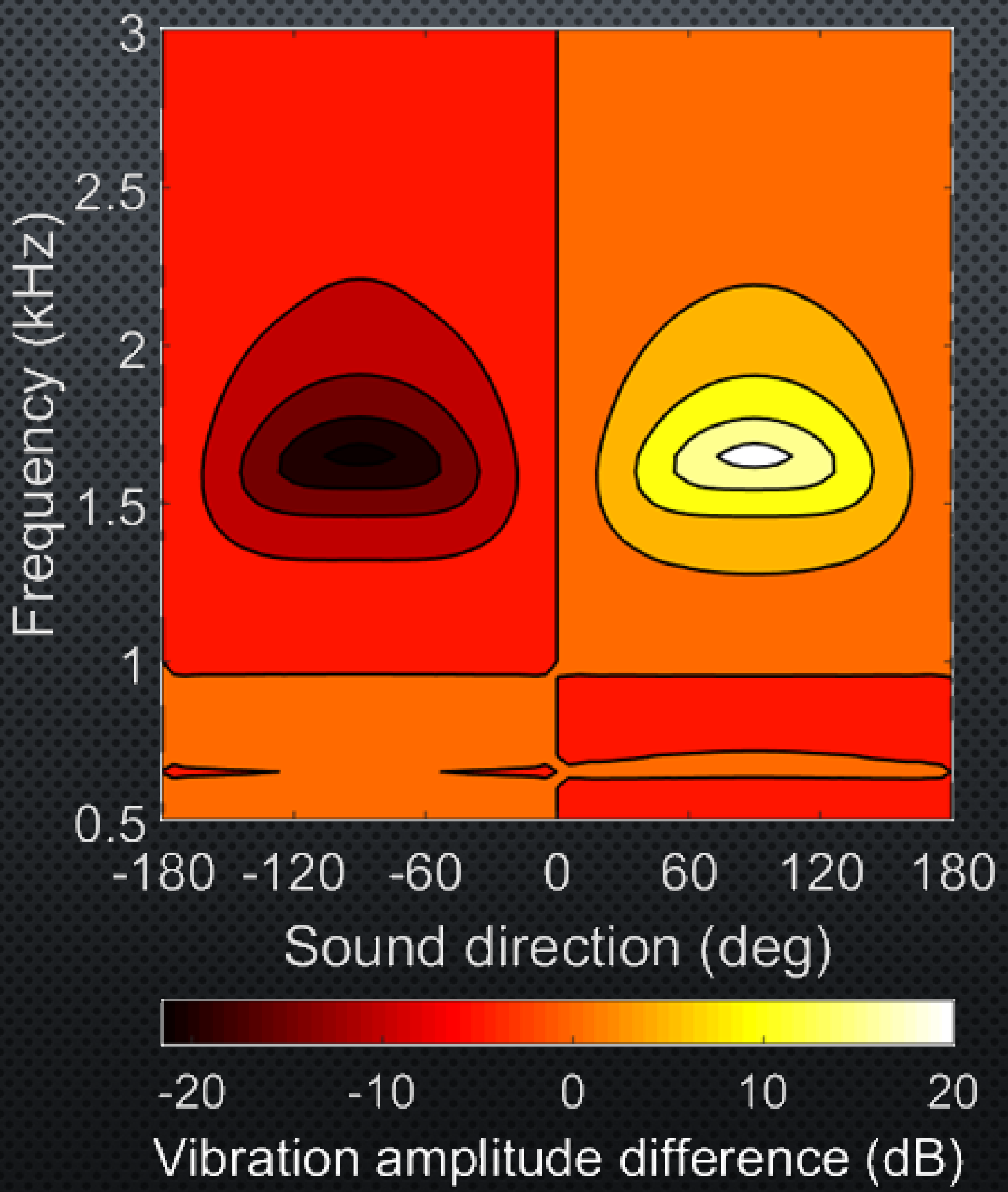
$$\left|\frac{i_{\mathrm{I}}}{i_{\mathrm{C}}}\right| = \left|\frac{G_{\mathrm{I}} \cdot V_{\mathrm{I}} + G_{\mathrm{C}} \cdot V_{\mathrm{C}}}{G_{\mathrm{C}} \cdot V_{\mathrm{I}} + G_{\mathrm{I}} \cdot V_{\mathrm{C}}}\right| \equiv 20\left(\log|i_{\mathrm{I}}| - \log|i_{\mathrm{C}}|\right)\ \mathrm{dB}$$

Model response in the azimuth

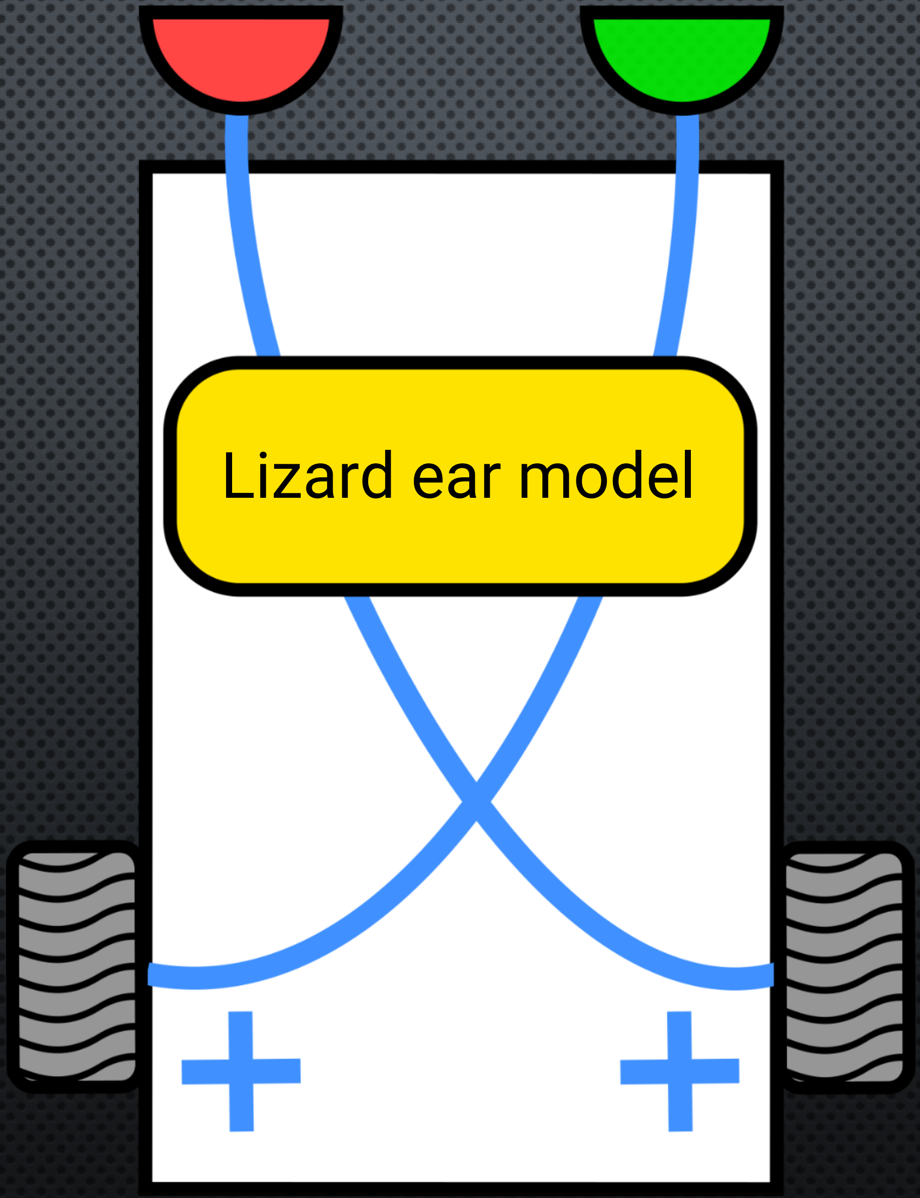




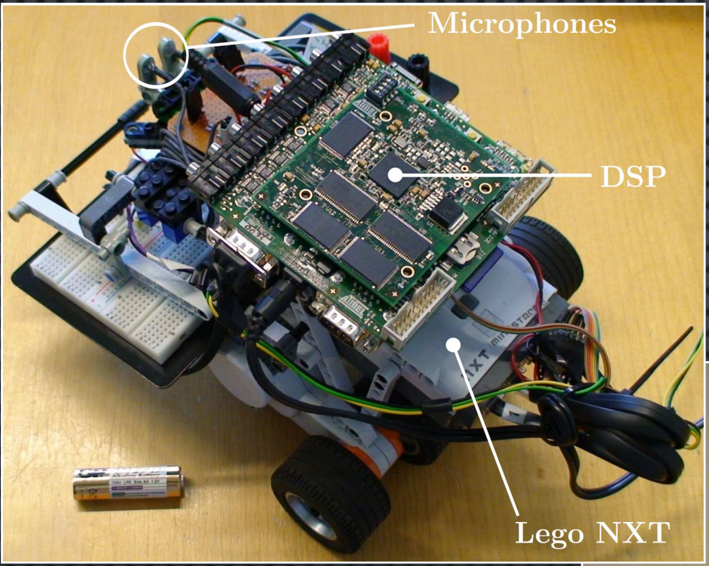
Model response in the azimuth

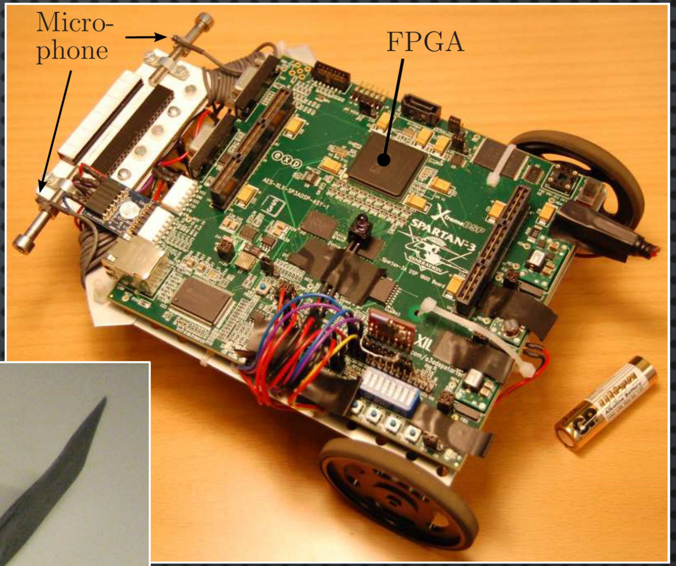


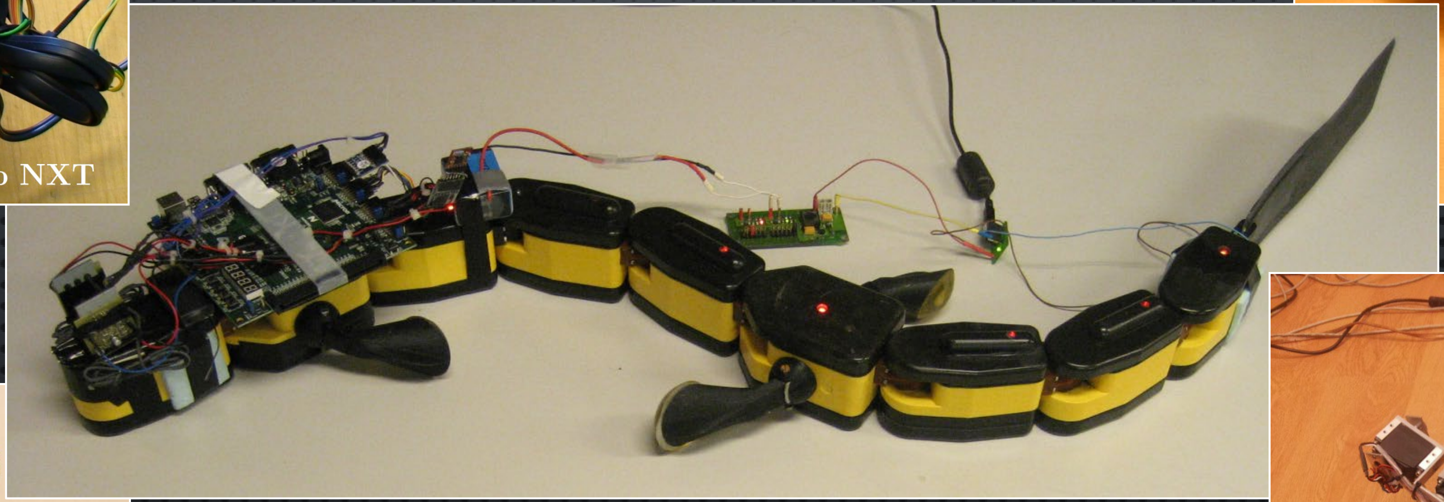
A Braitenberg model for sound localisation

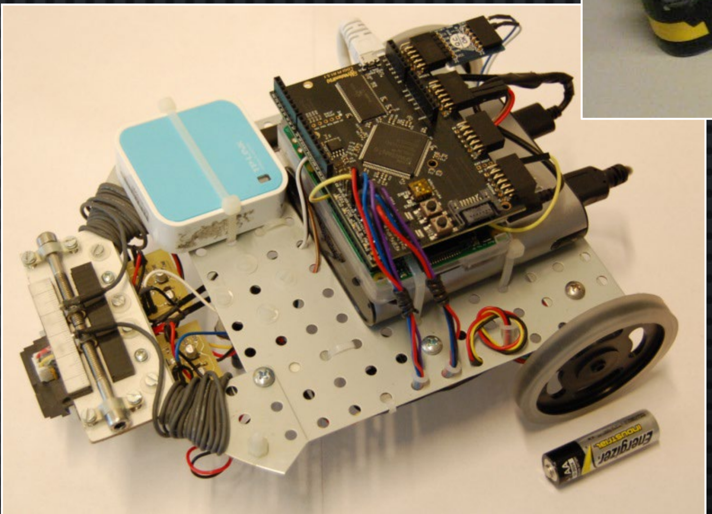


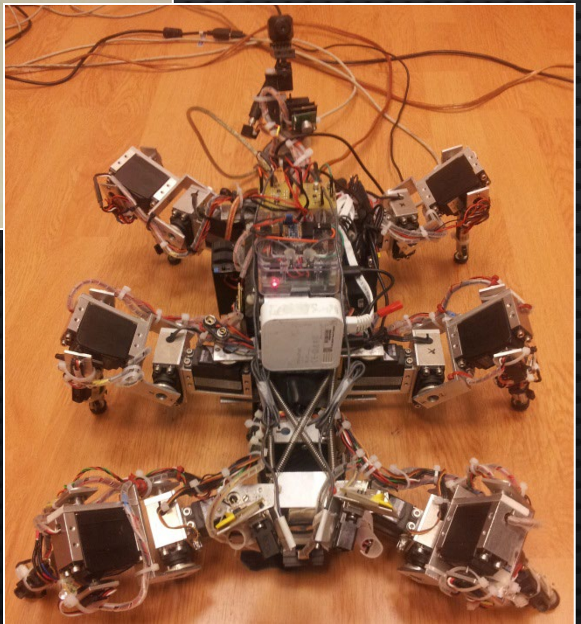
Biologically-inspired sound localising robots (by yours truly)



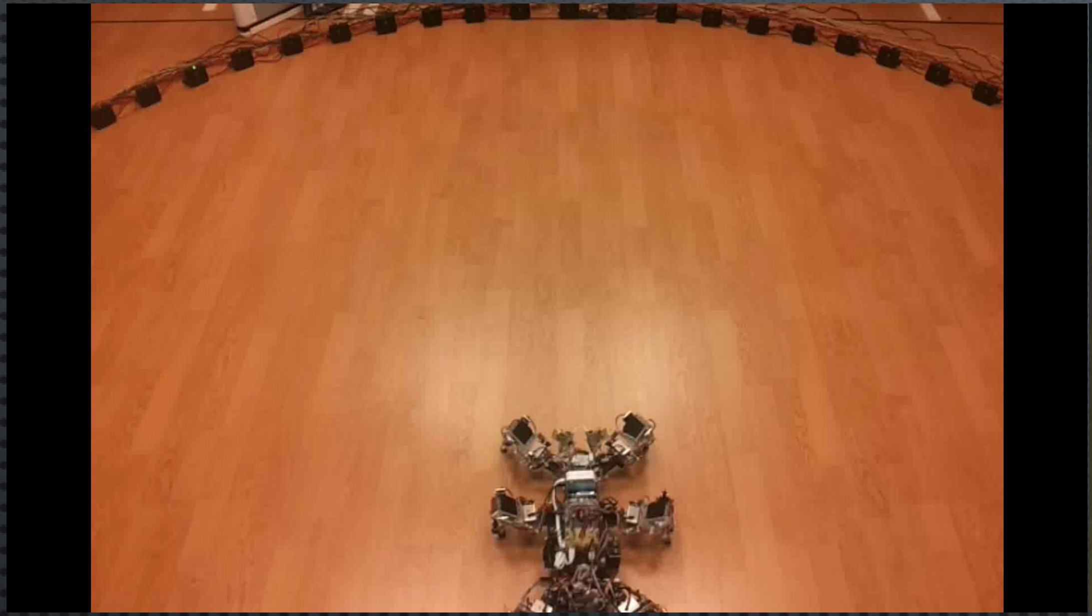
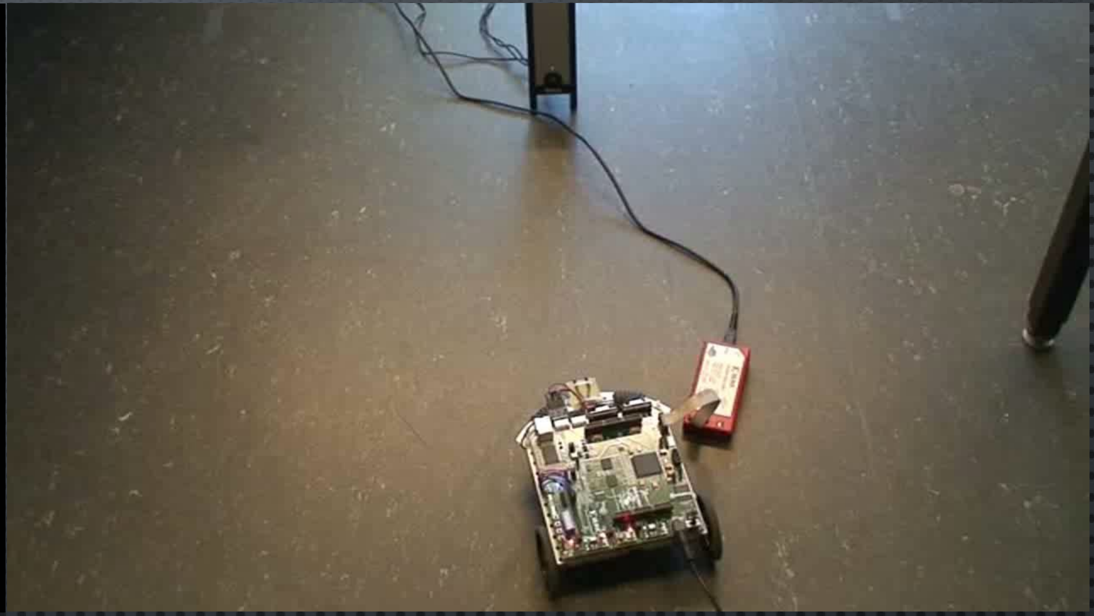


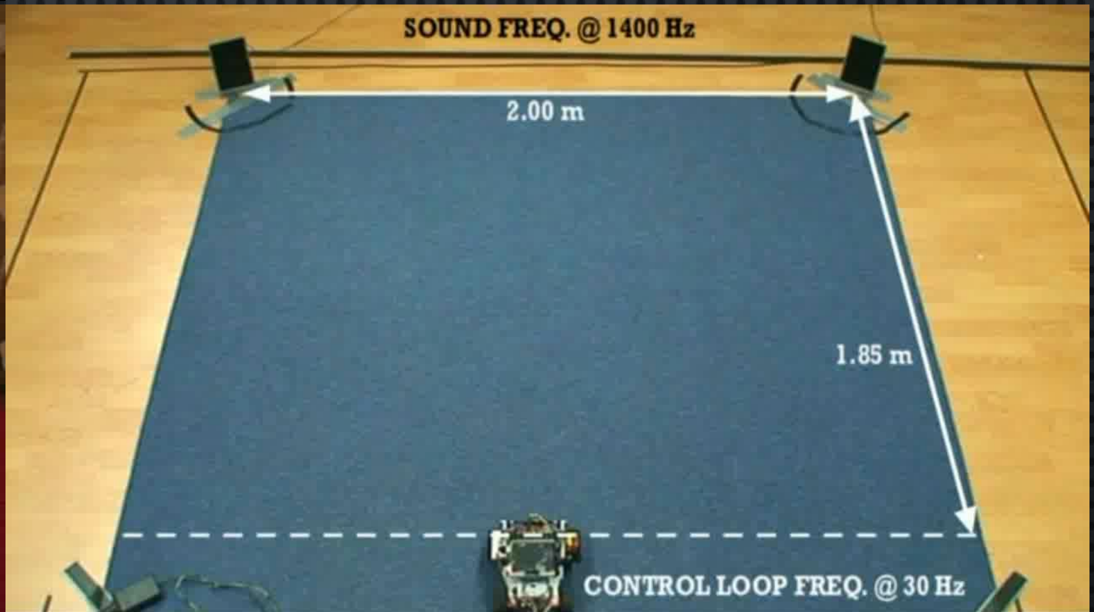






Biologically-inspired sound localising robots (by yours truly)



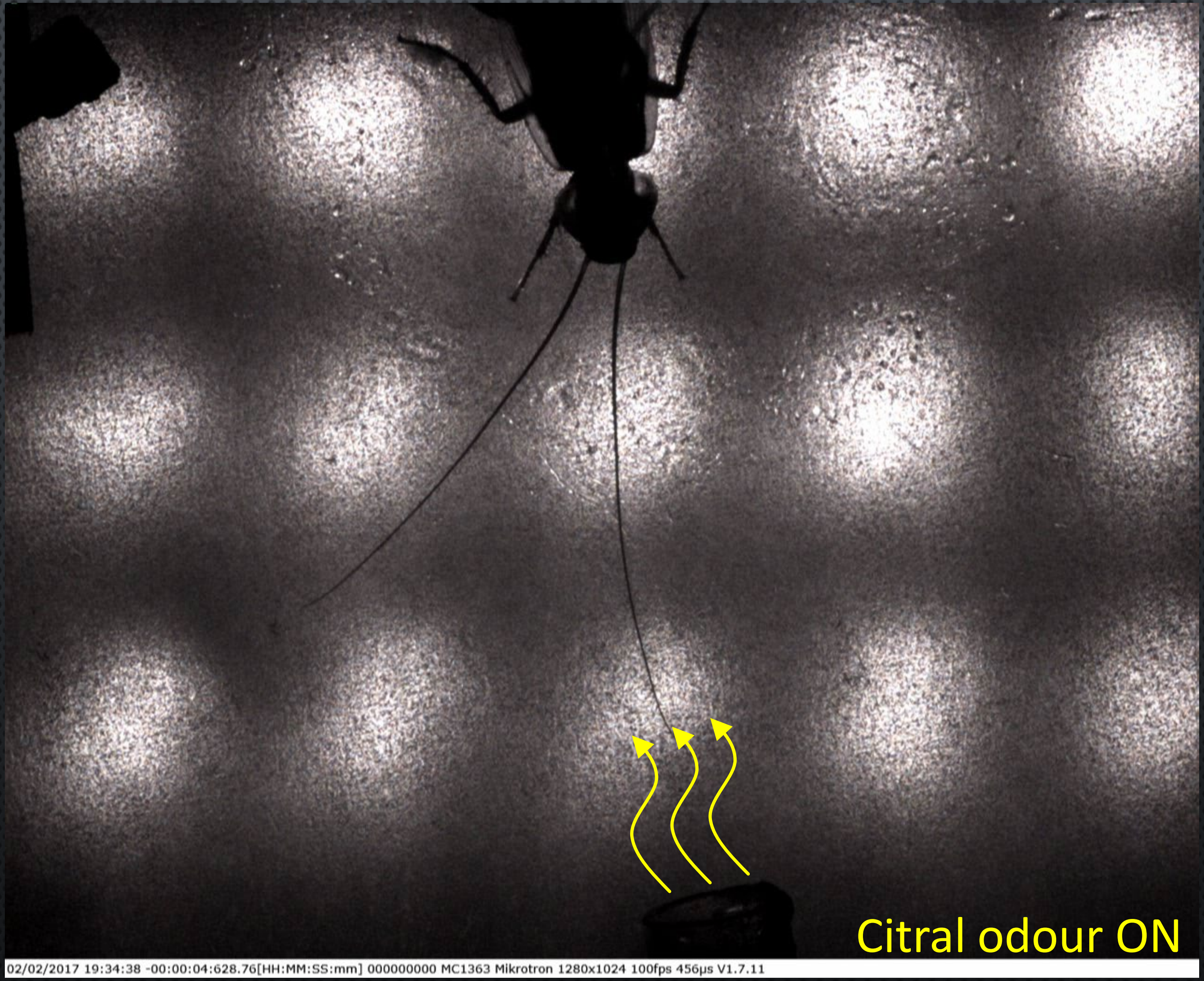
Case study 2

A cockroach-inspired gas leak localising robot

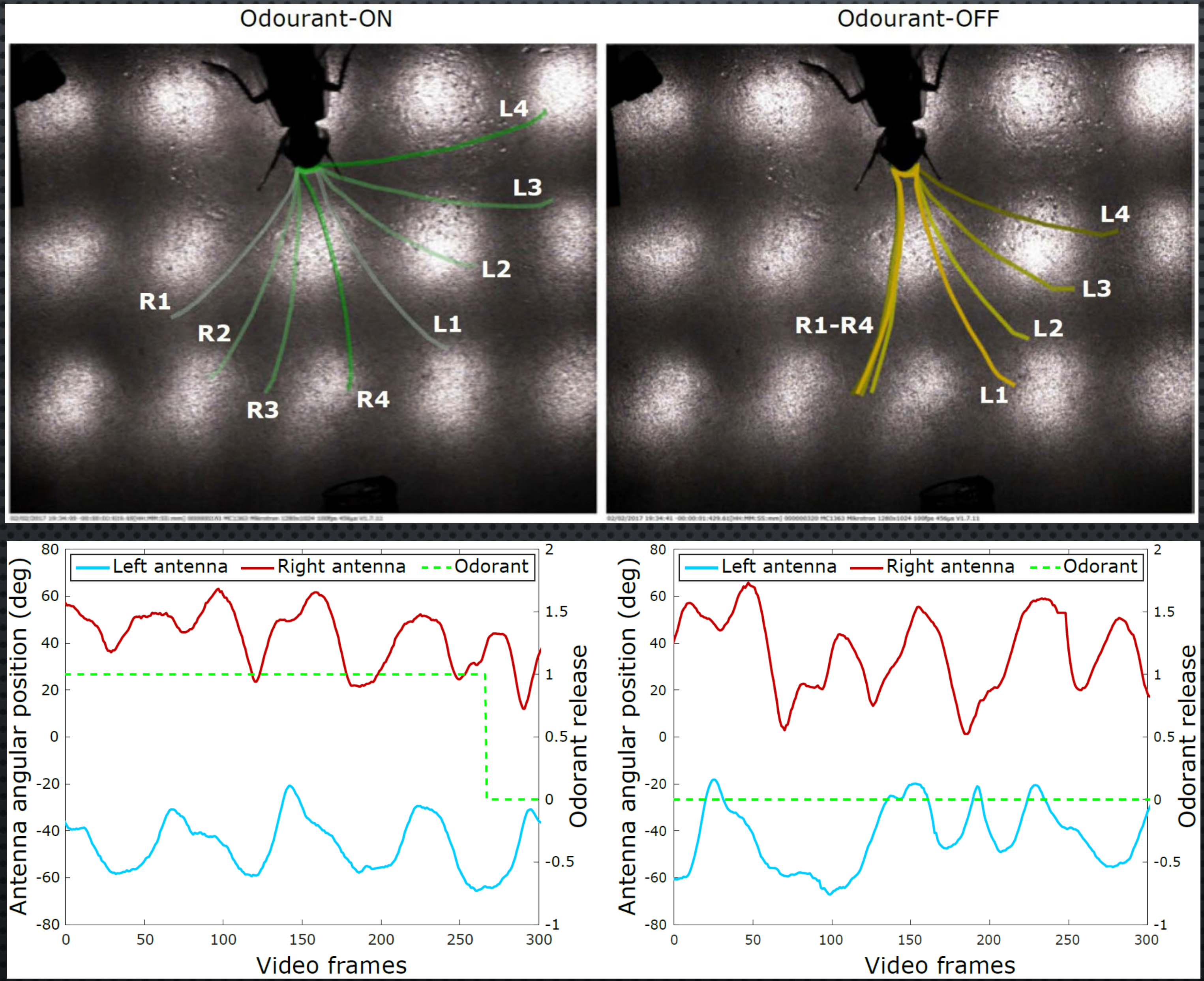
Cockroach olfaction

Observing the animal

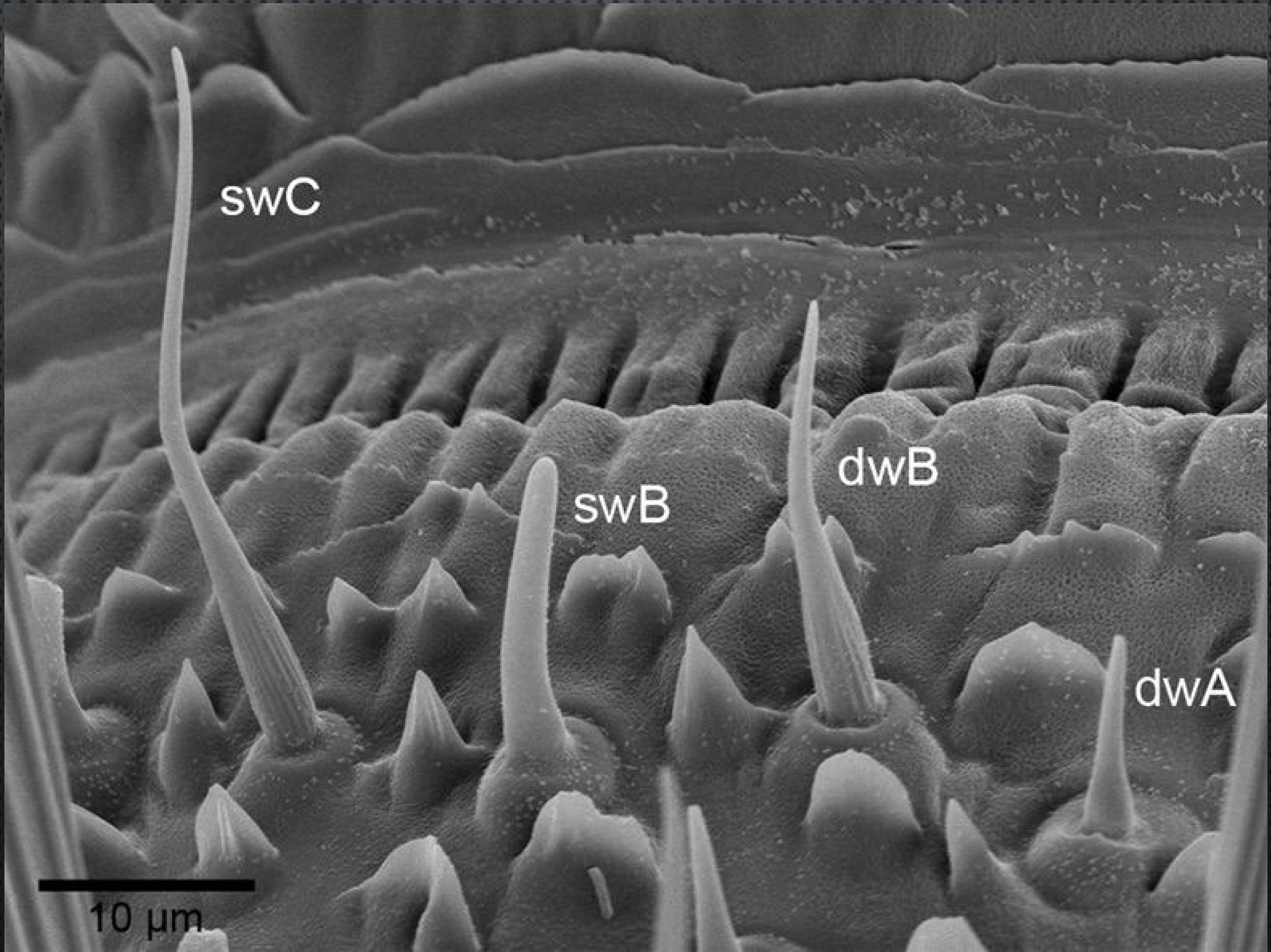
Antennal response to chemical odours



Antennal response to chemical odours

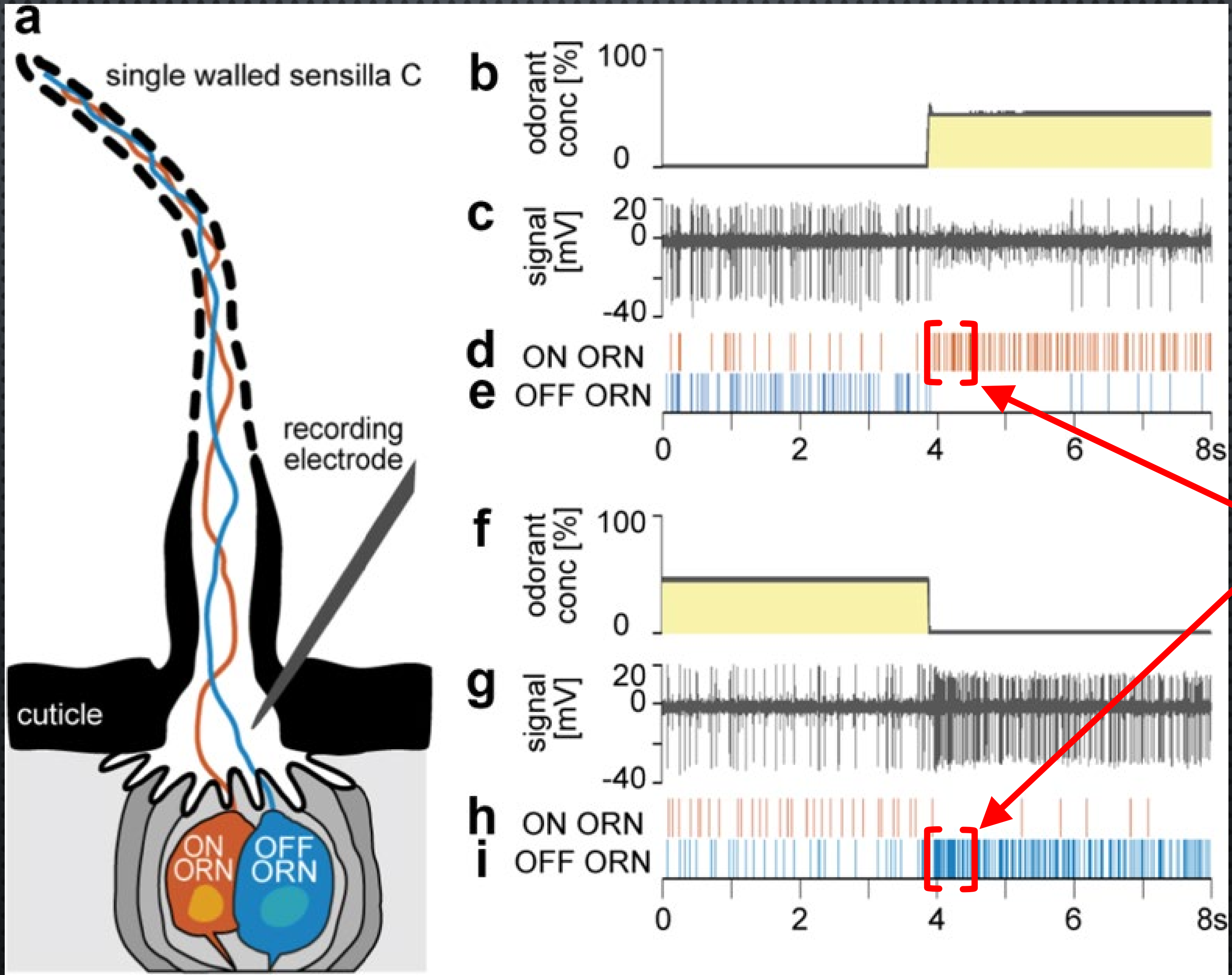


How do cockroaches smell?



Tichy & Hellwig, 2018

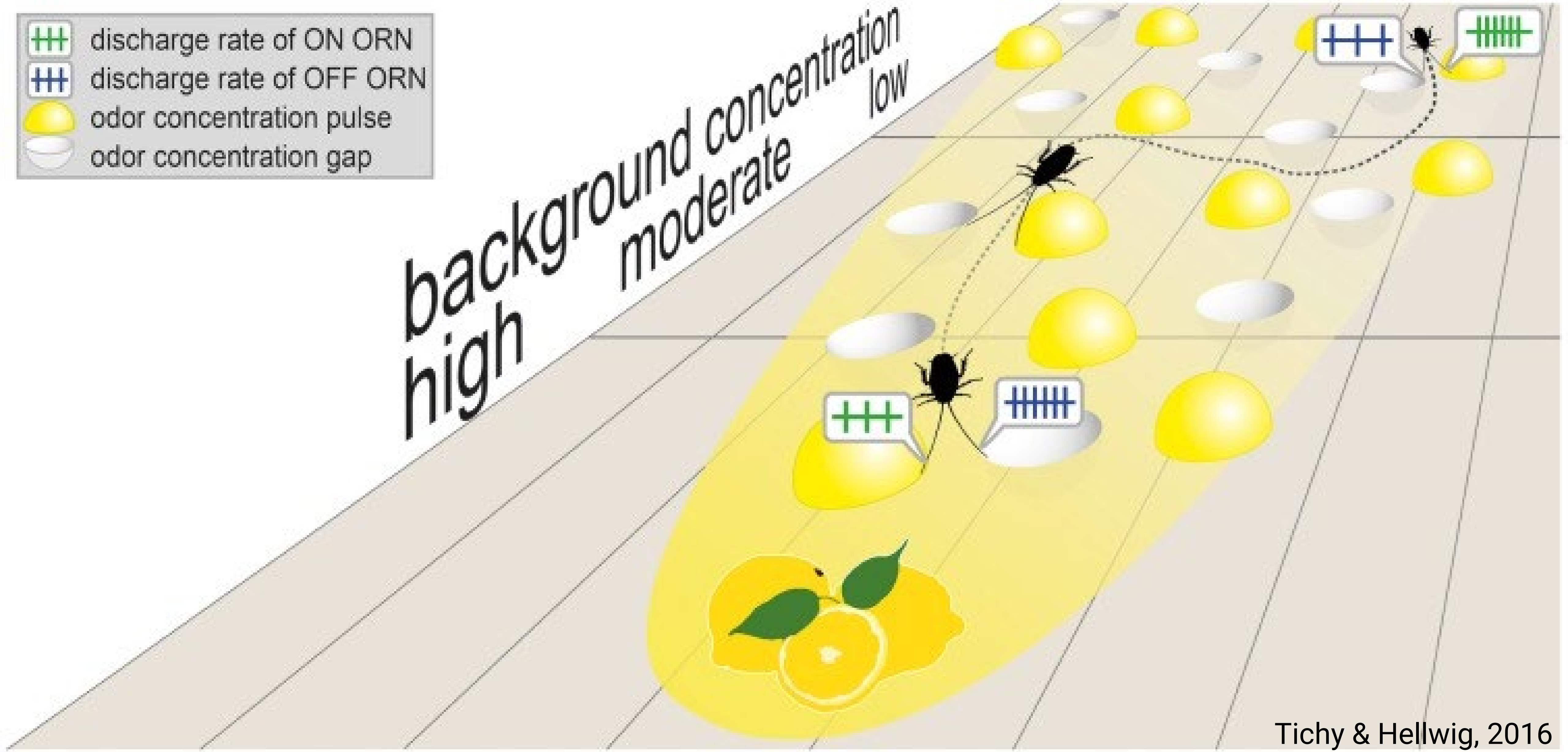
Olfactory neurons (ORNs) encode odour concentration gradient



Highest spiking due to odour gradient

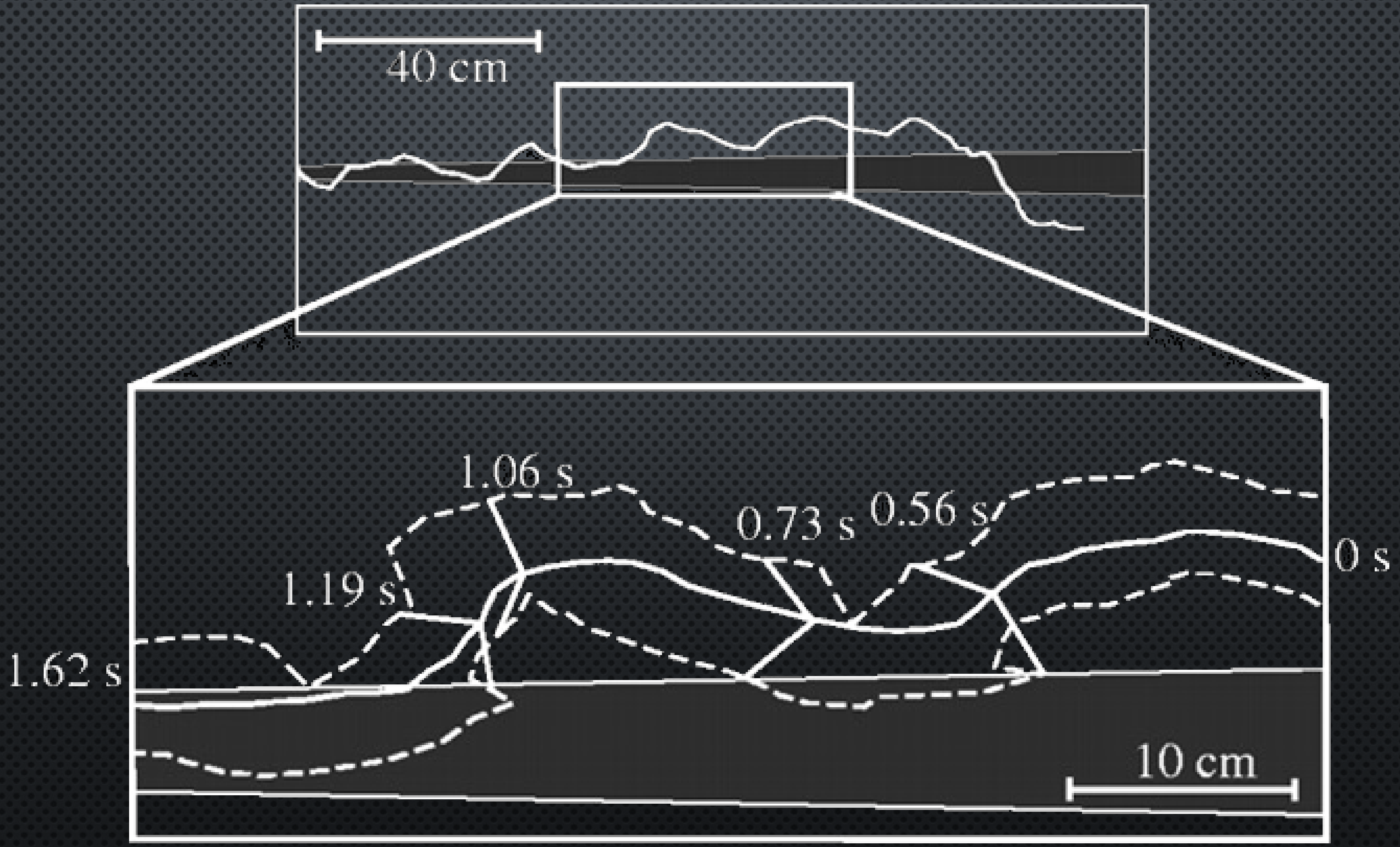
Tichy & Hellwig, 2018

From neurons to behaviour: a hypothesis



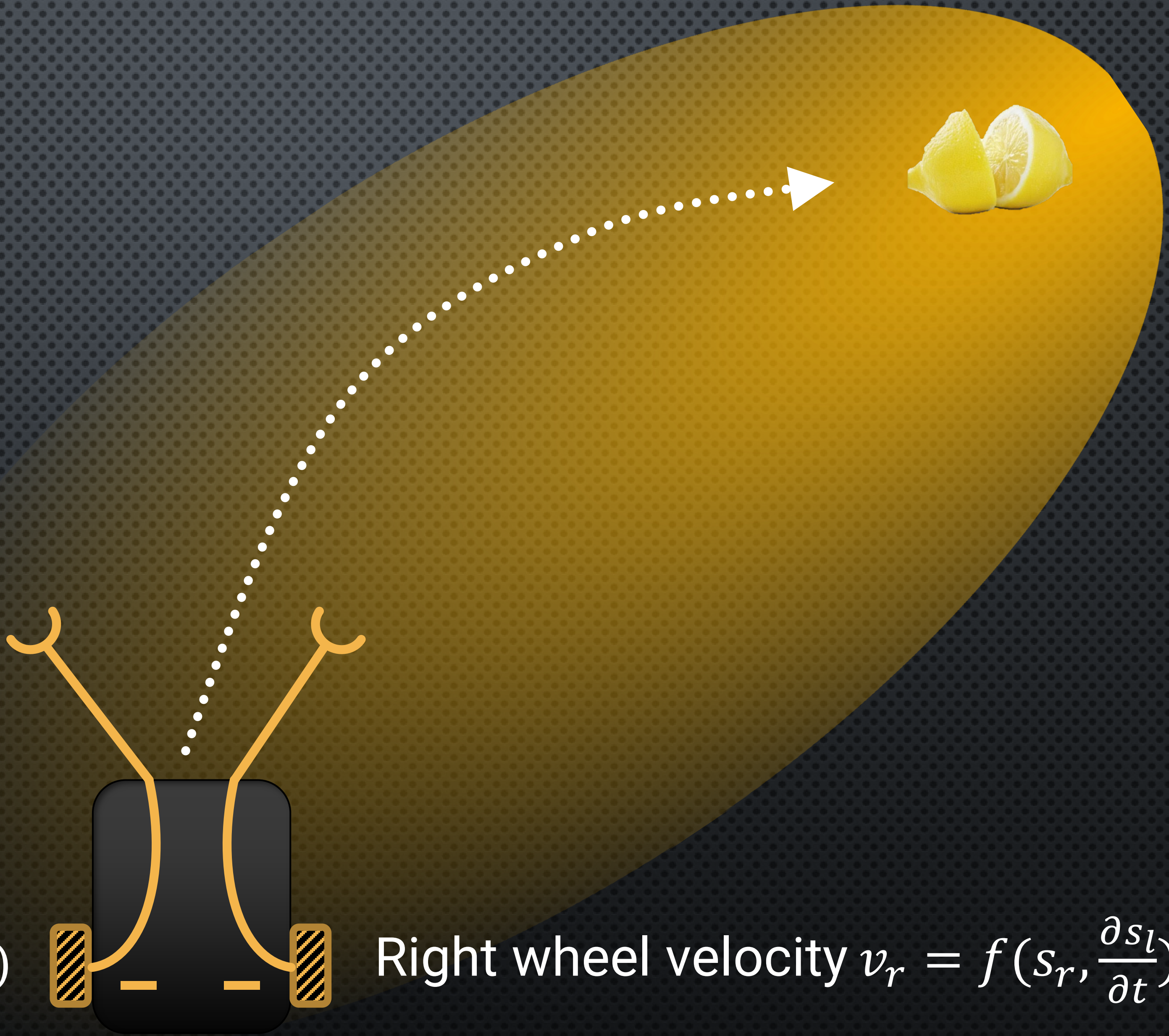
Tichy & Hellwig, 2016

Biological cockroach trajectories



Willis & Avondet, 2005

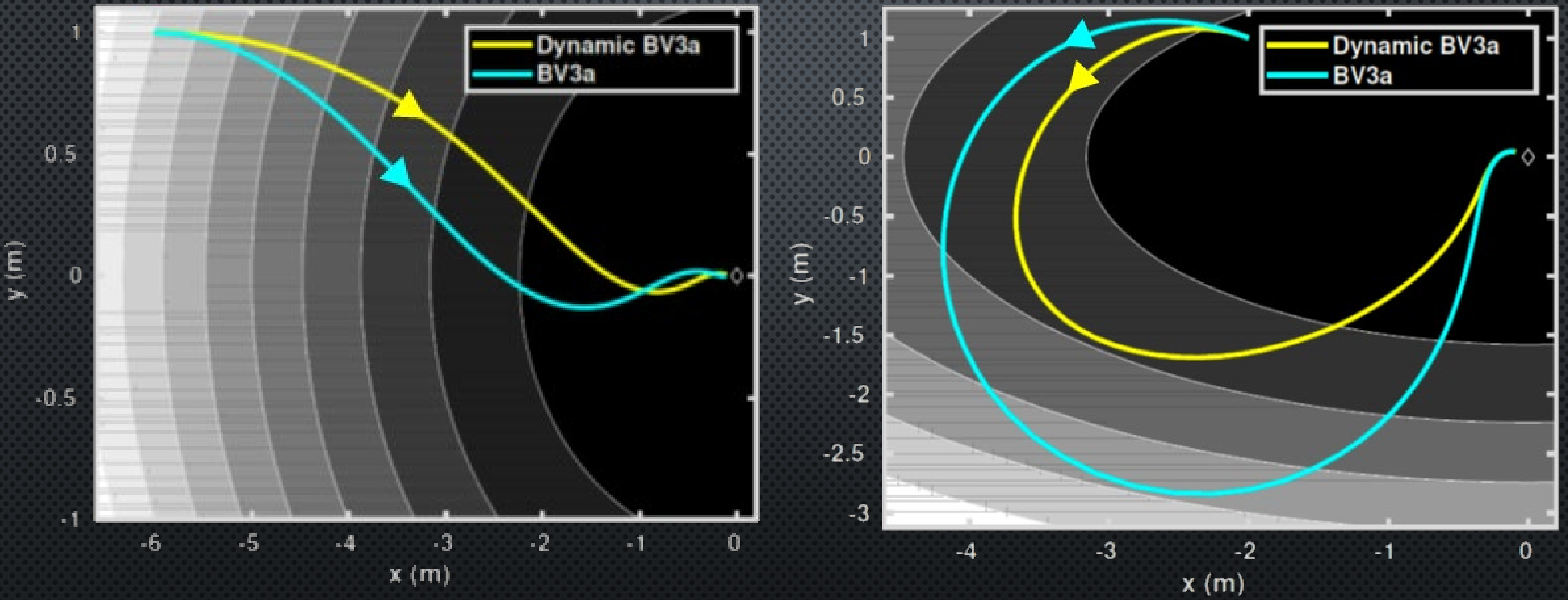
A Braitenberg model for odour localisation (static antennae)



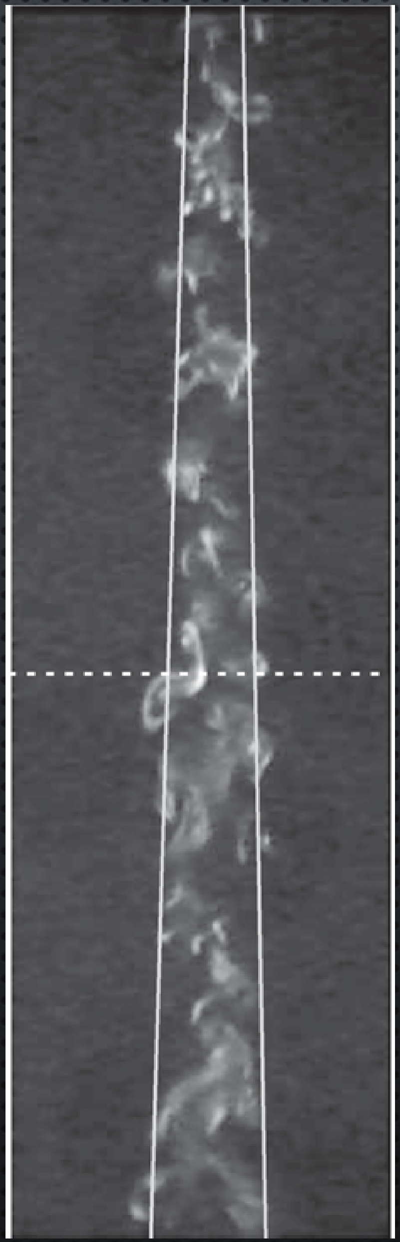
Left wheel velocity $v_l = f(s_l, \frac{\partial s_r}{\partial t})$

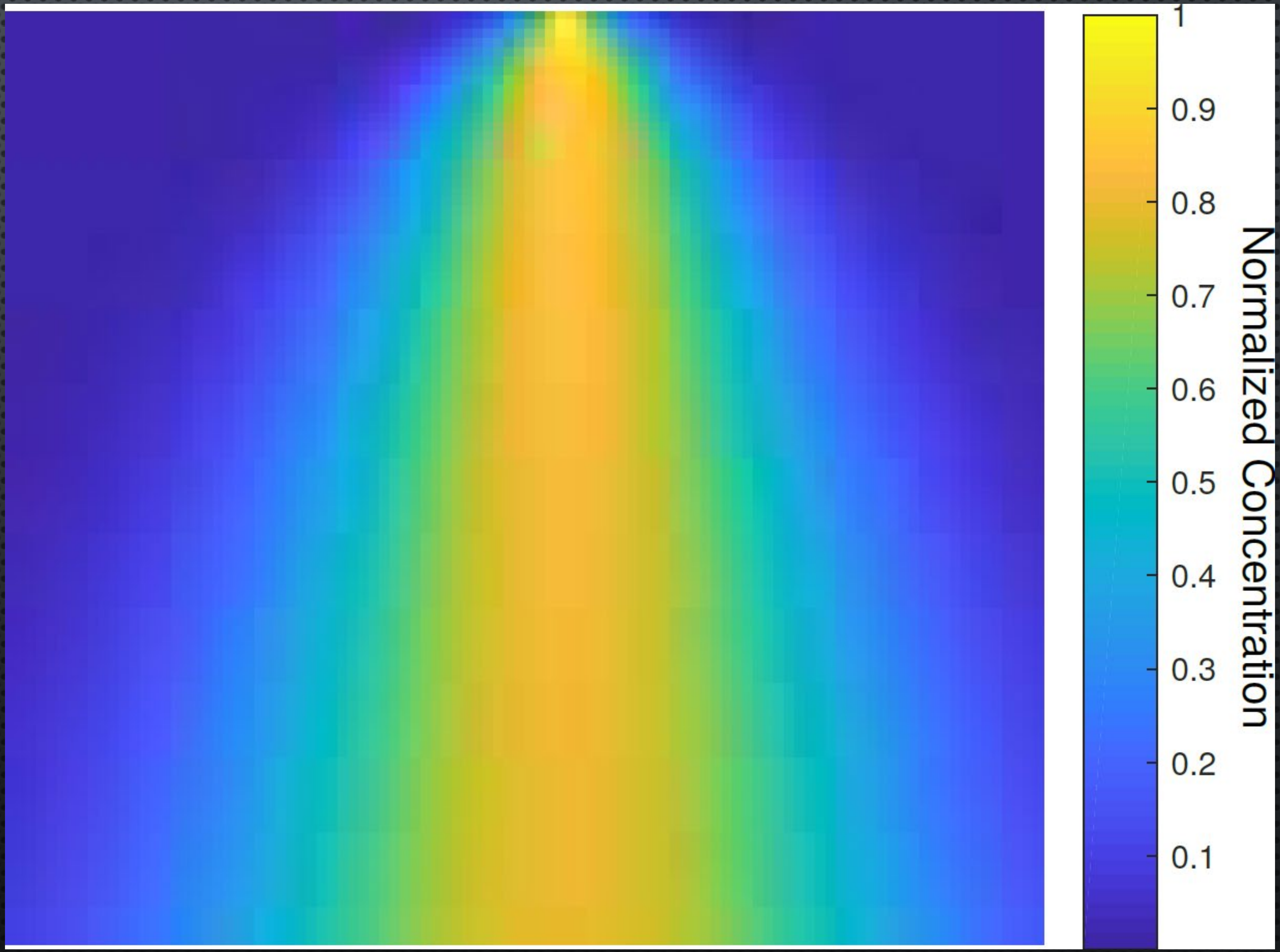
Right wheel velocity $v_r = f(s_r, \frac{\partial s_l}{\partial t})$

Taxis with and without stimulus dynamics (static antennae)



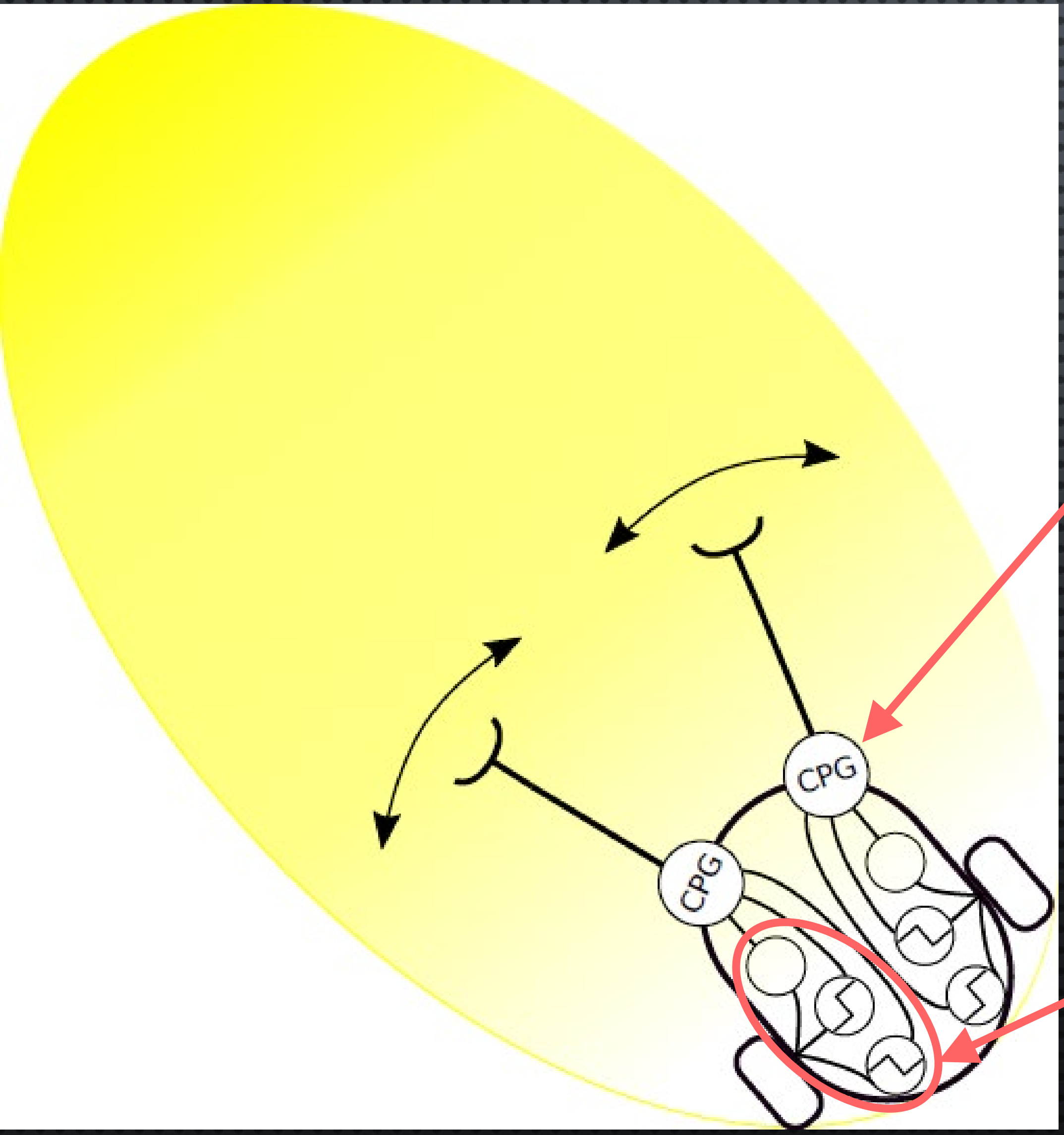
Task environment





Willis, 2008

A Braitenberg model for odour localisation (moving antennae)



$$\dot{u} = (\mu^2 - r) \cdot x + \omega y$$
$$\dot{v} = (\mu^2 - r) \cdot y + \omega x$$
$$r = (x^2 + y^2)$$

Antenna CPG parameters:

μ = steady state amplitude

$\omega = 2\pi f$ = angular velocity

$c_{l/r}$ = instantaneous odour concentration

$s_{l/r}$ = total stimulus

$v_{l/r}$ = wheel velocity

$f(\dot{c}_{r/l})$ = Heaviside step function (ON neuron model)

$f(-\dot{c}_{r/l})$ = Heaviside step function (OFF neuron model)

$$c_{l/r} = C\,(x_{l/r}\,, y_{l/r})$$
$$s_{l/r} = K_n c_{l/r} + K_d f(-\dot{c}_{r/l}) + K_i f(\dot{c}_{r/l})$$
$$v_{l/r} = \alpha(s_M - s_{l/r})$$

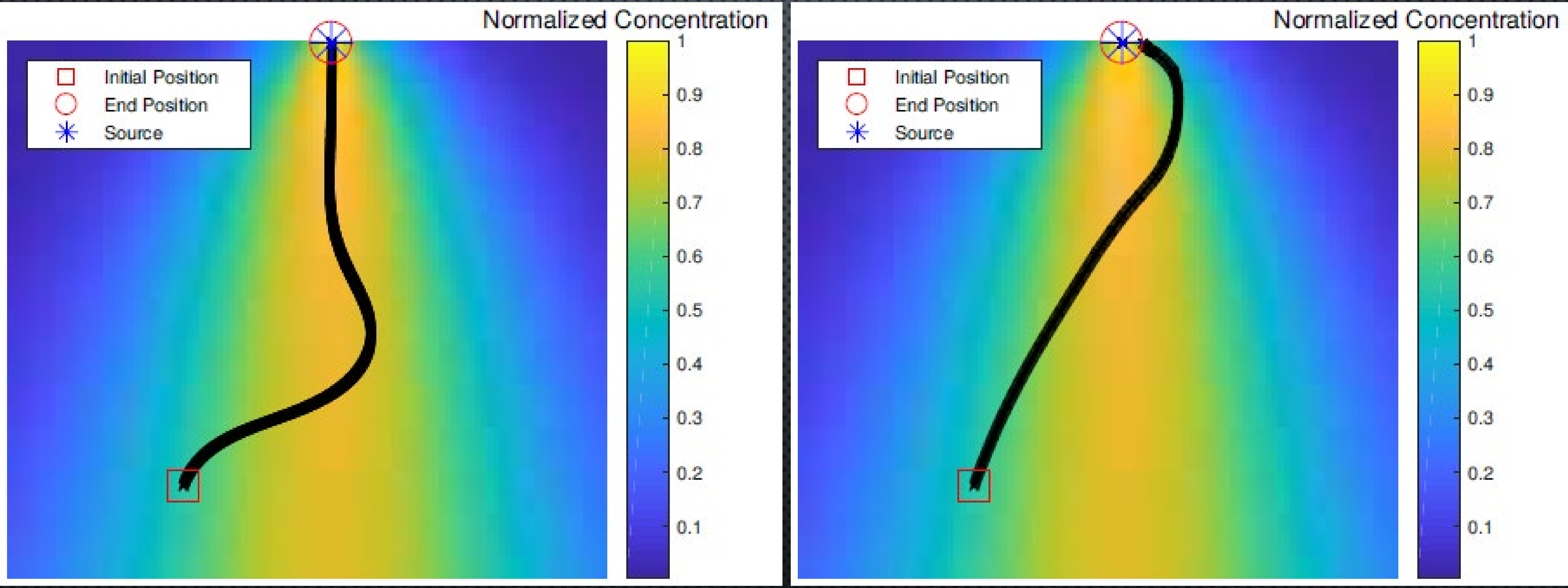
Sensorimotor coupling parameters:

α = gain term

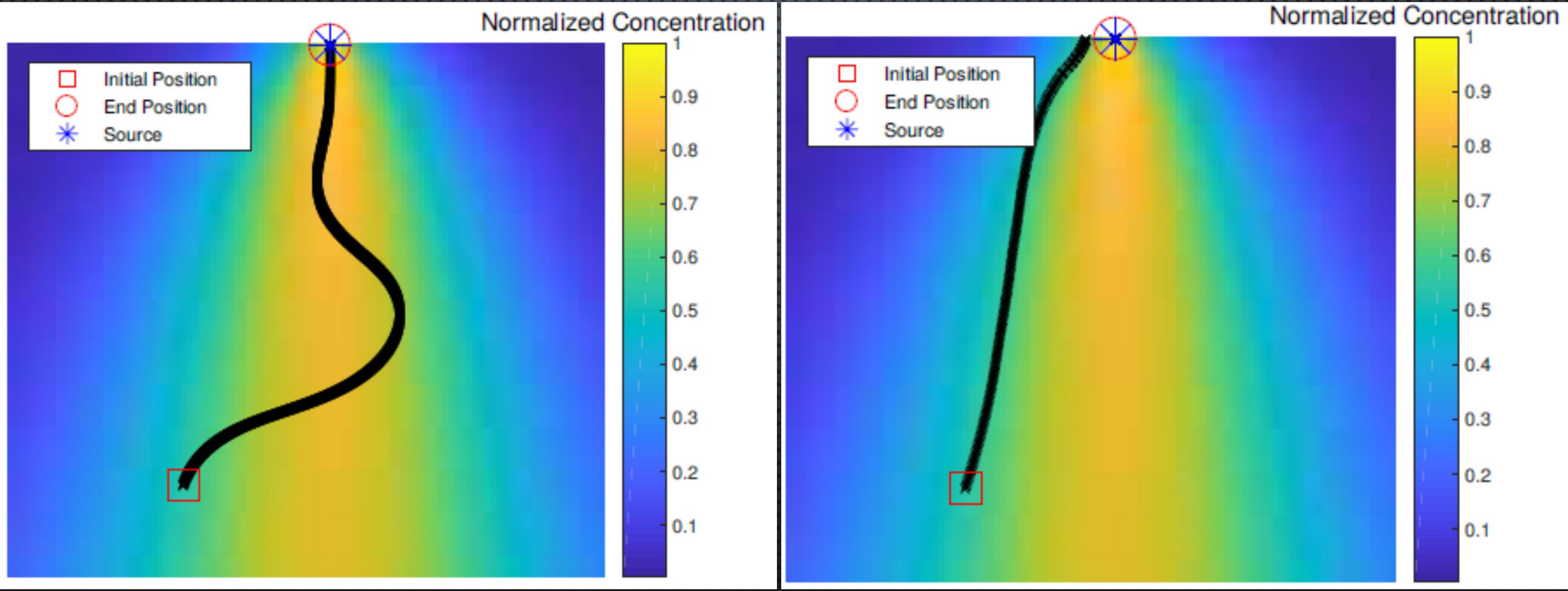
s_M = maximum stimulus

K_n, K_d, K_i = gain terms

Localising odour source (static antennae)

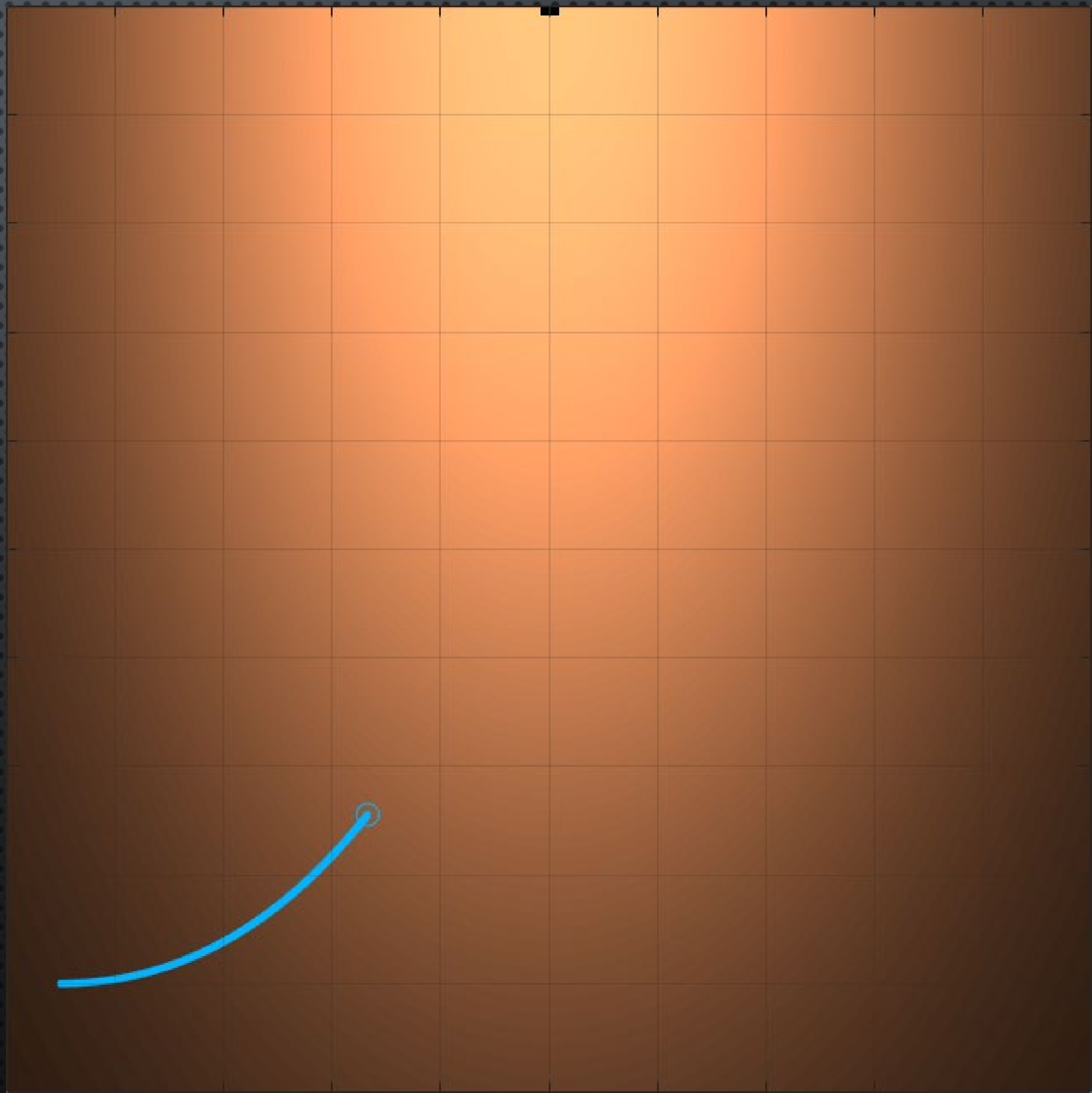


Localising odour source (moving antennae)



Matlab exercises

- Download "braitenberg_gradient.m"
- Implement various light following Braitenberg vehicles and observe their behaviour
- Vary parameters and observe behaviour
  - *sensor_width*
  - *add_noise* and *snr_db*
  - *robot_pose*

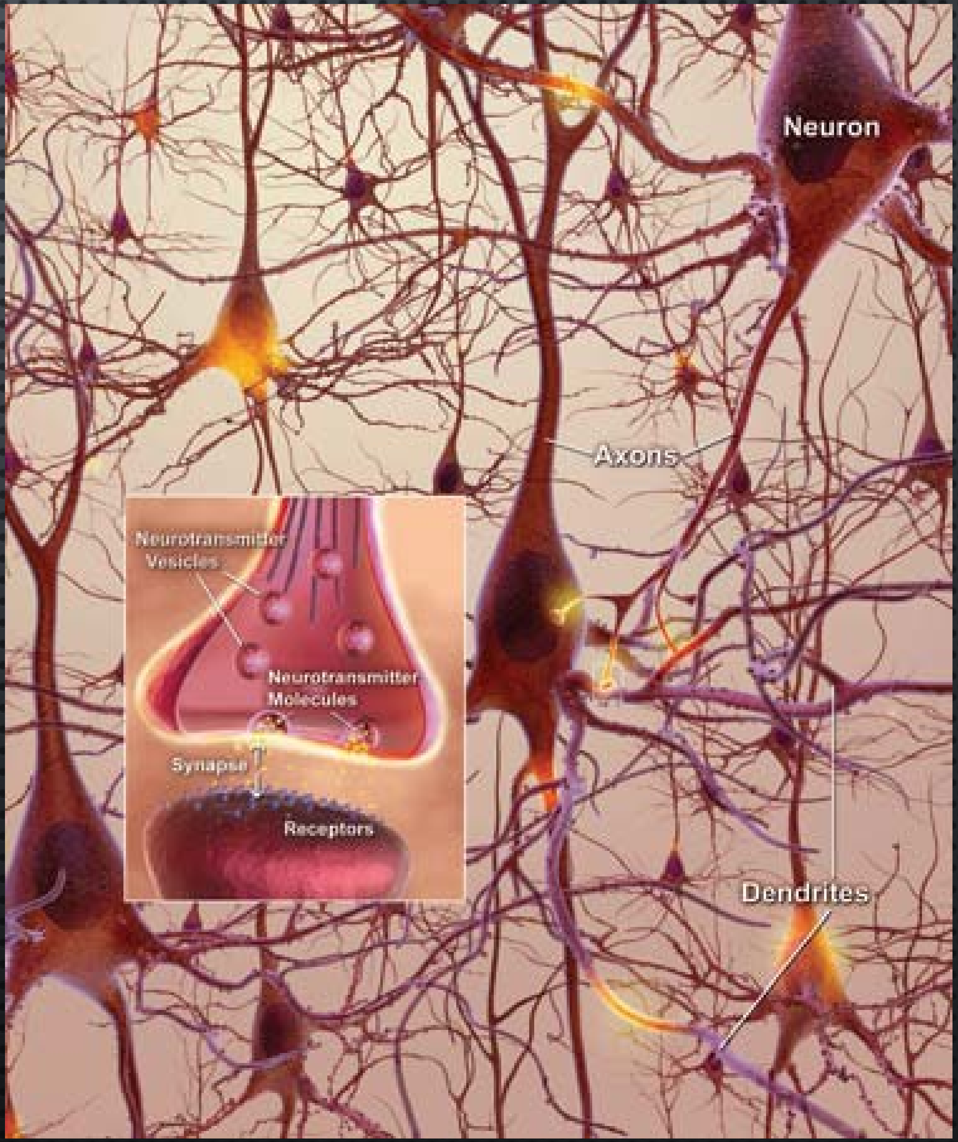


4 Learning mechanisms in biological neurons and their functional engineering models

Neurobiology ahead
(but now you love it ☻)

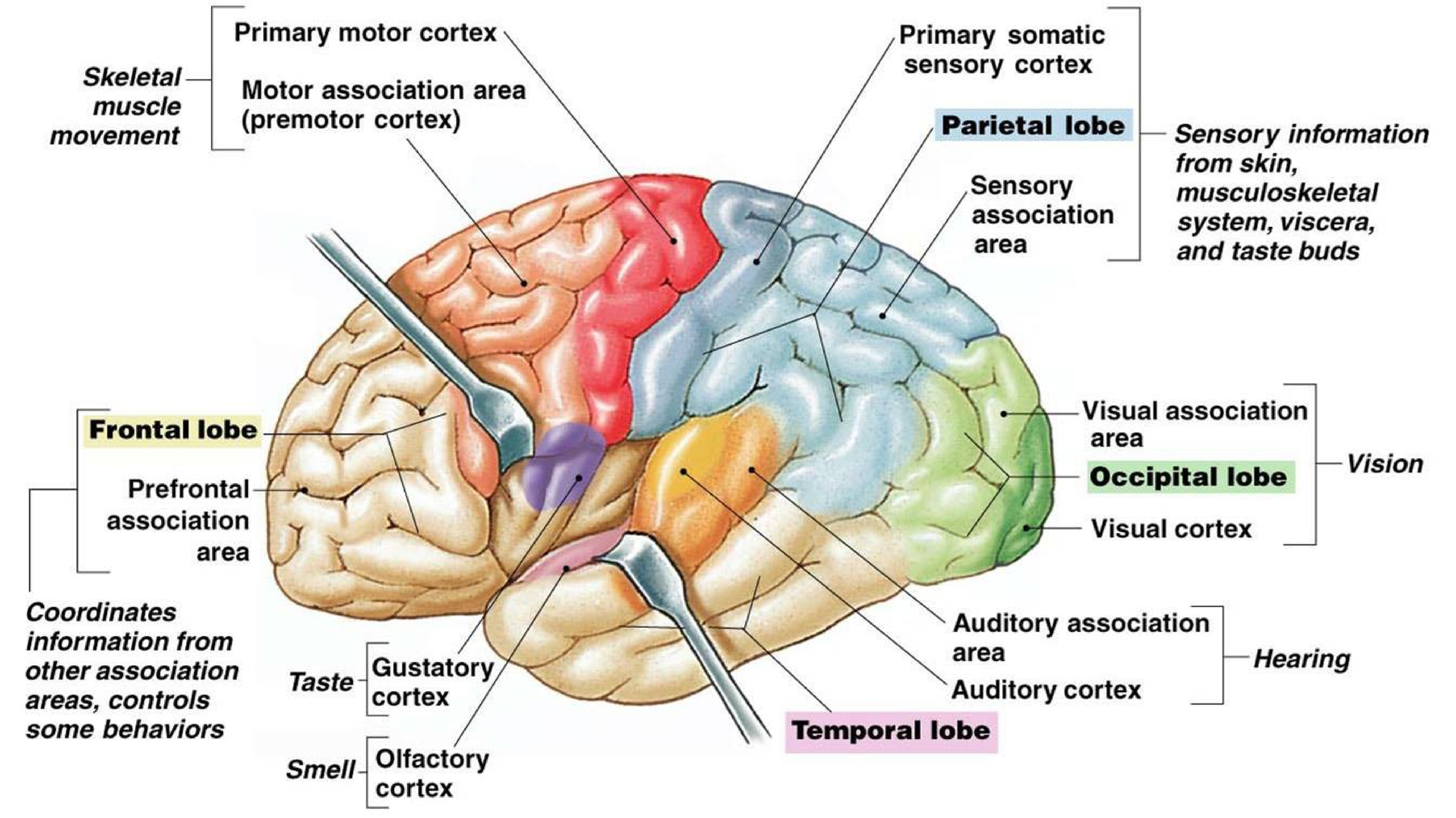
Neuroplasticity: a mechanism for biological learning

- Neuroplasticity is the ability of the brain to reorganise itself in both structure and function over time due to external and internal events
- Reorganisation can occur throughout the neural hierarchy – from single neurons to entire cortical areas
- Neuroplasticity is a continuous process that occurs at multiple timescales from milliseconds to days
- It is the underlying mechanism behind learning and memory formation



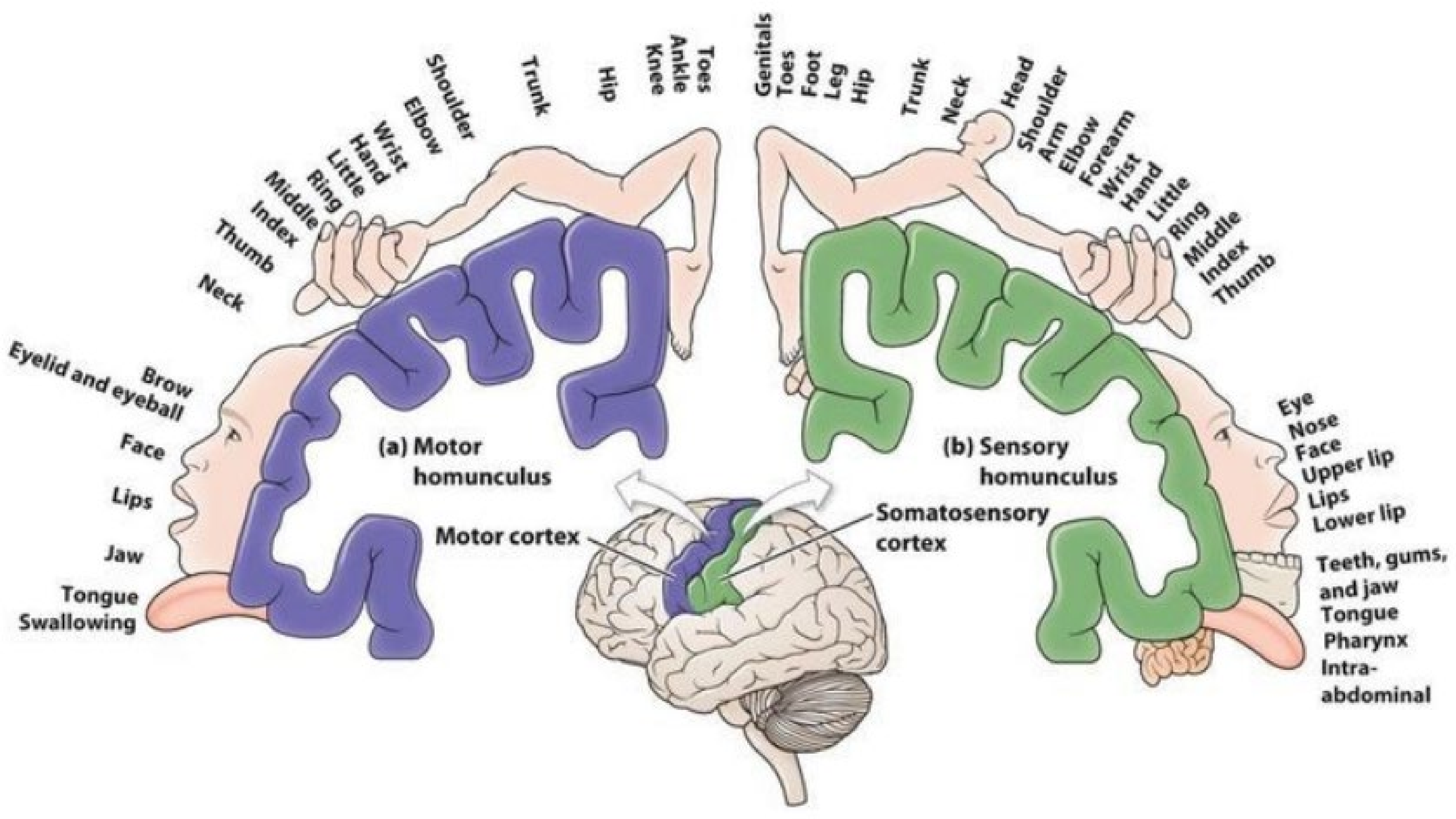
Encyclopedia Britannica

Brain areas

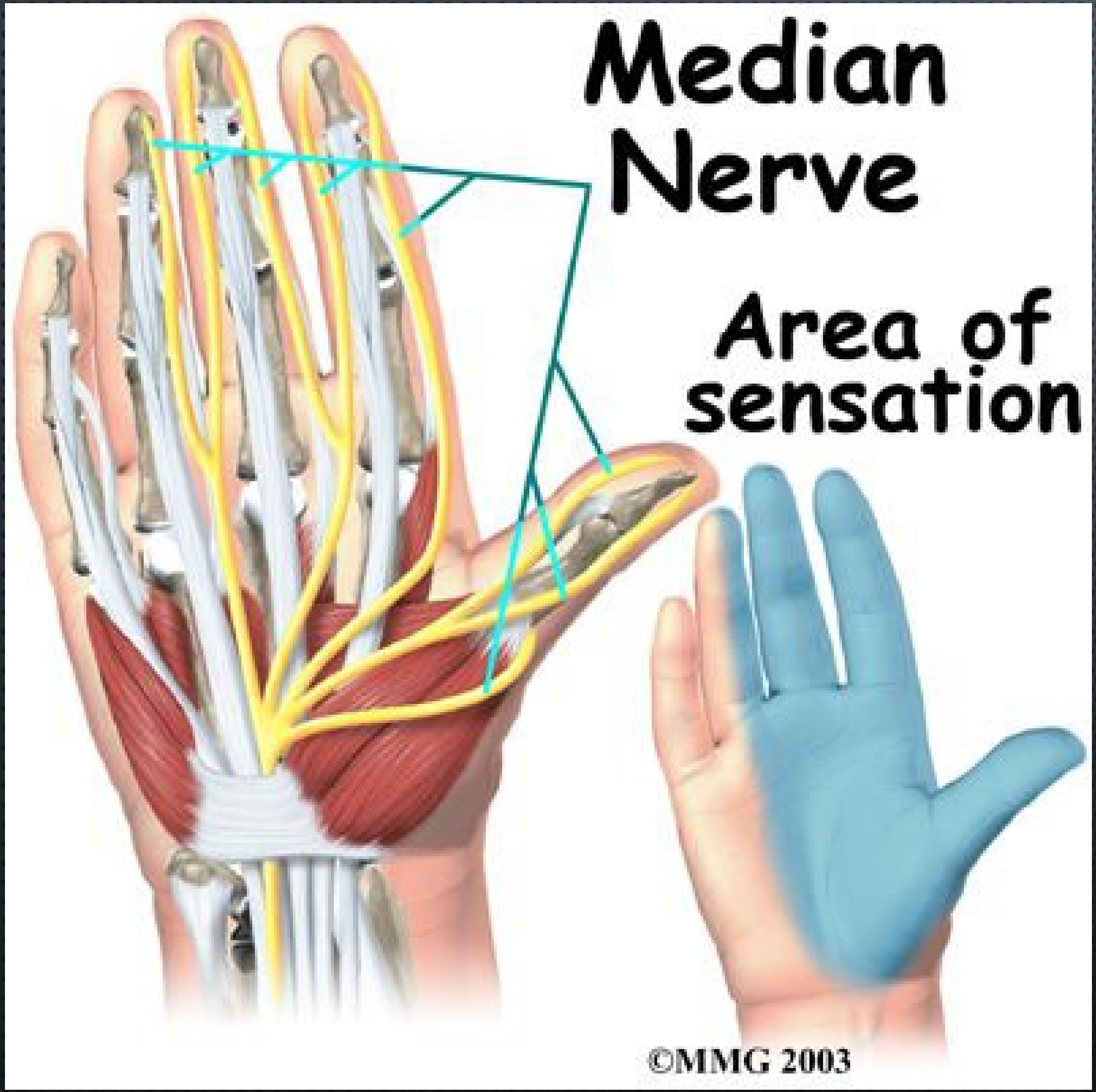


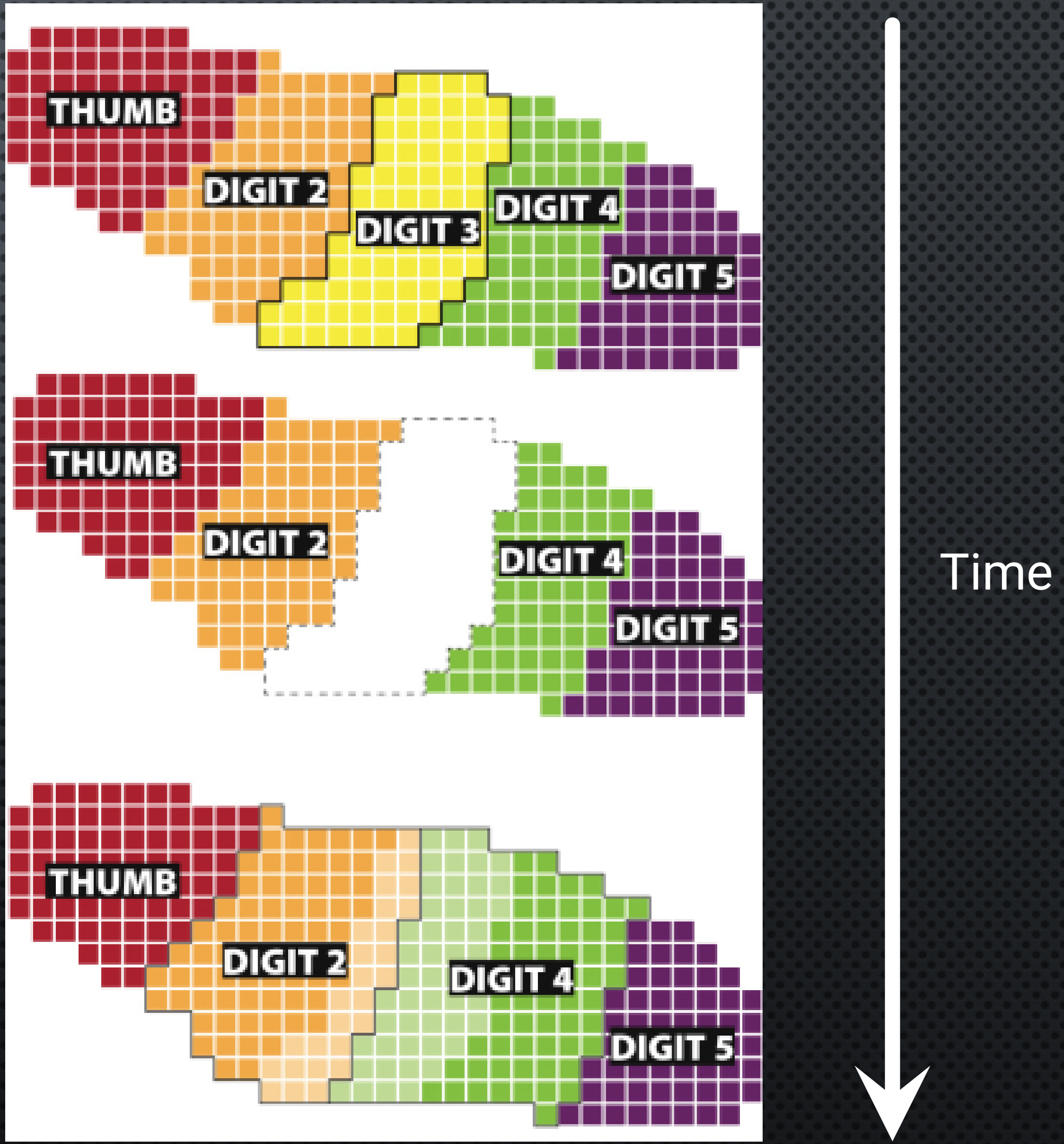
Skeletal muscle movement
Primary motor cortex
Motor association area (premotor cortex)
Primary somatic sensory cortex
Parietal lobe
Sensory association area
Sensory information from skin, musculoskeletal system, viscera, and taste buds
Frontal lobe
Prefrontal association area
Coordinates information from other association areas, controls some behaviors
Visual association area
Occipital lobe
Visual cortex
Vision
Taste
Gustatory cortex
Smell
Olfactory cortex
Auditory association area
Auditory cortex
Hearing
Temporal lobe

Neuroplasticity at the cortical level



Neuroplasticity at the cortical level





Types of neuroplasticity

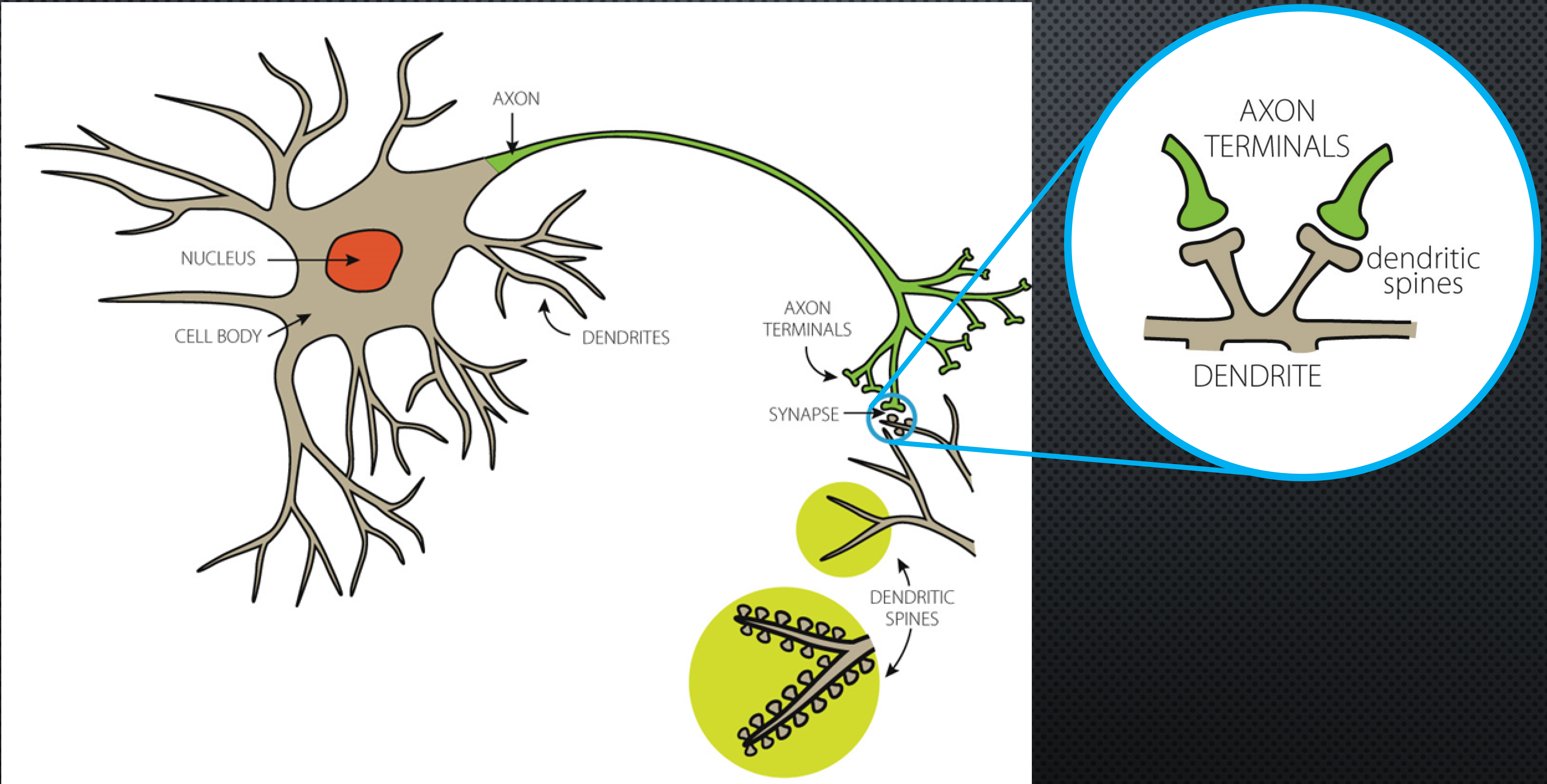
Structural plasticity

- mechanism describing the generation of new neural connections
- long-term changes
- Biological mechanisms:
  - Synapses — changes in number
  - Synaptic receptors — changes in density
  - Dendritic arborization (complexity of the dendritic tree)
  - Dendritic spines — changes in density
  - Axonal arborization (sprouting)
  - Glial and neuron interactions
  - Vascular processes and angiogenesis (new blood vessel growth)
  - Cell proliferation (including neurogenesis)

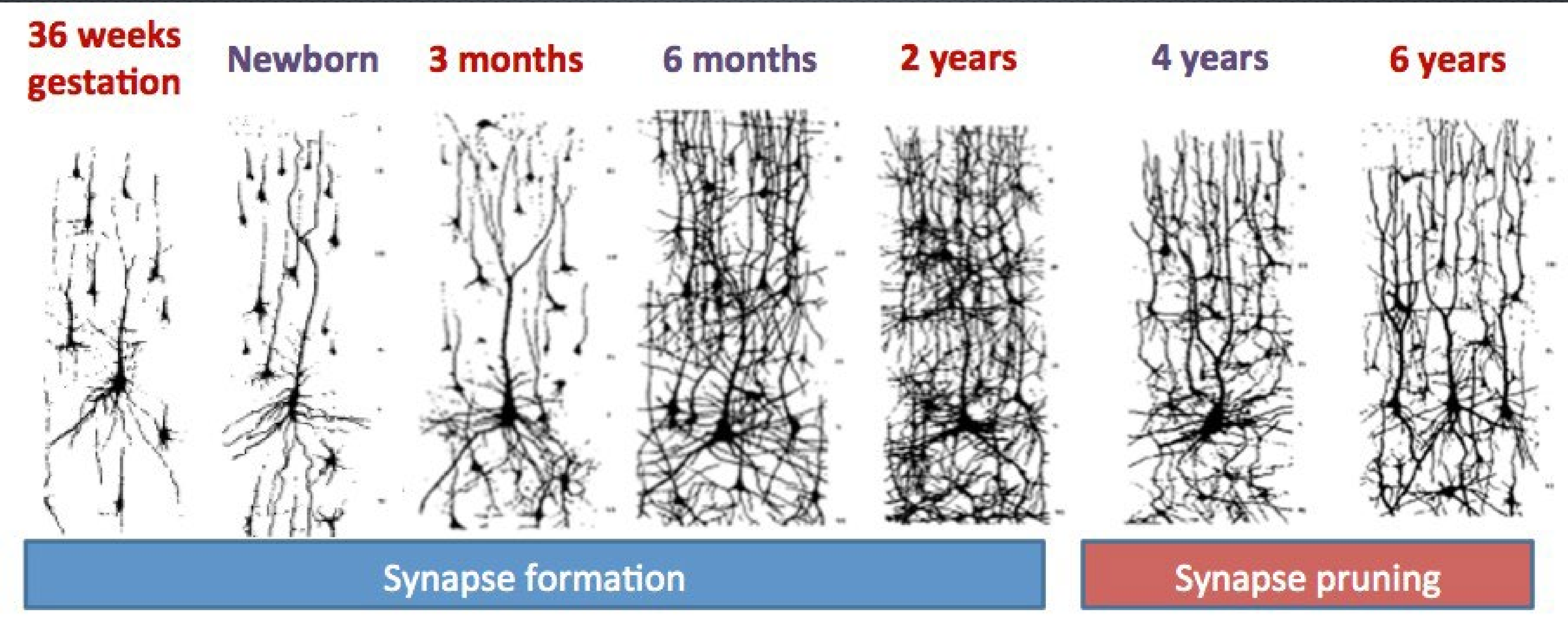
Functional plasticity

- mechanism of changing the strength of existing neural connections
- short term changes
- Biological mechanisms:
  - Synapses — changes in strength. Widely known as synaptic plasticity

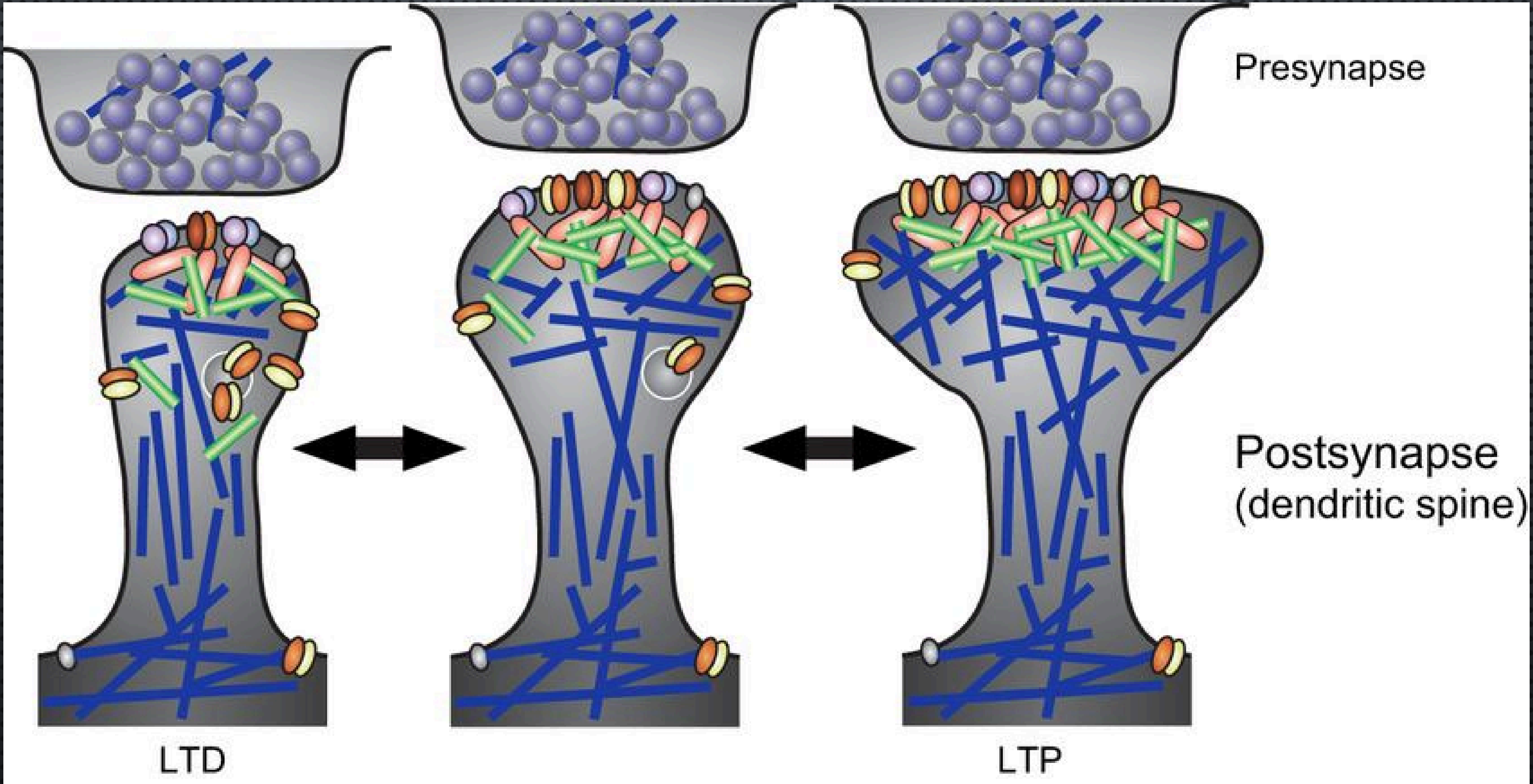
Recap: neuron structure



Structural plasticity: number of synapses

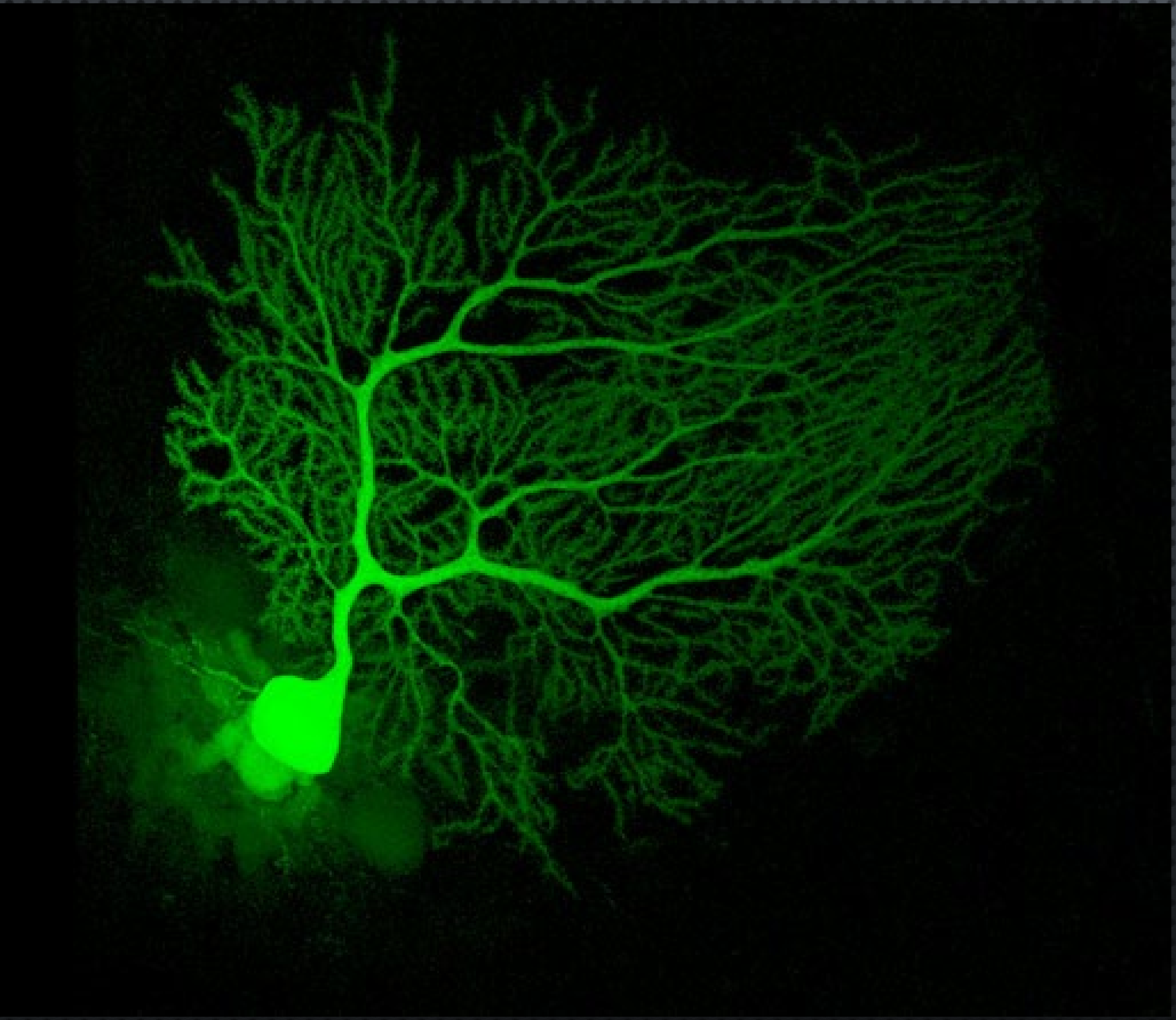


Structural plasticity: synaptic receptor density



Kessels and Qualmann, *Jour. Cell Sci.*, 2015

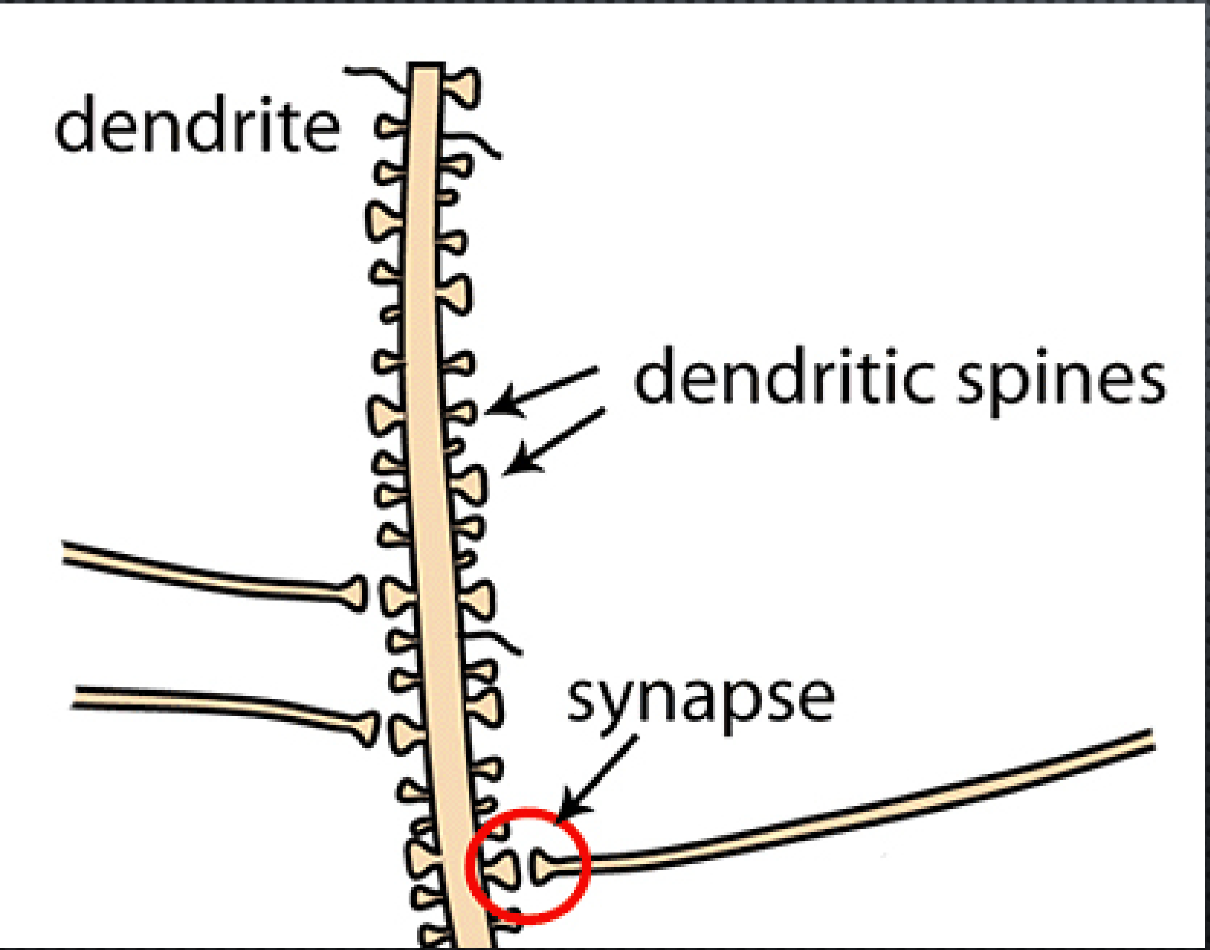
Structural plasticity: dendritic arborization

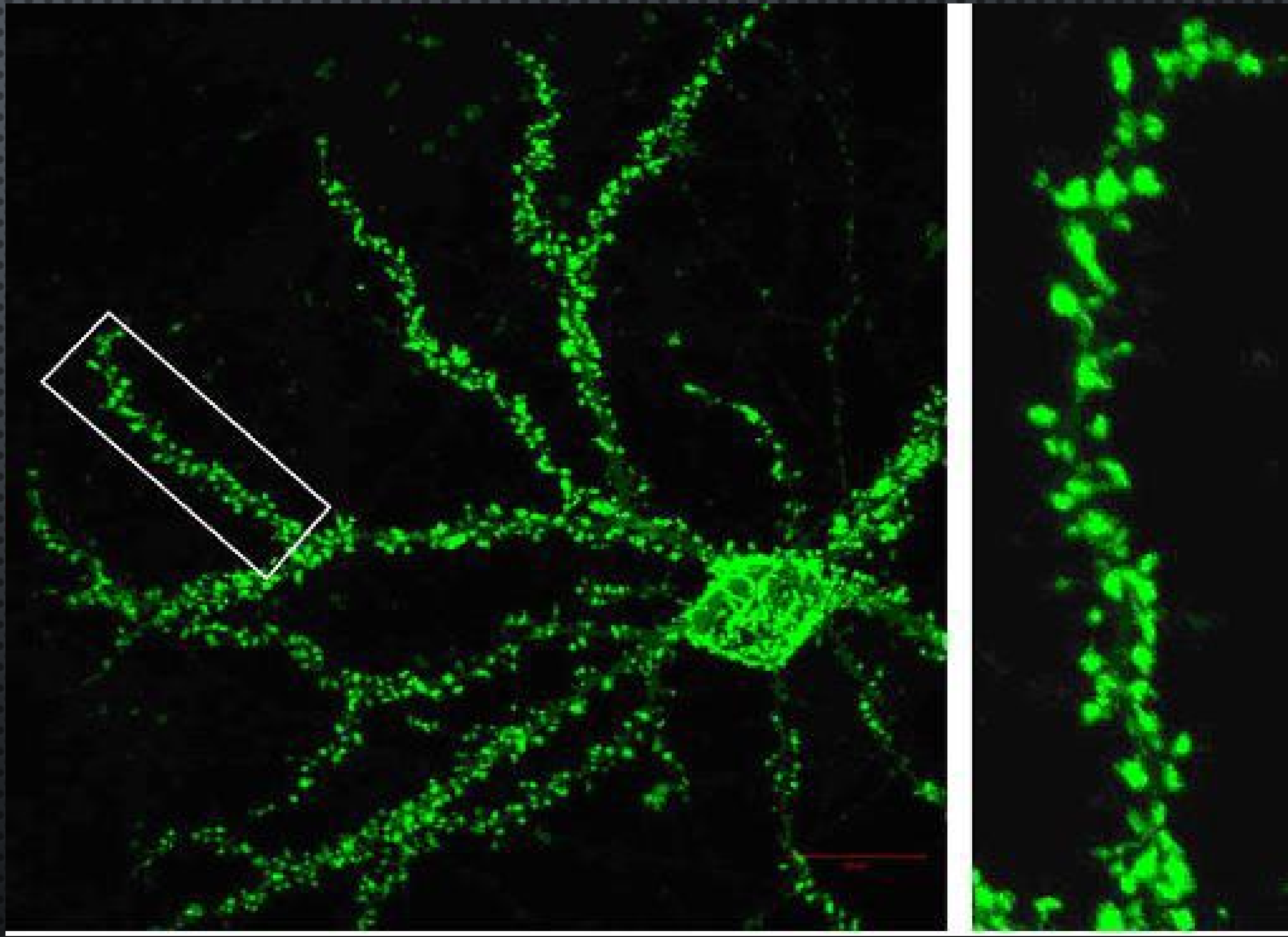


Purkinje neuron in the cerebellum

Hippocampal CA1 pyramidal neuron

Structural plasticity: dendritic spine density

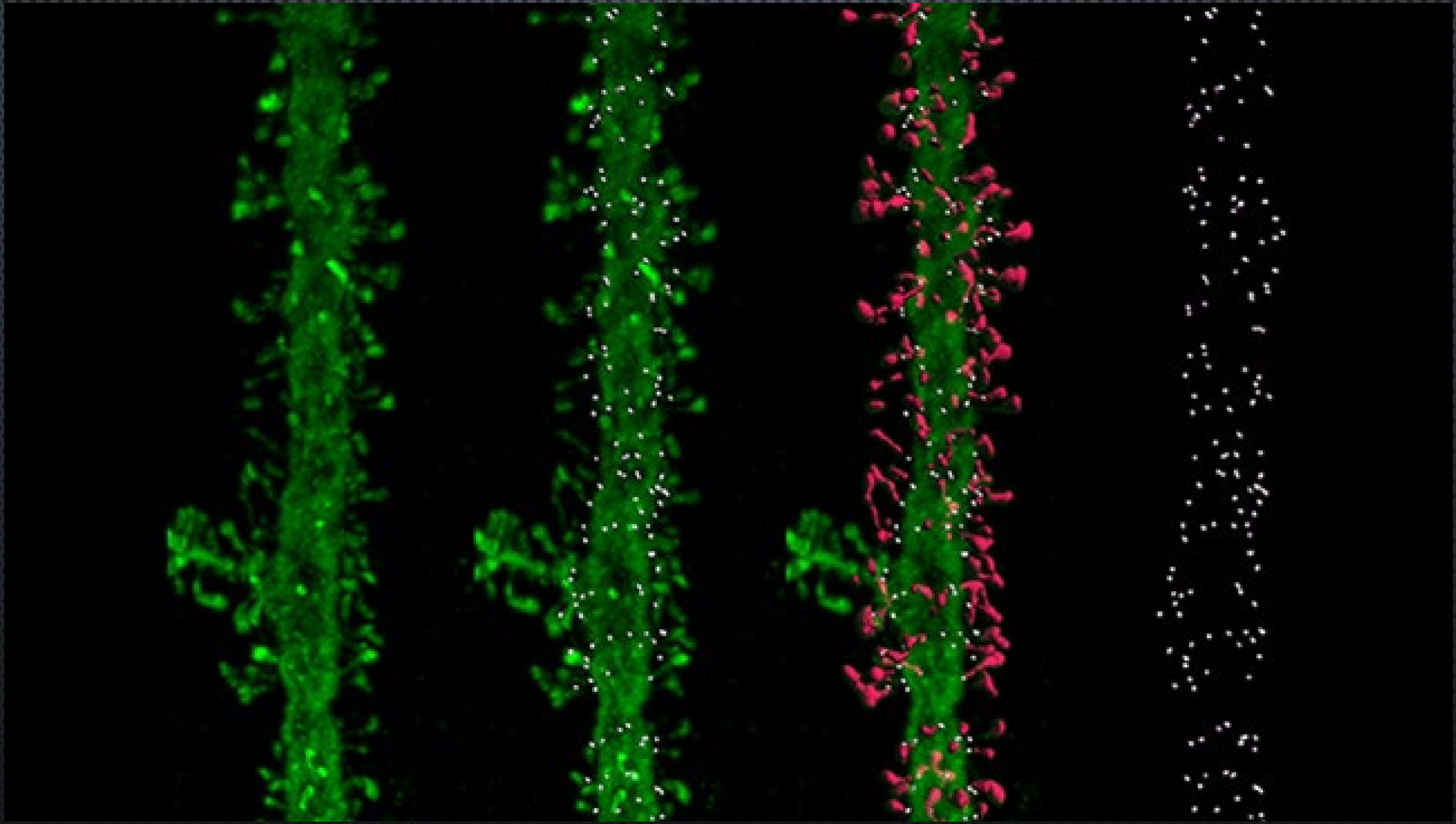




https://neuwritesd.org/2017/03/16/mind-the-gap-spaced-learning-and-dendritic-spines/

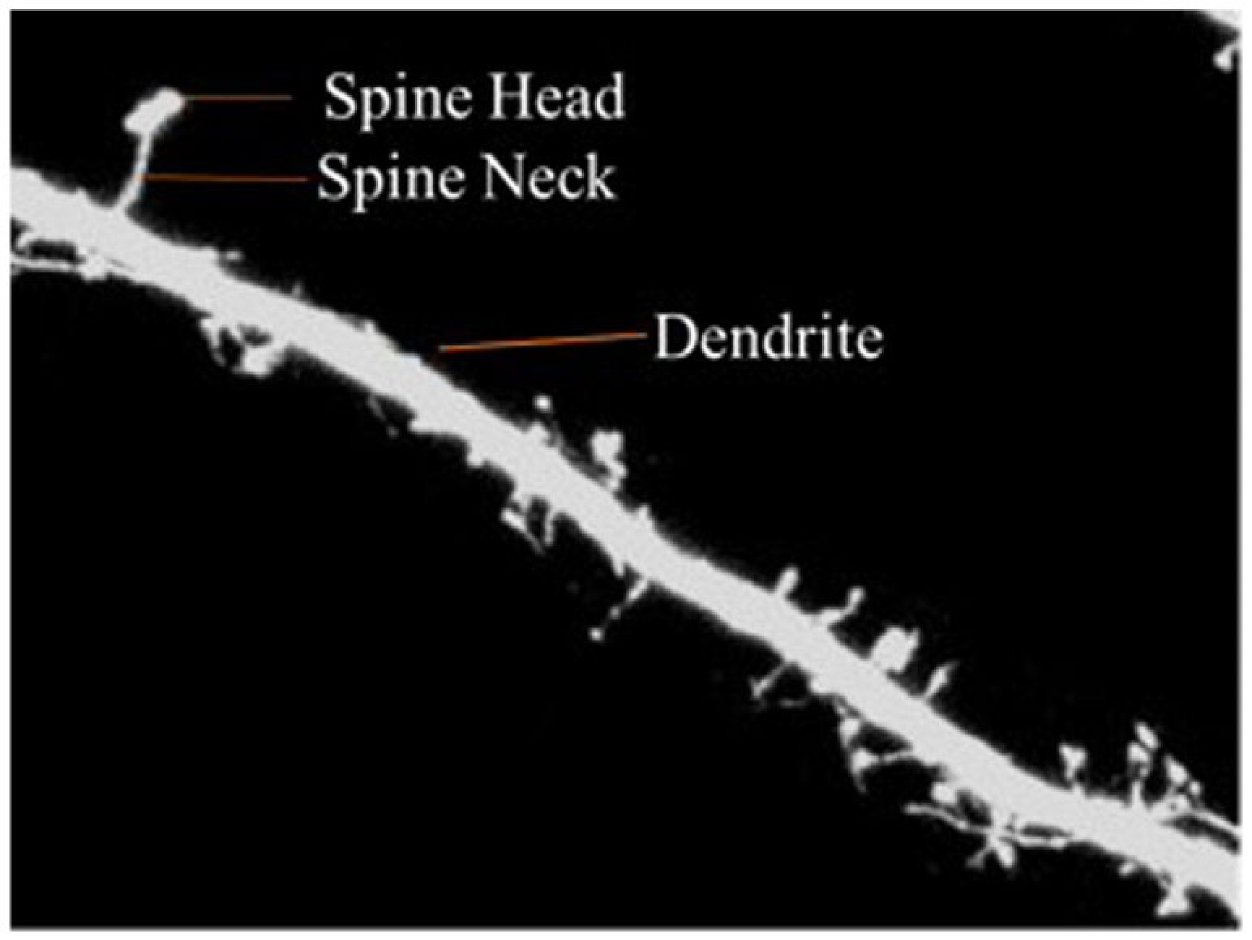
http://www.lacasamorett.com/foxgallery/dendritic-spines.html

Structural plasticity: dendritic spine density

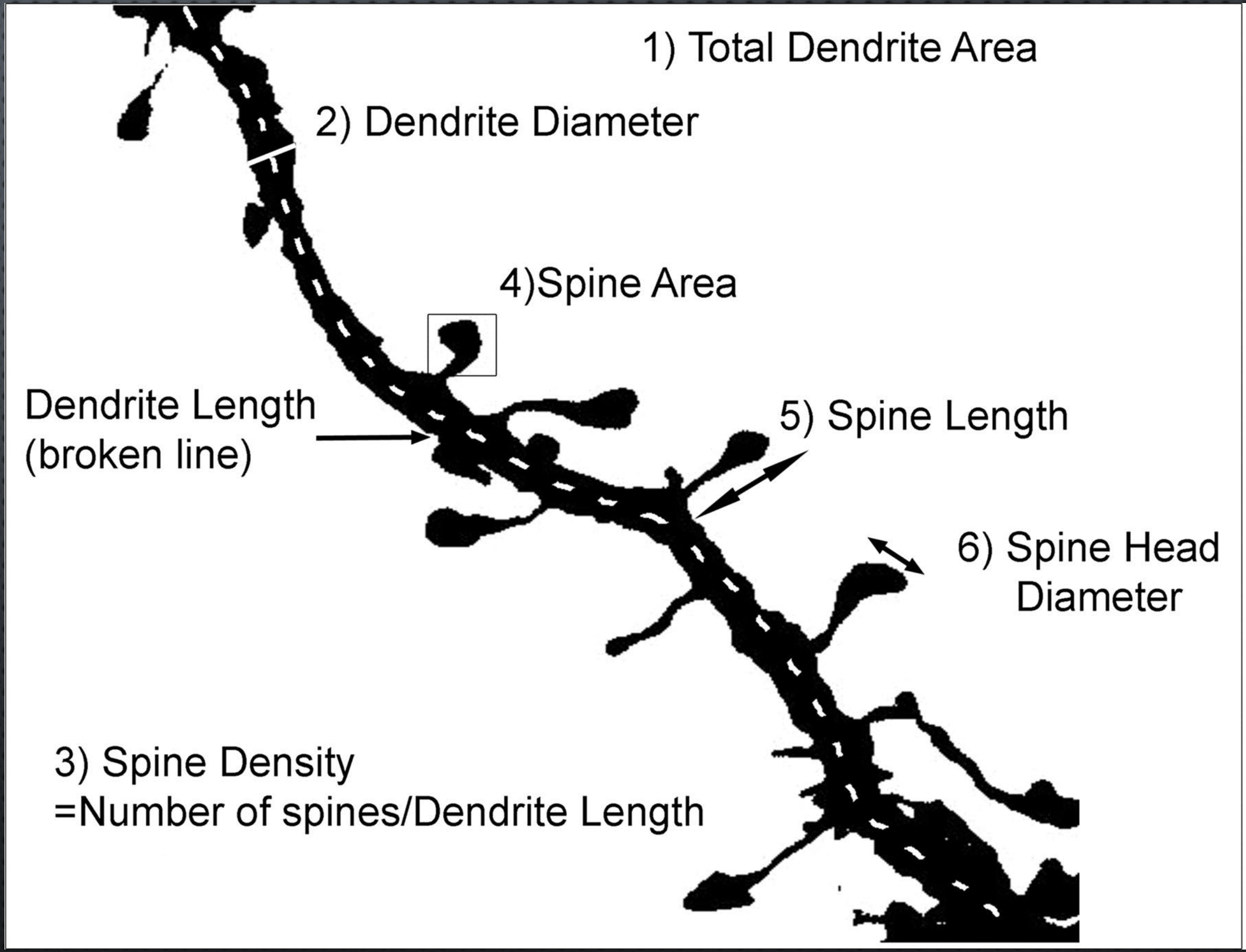


Morales et al., *Jour. Neuro.*, 2014

Structural plasticity: dendritic spine density

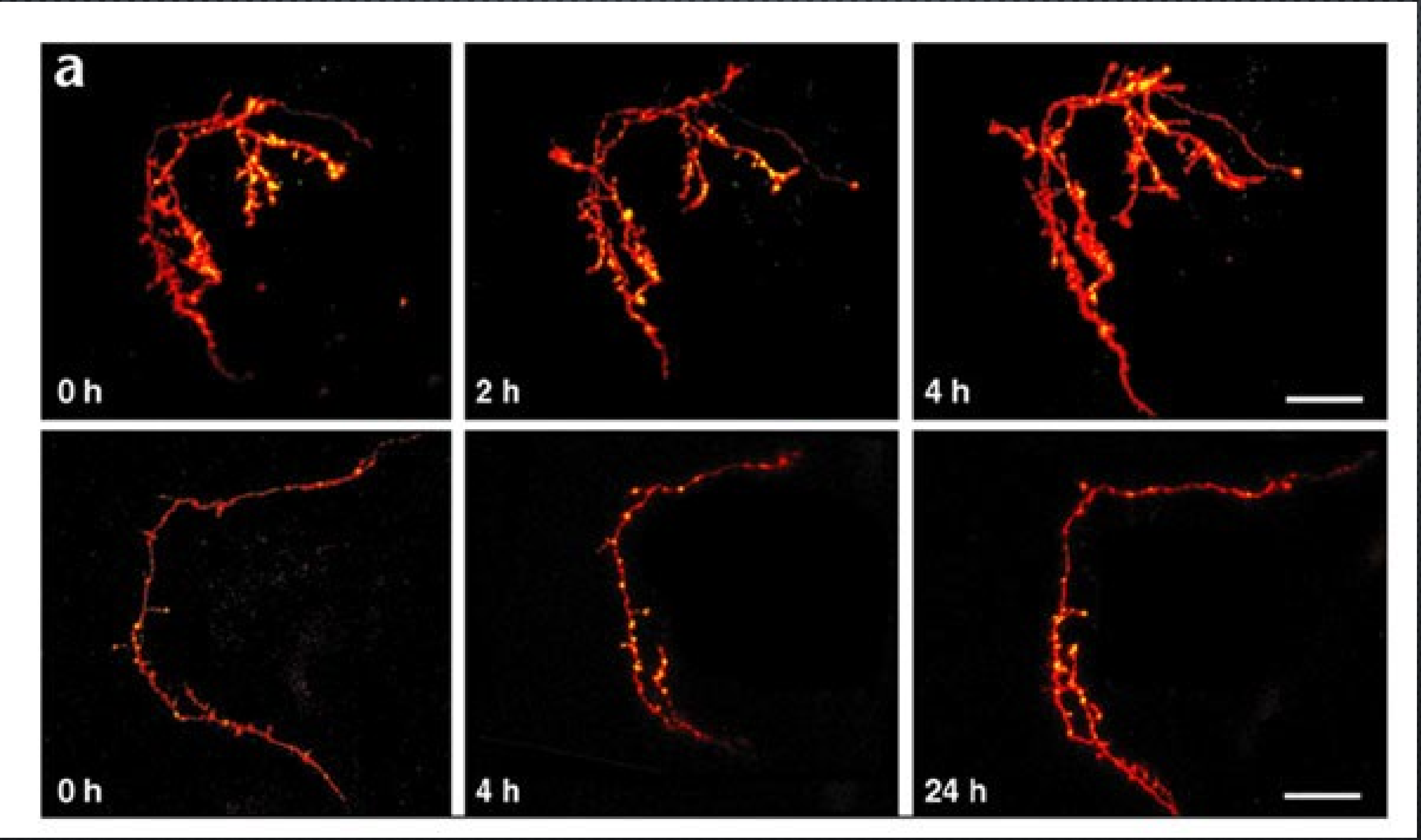


Wang et al., *PeerJ*, 2016



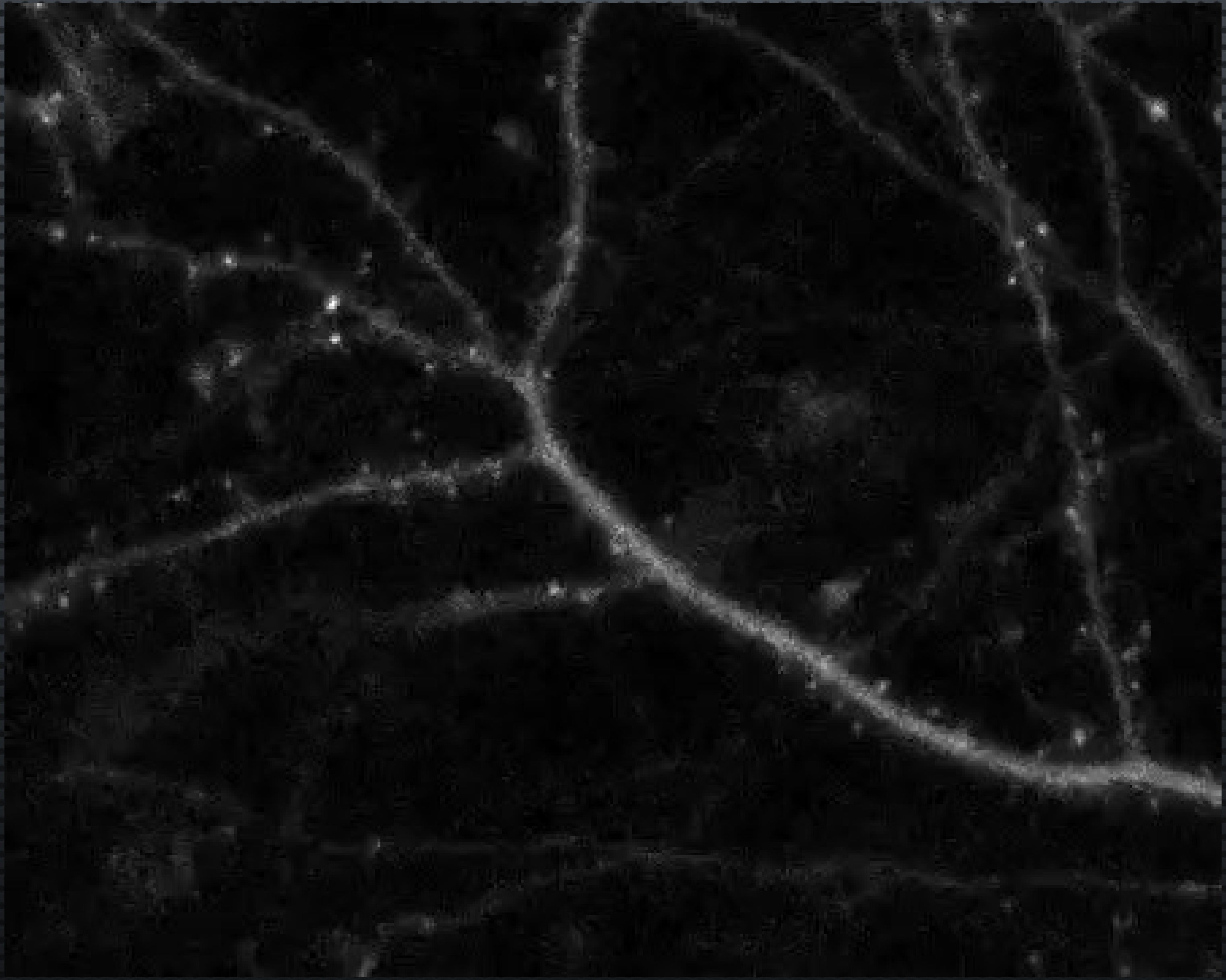
Smith et al., PNAS, 2009

Structural plasticity: axonal arborization



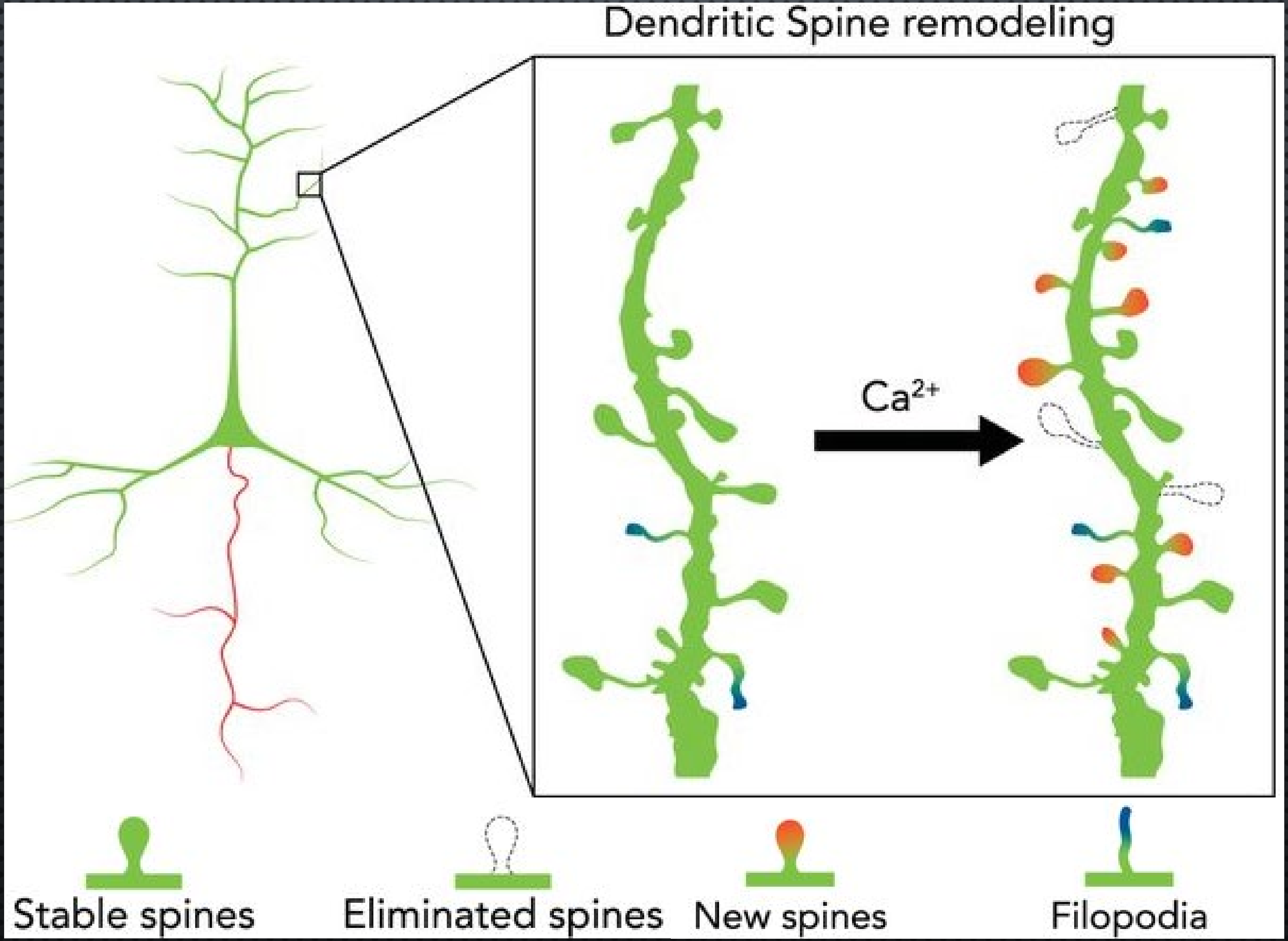
Alsina et al., *Nat. Neuro.*, 2001

Structural plasticity in action

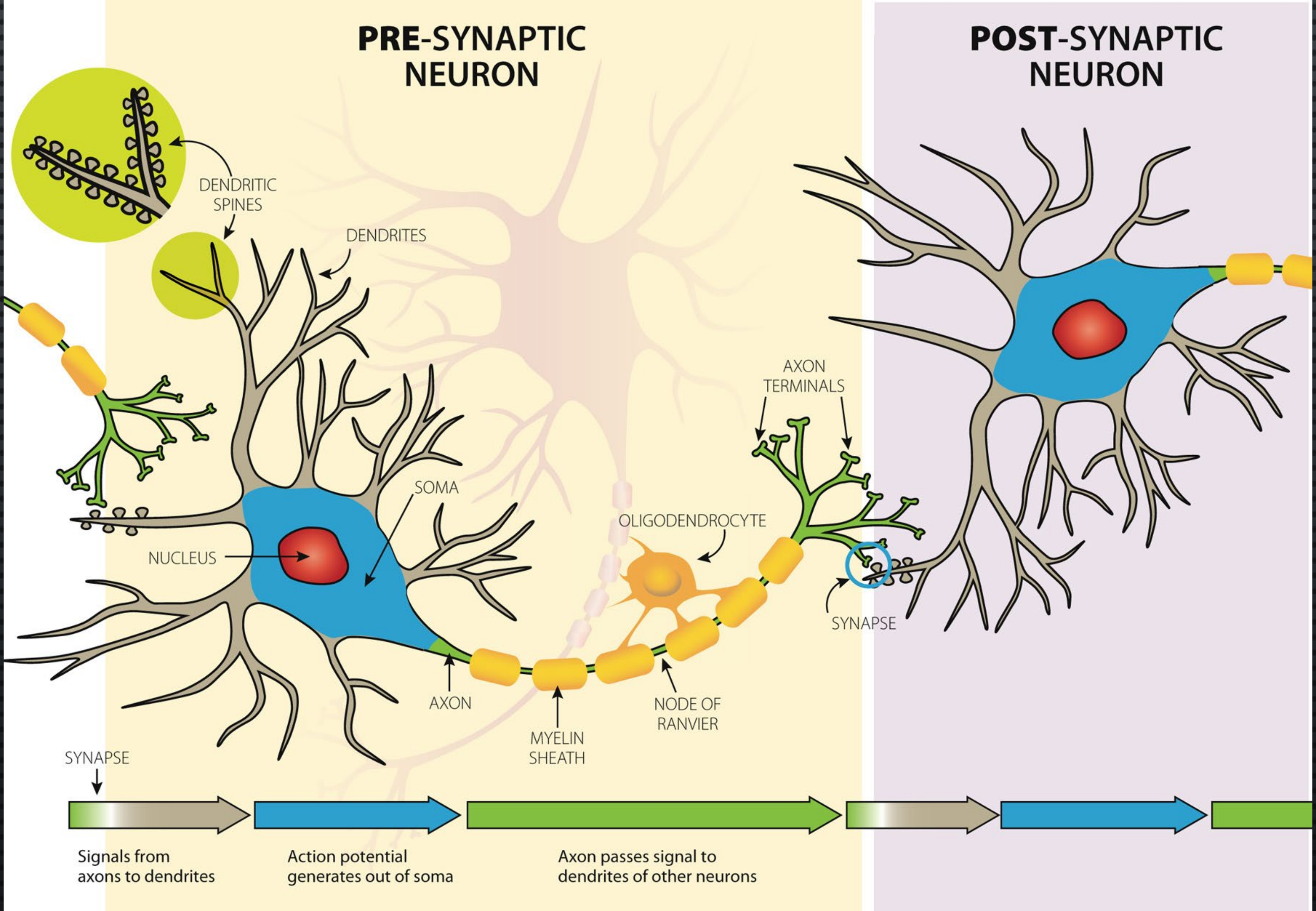


Minerbi et al., *PLOS Biol.*, 2009

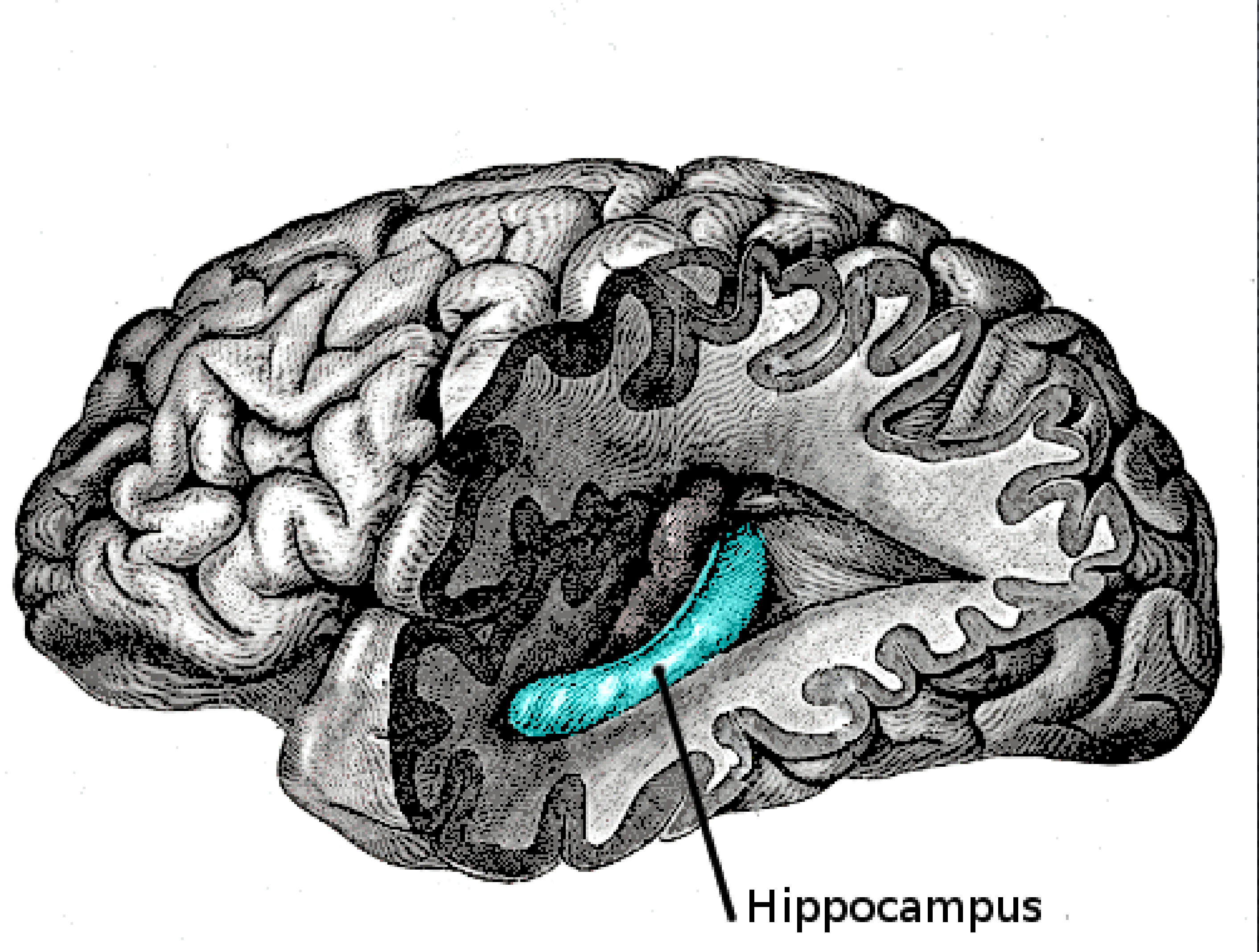
Structural plasticity in the hippocampus



Functional (synaptic) plasticity

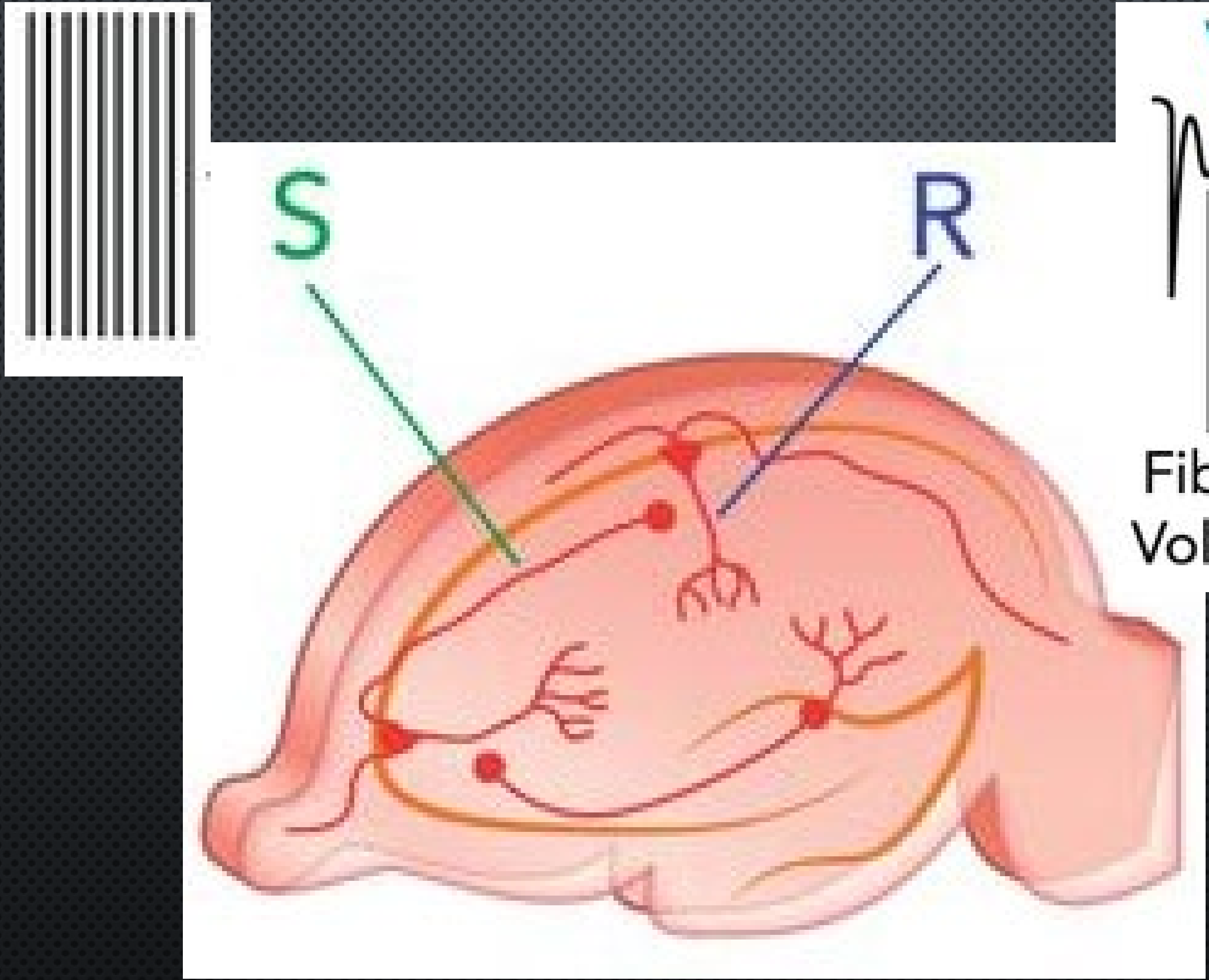


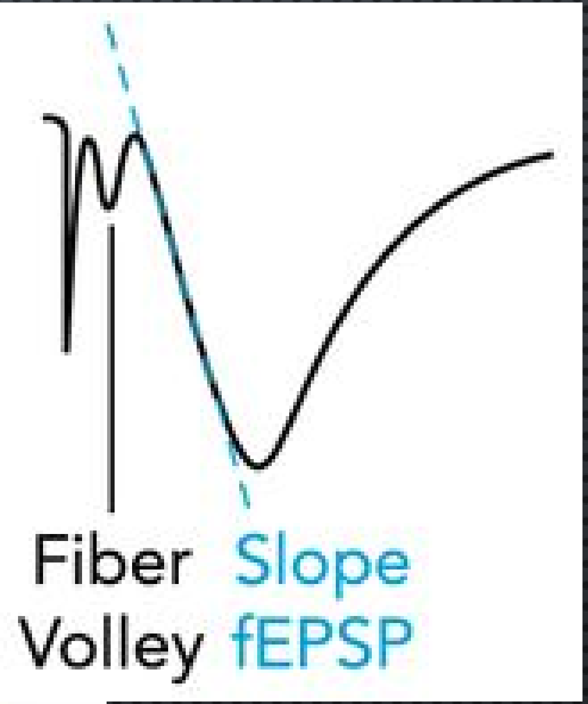
Synaptic plasticity in the hippocampus



http://www.fil.ion.ucl.ac.uk/memo/brain.html

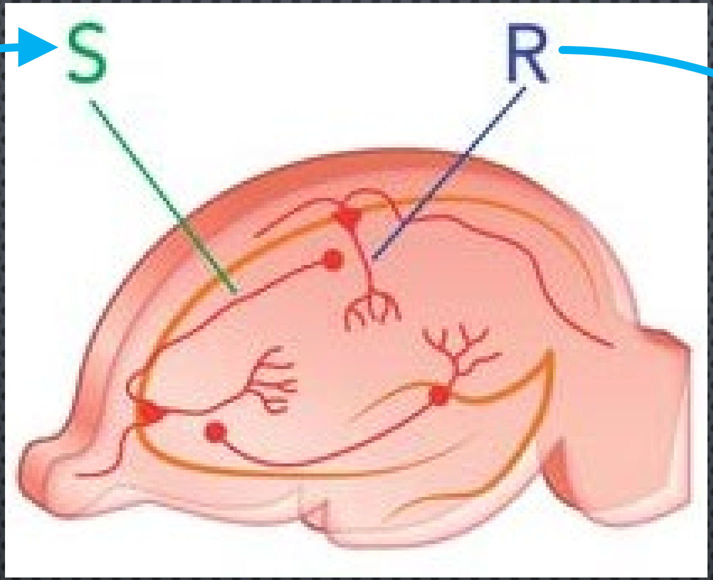
Mechanisms underlying synaptic plasticity

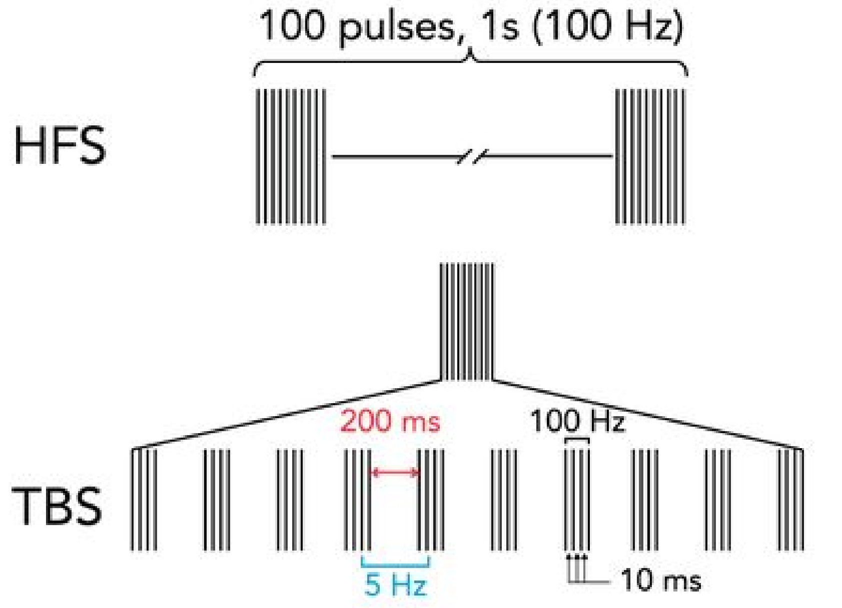


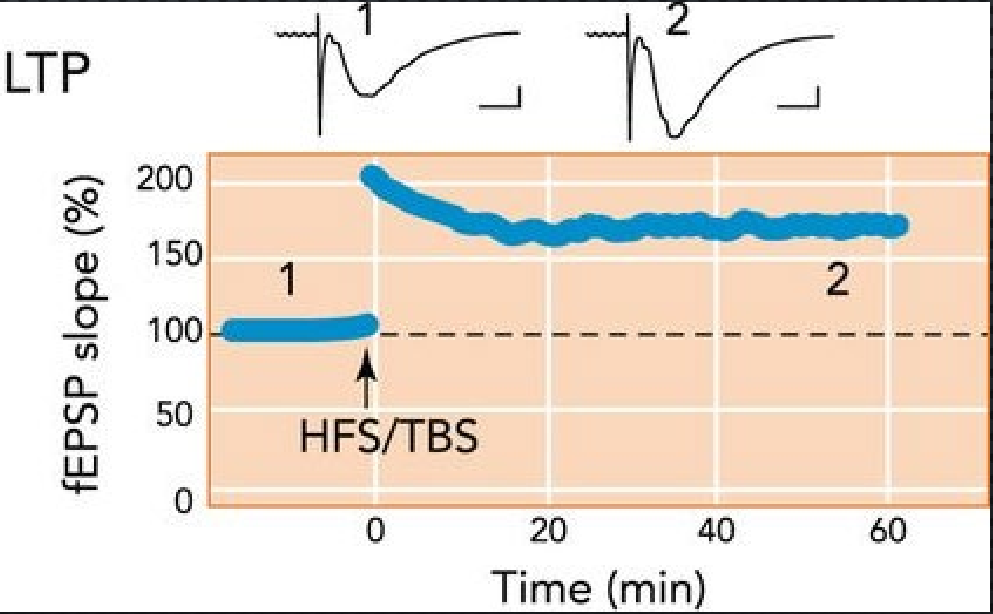


Hidalgo and Arias-Cavieres, *Physiology*, 2016

Long Term Potentiation (LTP)

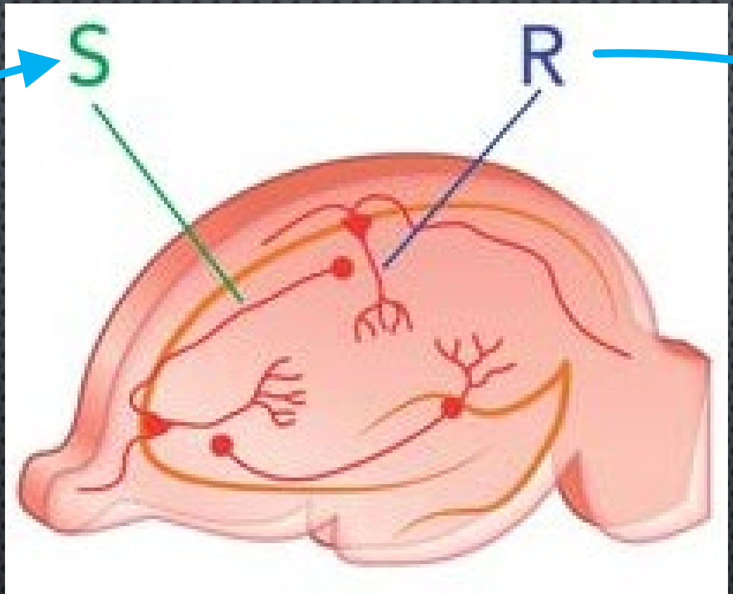


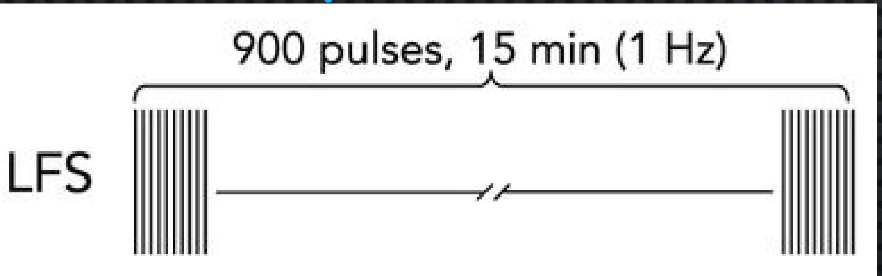


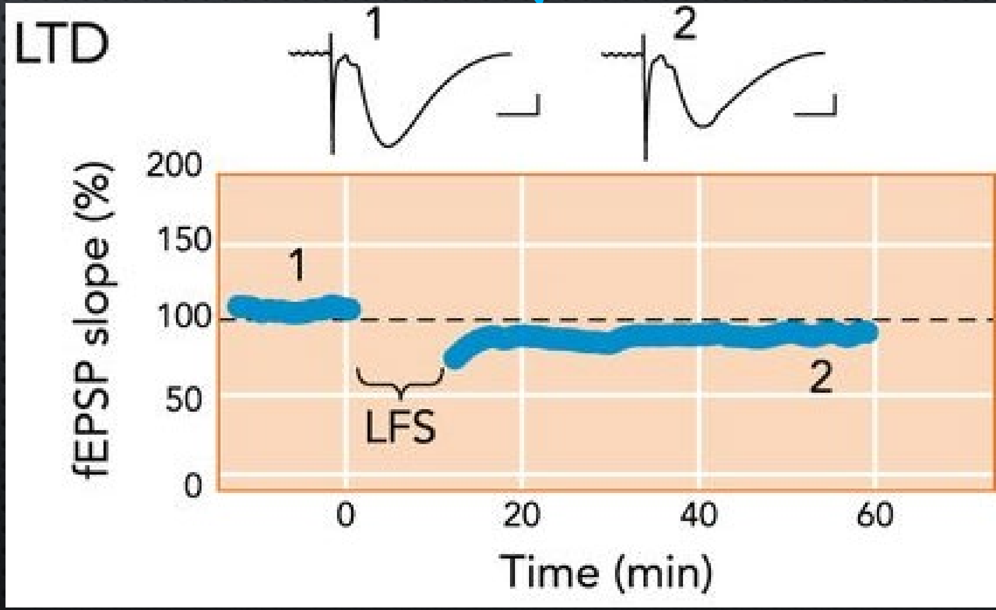


Hidalgo and Arias-Cavieres, *Physiology*, 2016

Long Term Depression (LTD)

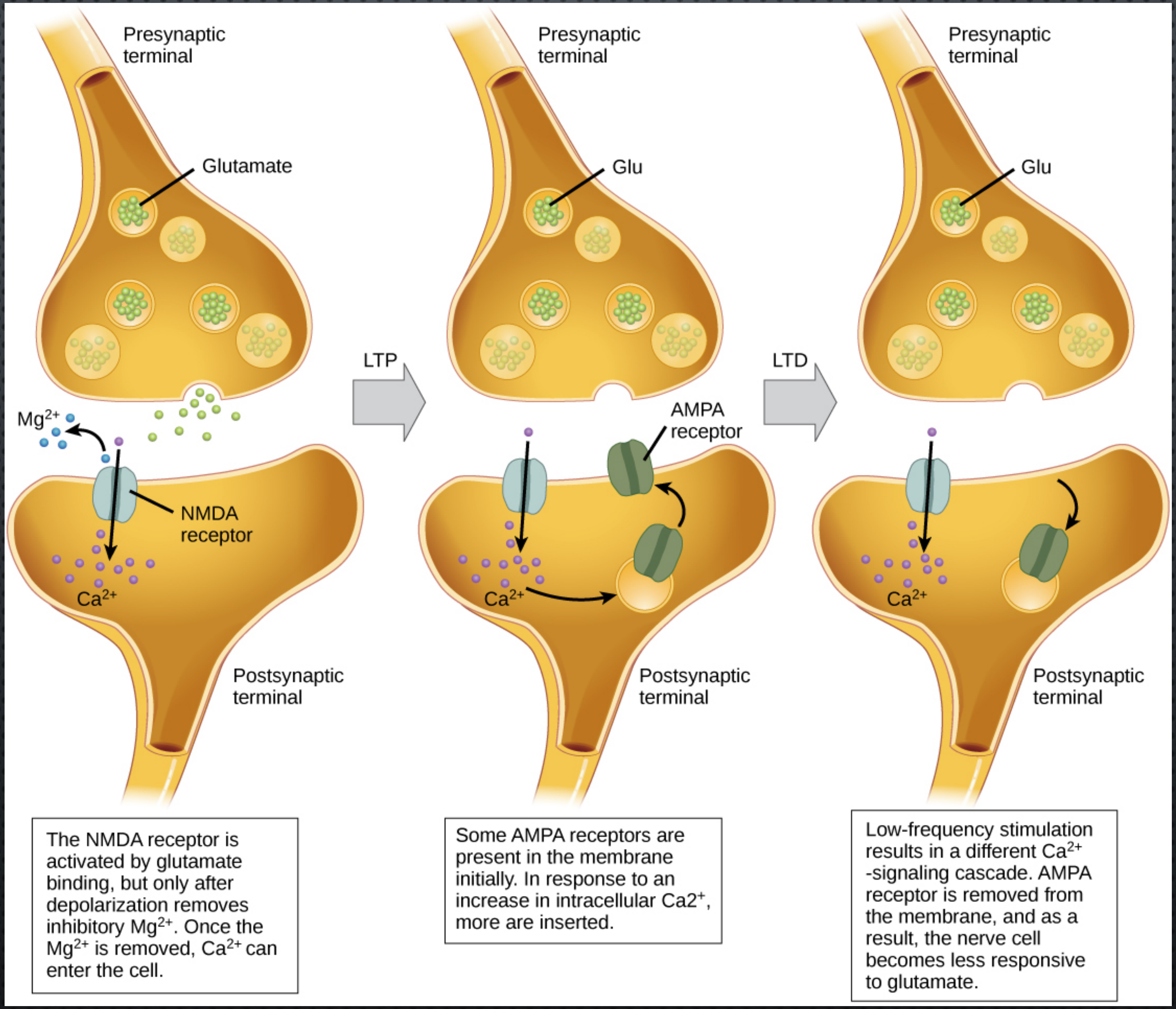






Hidalgo and Arias-Cavieres, *Physiology*, 2016

Chemical basis for LTP and LTD



https://archive.cnx.org/contents/939dbad7-f735-4ea2-bd5f-75384ffbc06b@1/derived-copy-of-how-neurons-communicate

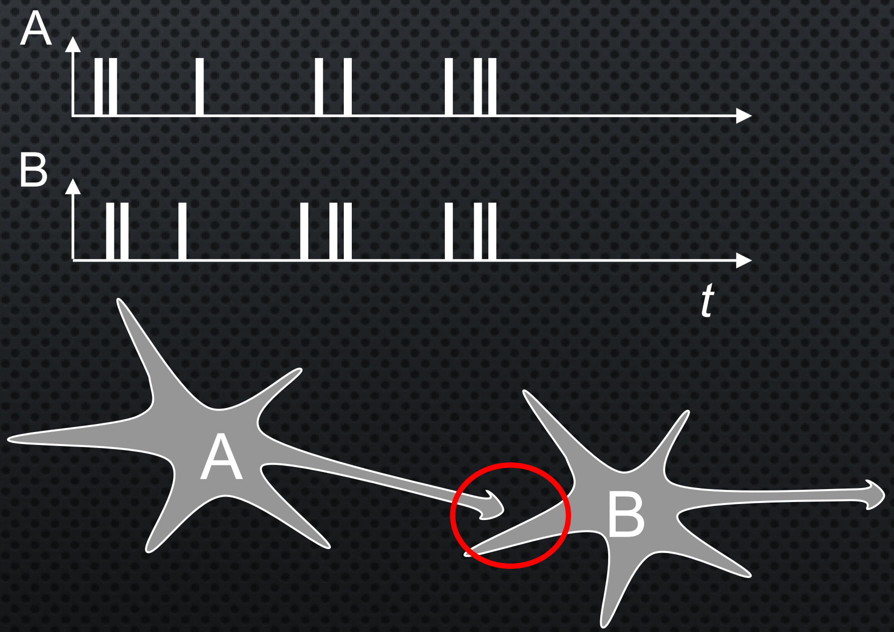
Hebbian learning

- Postulated in 1949 by Canadian psychologist Donald O. Hebb in his book "The organization of behaviour"

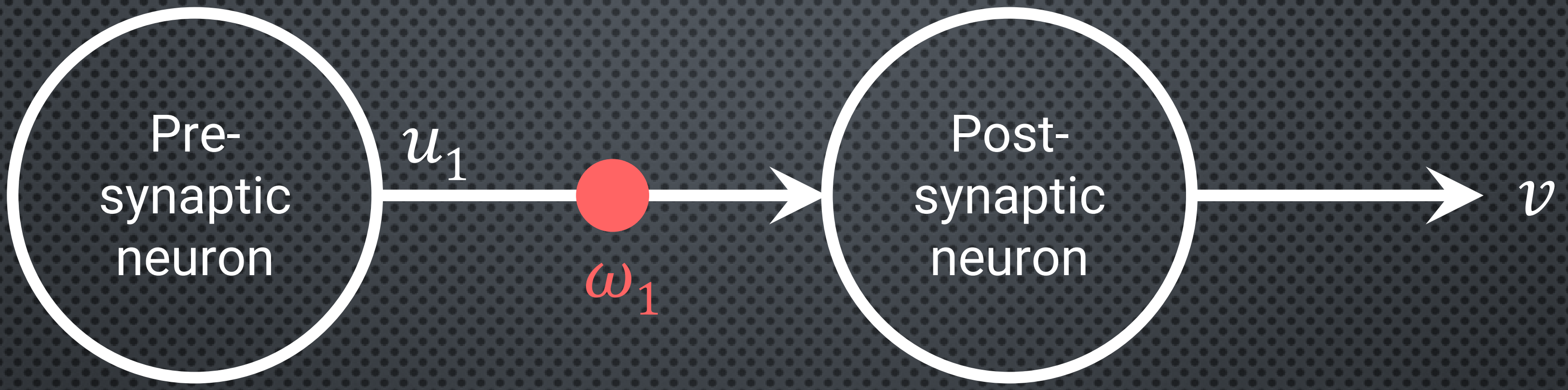
- *"When an axon of a cell A is near enough to excite cell B or repeatedly or persistently takes part in firing it, some growth or metabolic change takes place in both cells such that A's efficiency, as one of the cells firing B, is increased."*

- The functional basis for this change is correlation between presynaptic and postsynaptic activity – "neurons that fire together, wire together"

- Hebbian learning is temporal correlation-based, associative, unsupervised learning



Hebbian learning: a simple model



ω_1 always grows → **UNSTABLE !**

$$\frac{d\omega_1}{dt} = \mu \cdot v \cdot u_1$$

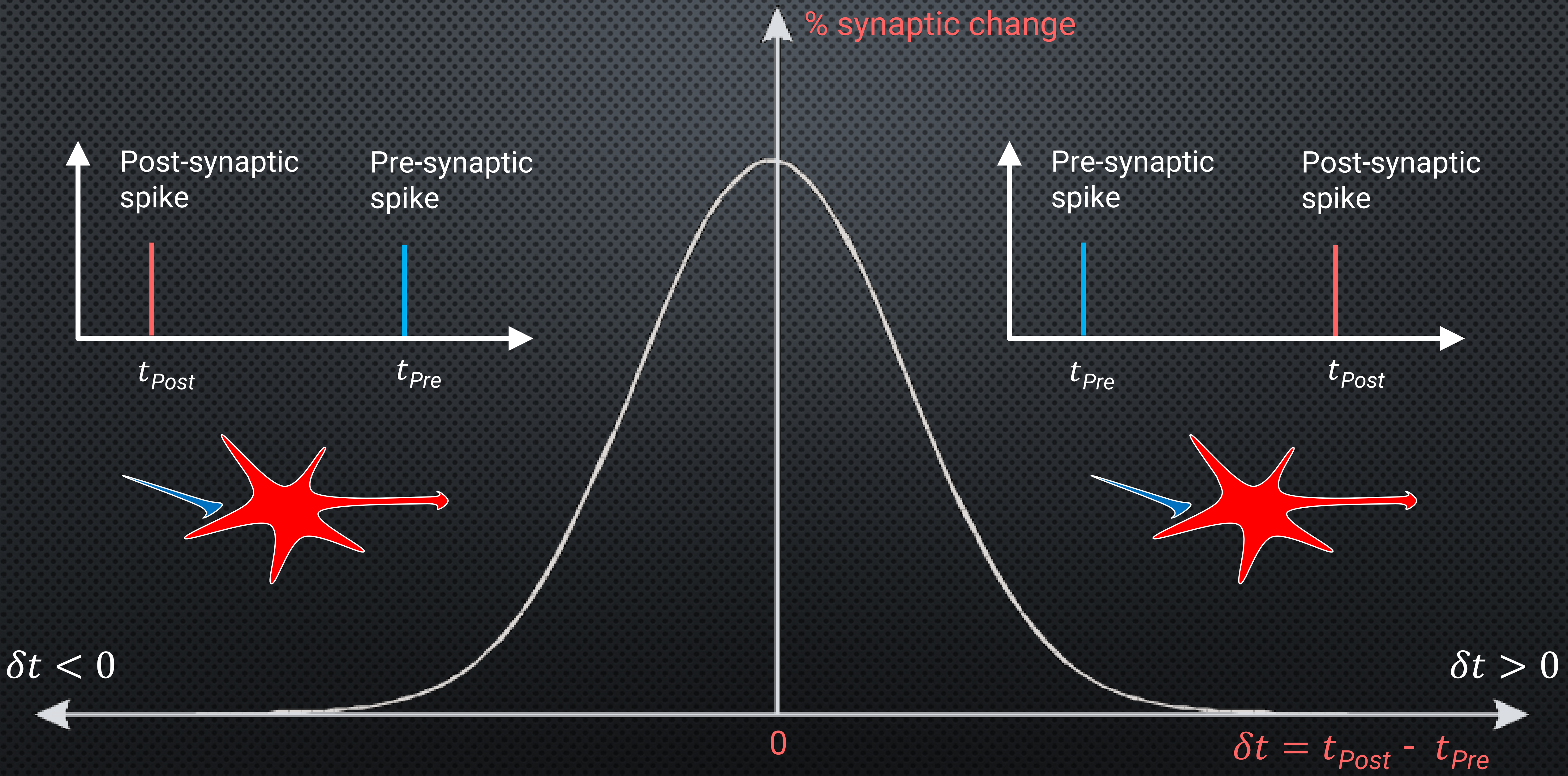
Output of pre-synaptic neuron / input to post-synaptic neuron

Output of post-synaptic neuron

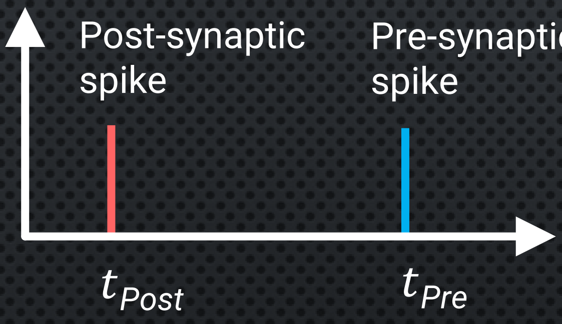
Change in synaptic weight

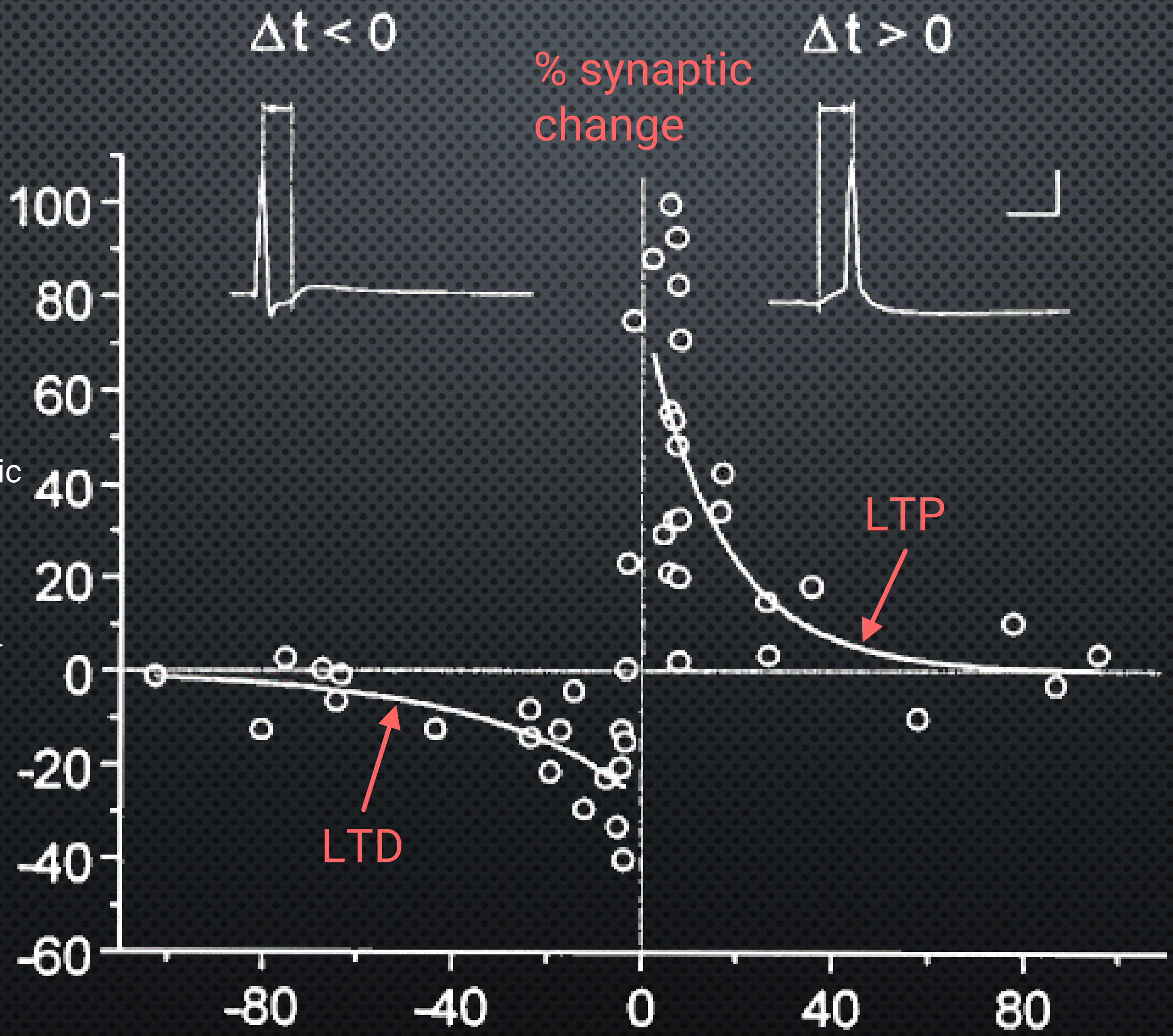
Learning rate $(\mu \ll 1)$

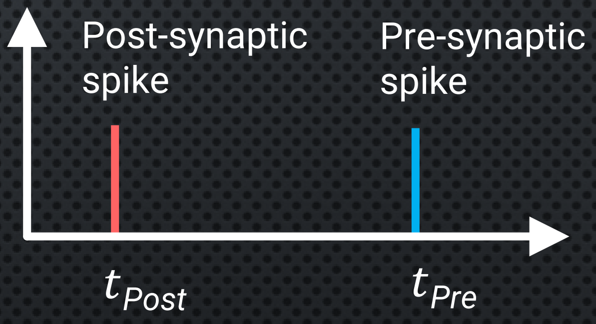
Hebbian learning: LTP



Spike-timing dependent plasticity (STDP): LTP and LTD



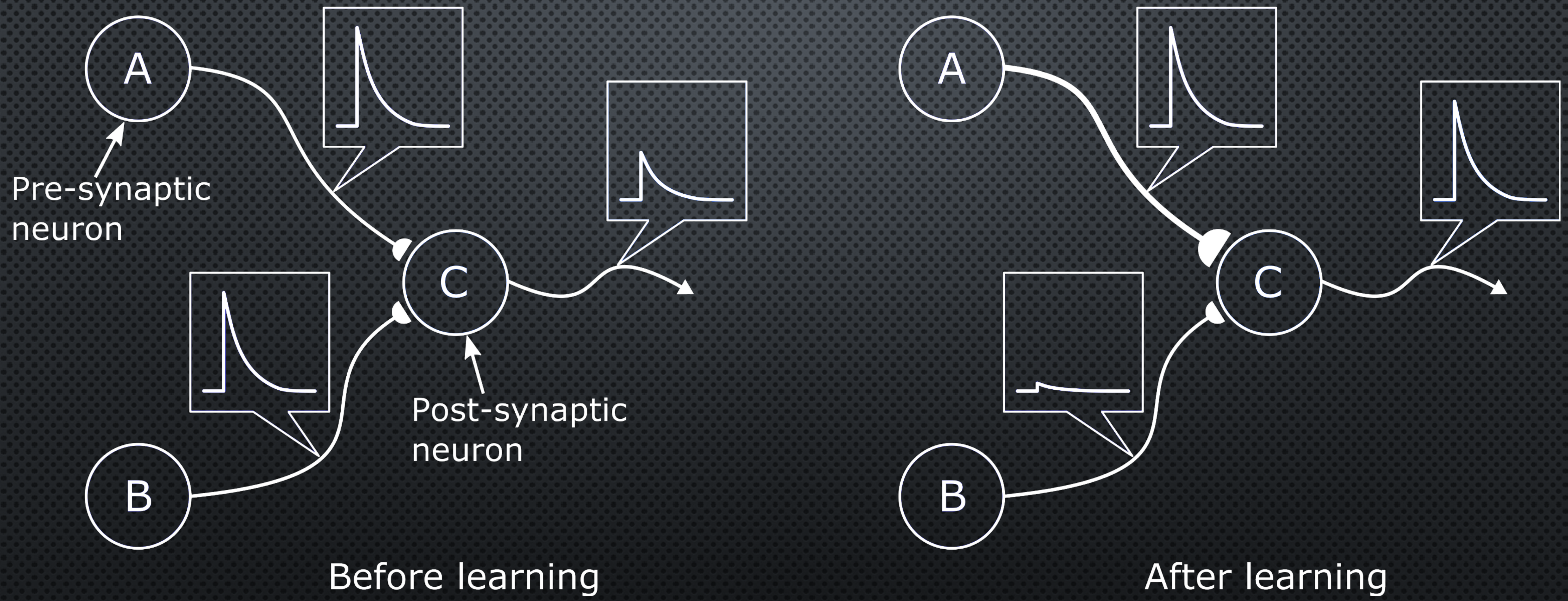




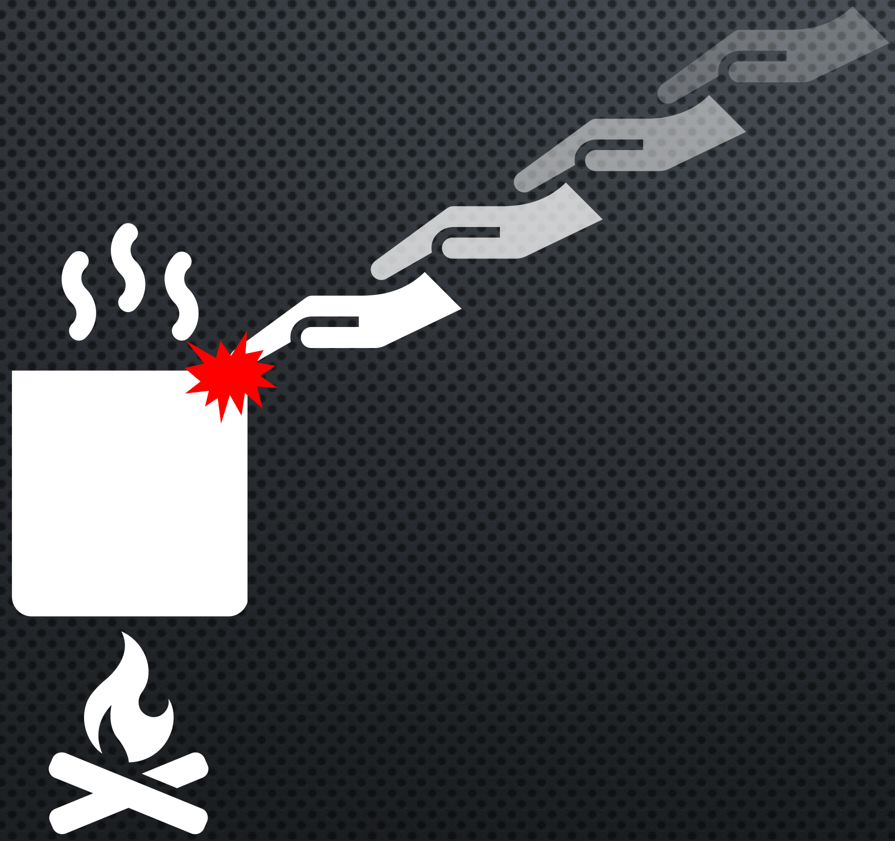
Other forms of functional plasticity

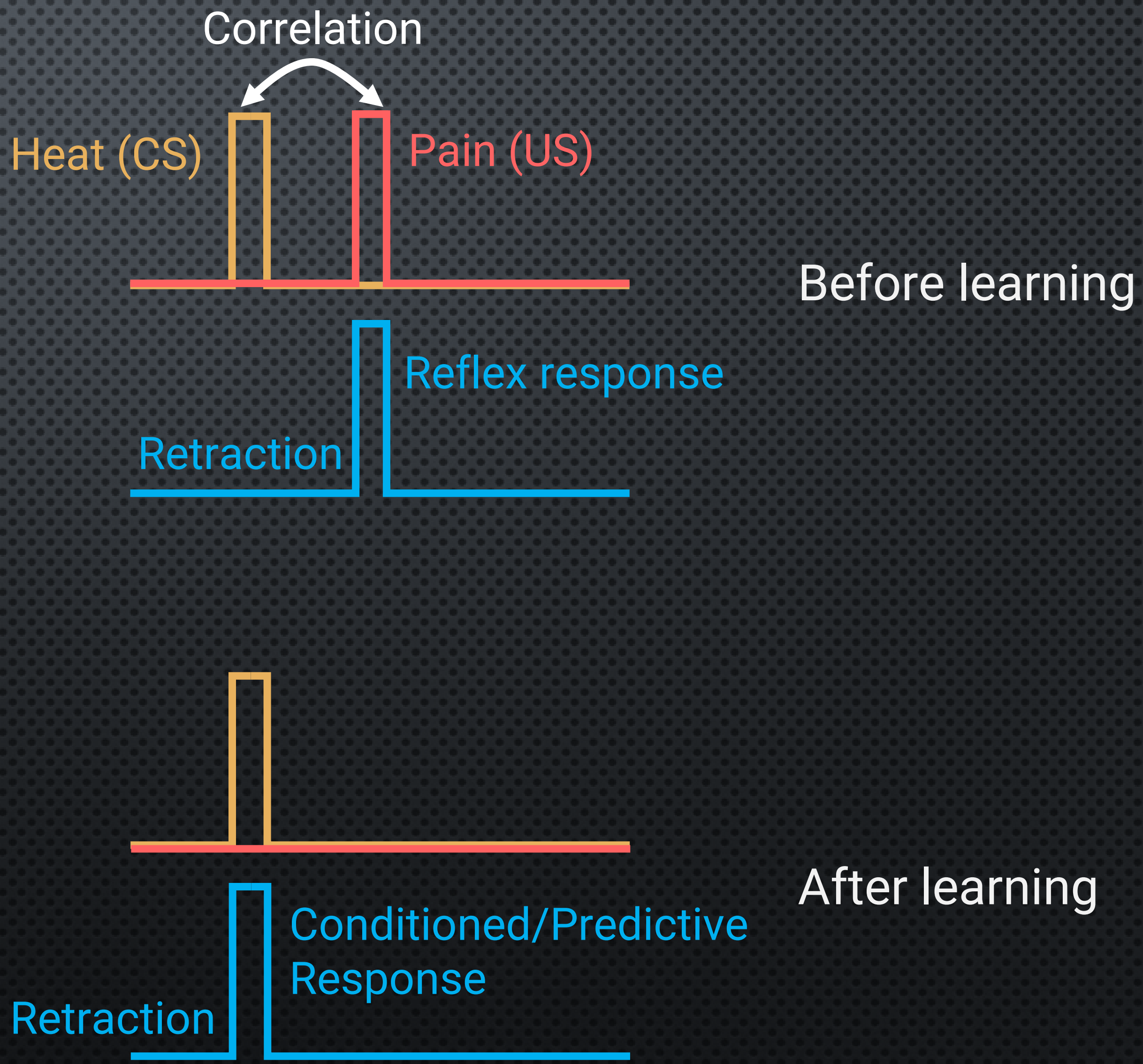
Homosynaptic plasticity	changes in synapse strength occur only at post-synaptic targets that are specifically stimulated by a pre-synaptic target
Heterosynaptic plasticity	activity of a third neuron can releases chemical neuromodulators that induce changes in synaptic strength between two other neurons
Non-synaptic plasticity	intrinsic excitability, i.e. sensitivity to synaptic input, of neurons can be altered and is manifested as changes in the firing characteristics of the neuron itself
Homeostatic plasticity	capacity of neurons to regulate their own excitability relative to network activity, a compensatory adjustment that occurs over the timescale of days

Heterosynaptic plasticity



Learning quickly from temporal correlations

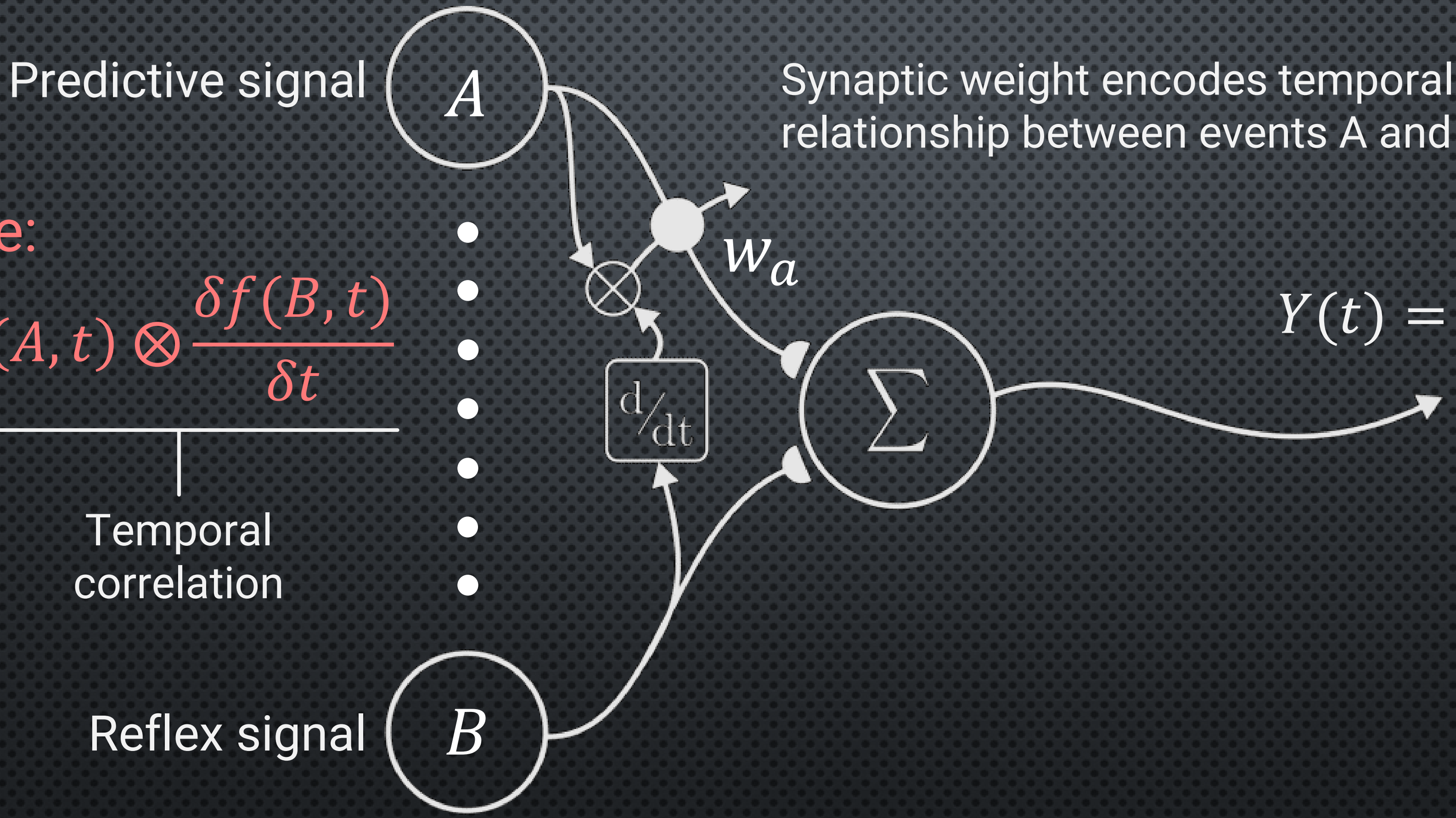




Input correlation learning (ICO learning)

Learning rule:

$$\frac{\delta w_a}{\delta t} = \underbrace{\eta}_{\text{Learning rate}} \cdot \underbrace{f(A,t) \otimes \frac{\delta f(B,t)}{\delta t}}_{\text{Temporal correlation}}$$



Synaptic weight encodes temporal relationship between events A and B

$$Y(t) = w_a \cdot f(A,t) + f(B,t)$$

⊗ Cross-correlation

Plastic synapse

Excitatory synaptic input

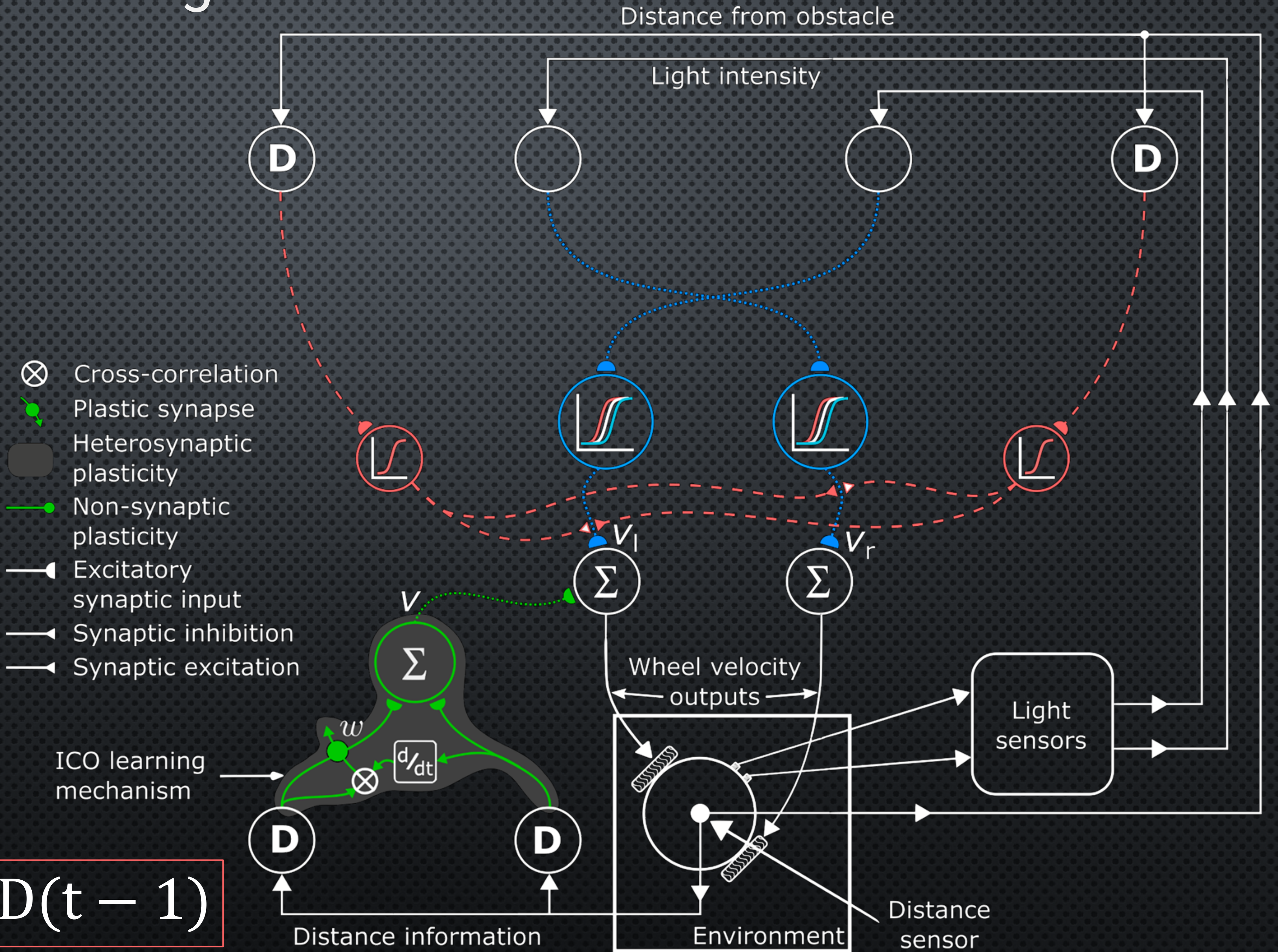
Porr et al., 2006

Artificial brain with learning

Learning rule:

$$\frac{dw}{dt} = \mu \cdot \mathrm{D} \cdot \frac{Dt}{dt}$$

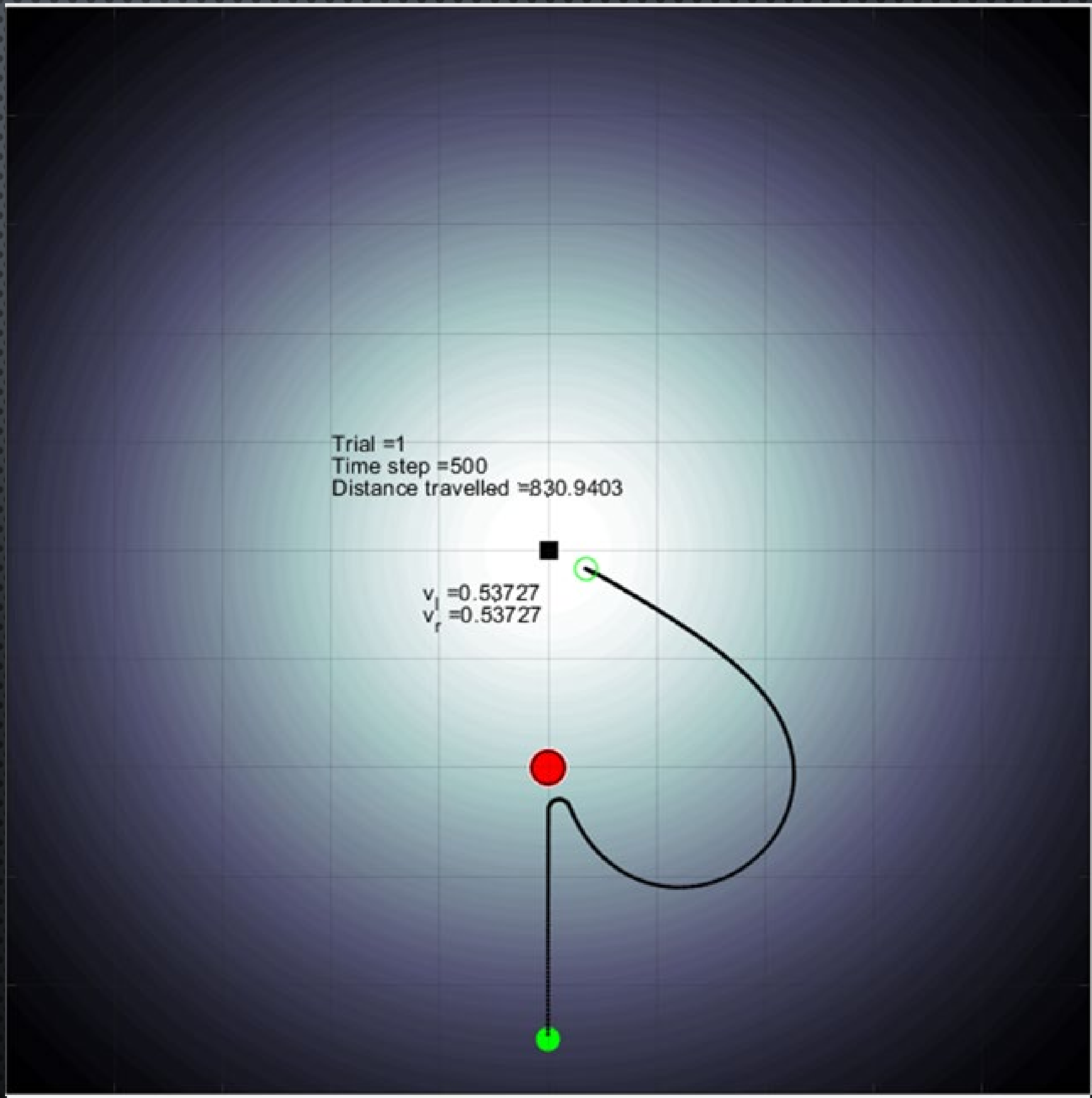
$$\boxed{v = \omega \cdot \mathrm{D(t)} + \mathrm{D(t-1)}}$$



Matlab exercise: obstacle avoidance learning

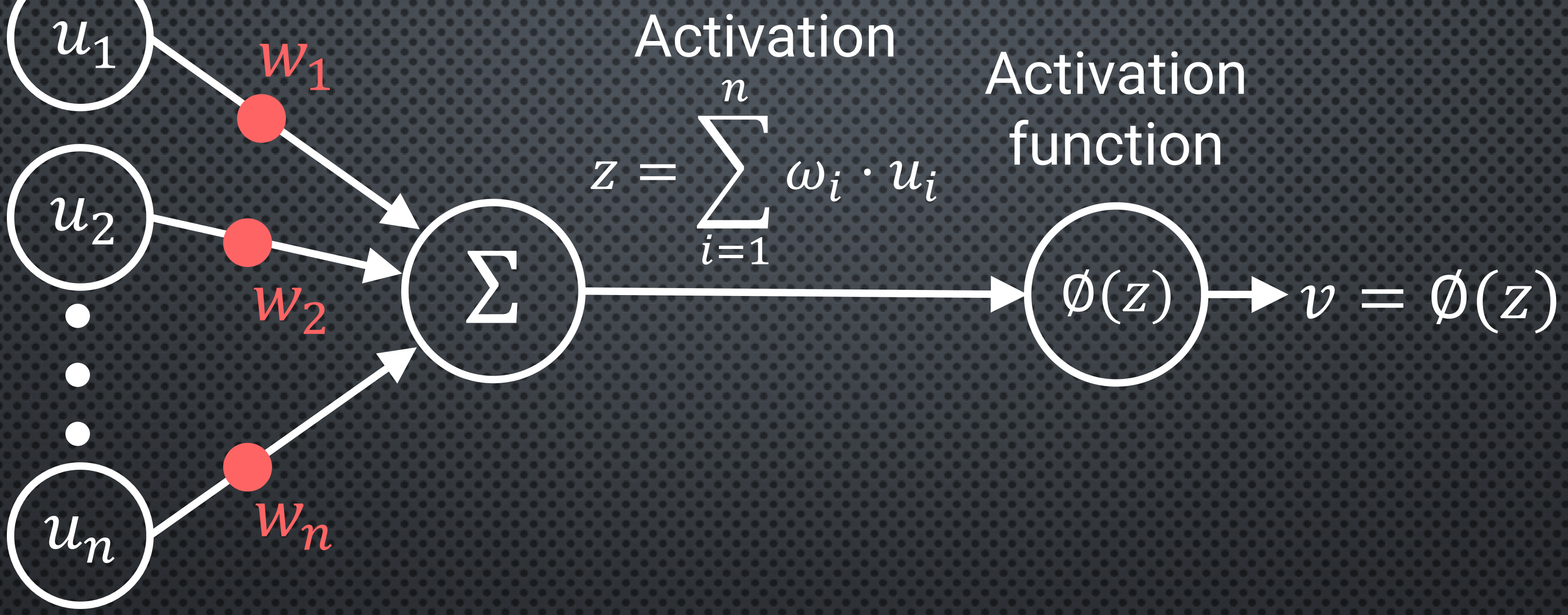
- Download “Assignment 3.pdf” and “obstacle_avoidance_ learning.m”

- Follow instructions in PDF file



5 Supervised and unsupervised learning in artificial neural brains

Learning in perceptrons



Perceptrons have 2 sets of parameters - weights and activation function, and both affect the output.

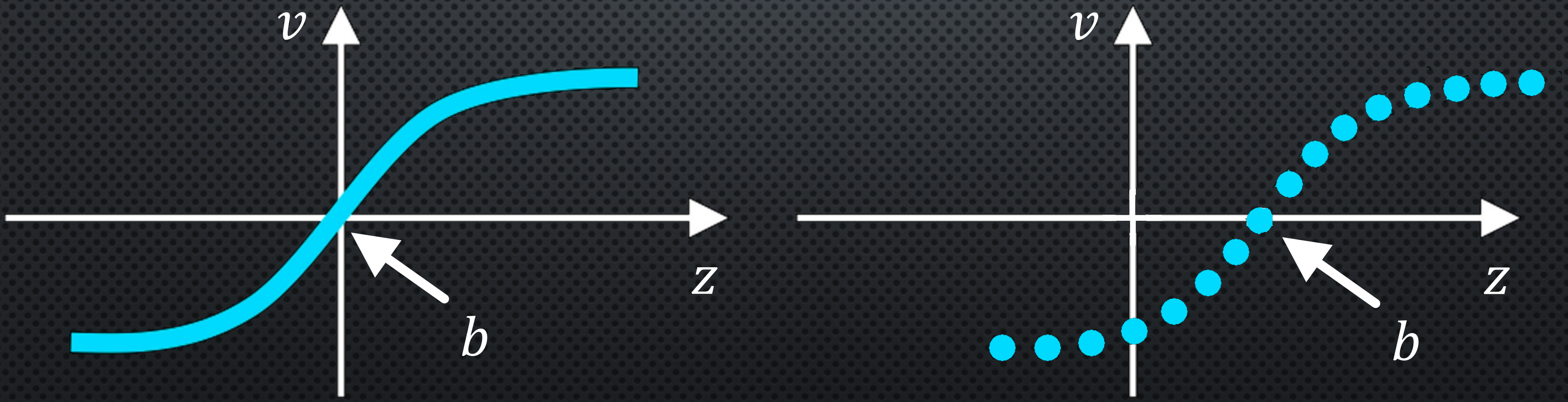
Question: So which one should we modify to learn?

Answer: Ideally, both should be modified.

Introducing *bias*: a way to describe the activation function

- Bias is the point on the z-axis at which $v = 0$
- By changing bias, one can shift the activation function to the left or right.

Example: Sigmoid activation function $v = \dfrac{1}{1 + e^{-S(z-b)}}$ where b is the bias and S determines the slope of linear part.



Question: So how do we decide what should be the value of bias b?

Answer: We could learn it by considering b as another “weight”.

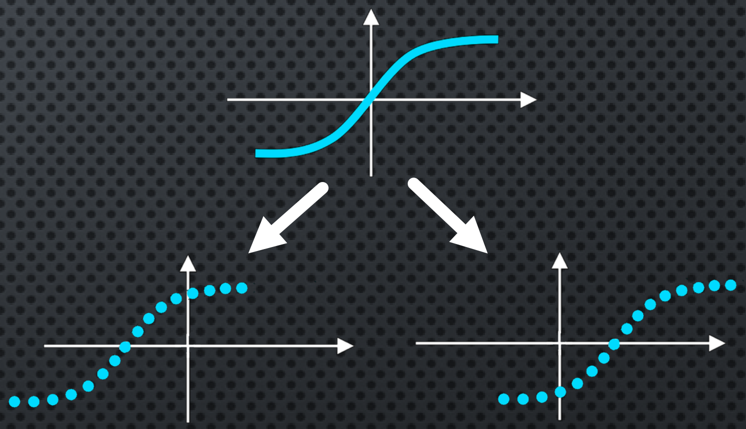
Bias b as a weight

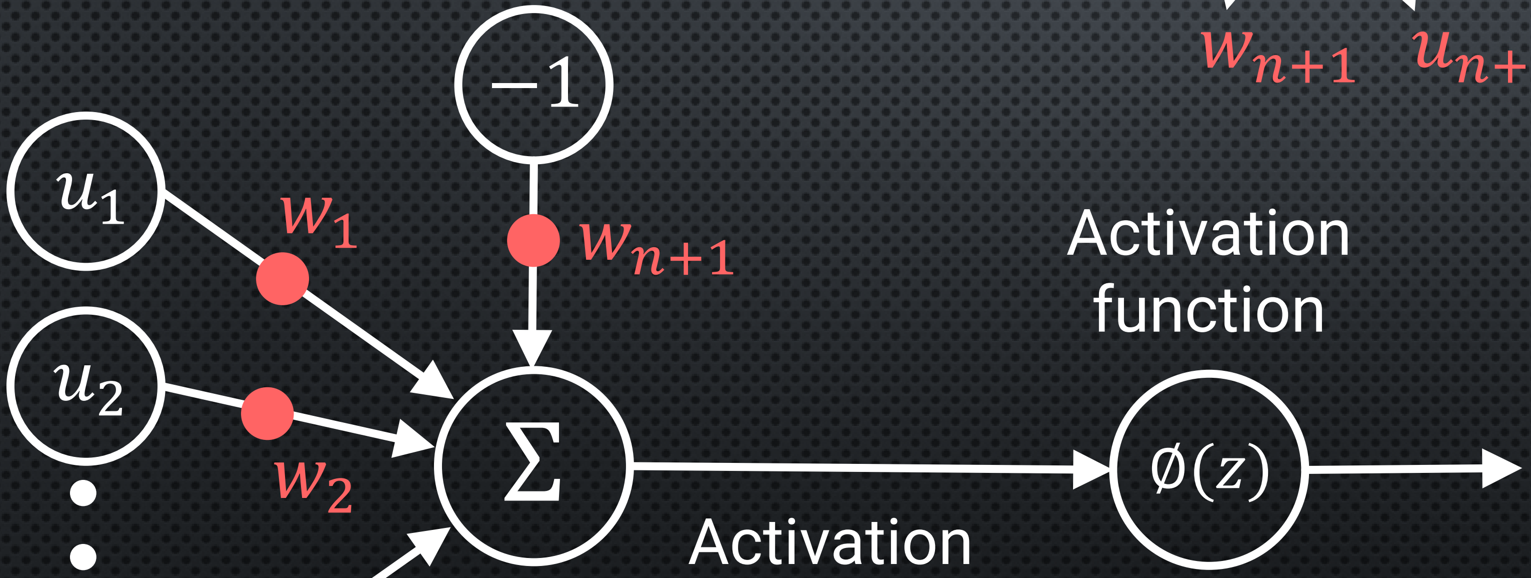
$$z = w_1 u_1 + w_2 u_2 + \cdots + w_n u_n$$

$$\rightarrow (z - b) = w_1 u_1 + w_2 u_2 + \cdots + w_n u_n - b$$

$$\rightarrow (z - b) = w_1 u_1 + w_2 u_2 \ldots + w_n u_n + \underbrace{b}_{w_{n+1}}\ \underbrace{(-1)}_{u_{n+1}}$$

This small change sets the AF in the centre **<u>before learning</u>**. By learning w_{n+1}, we can shift the activation function automatically.



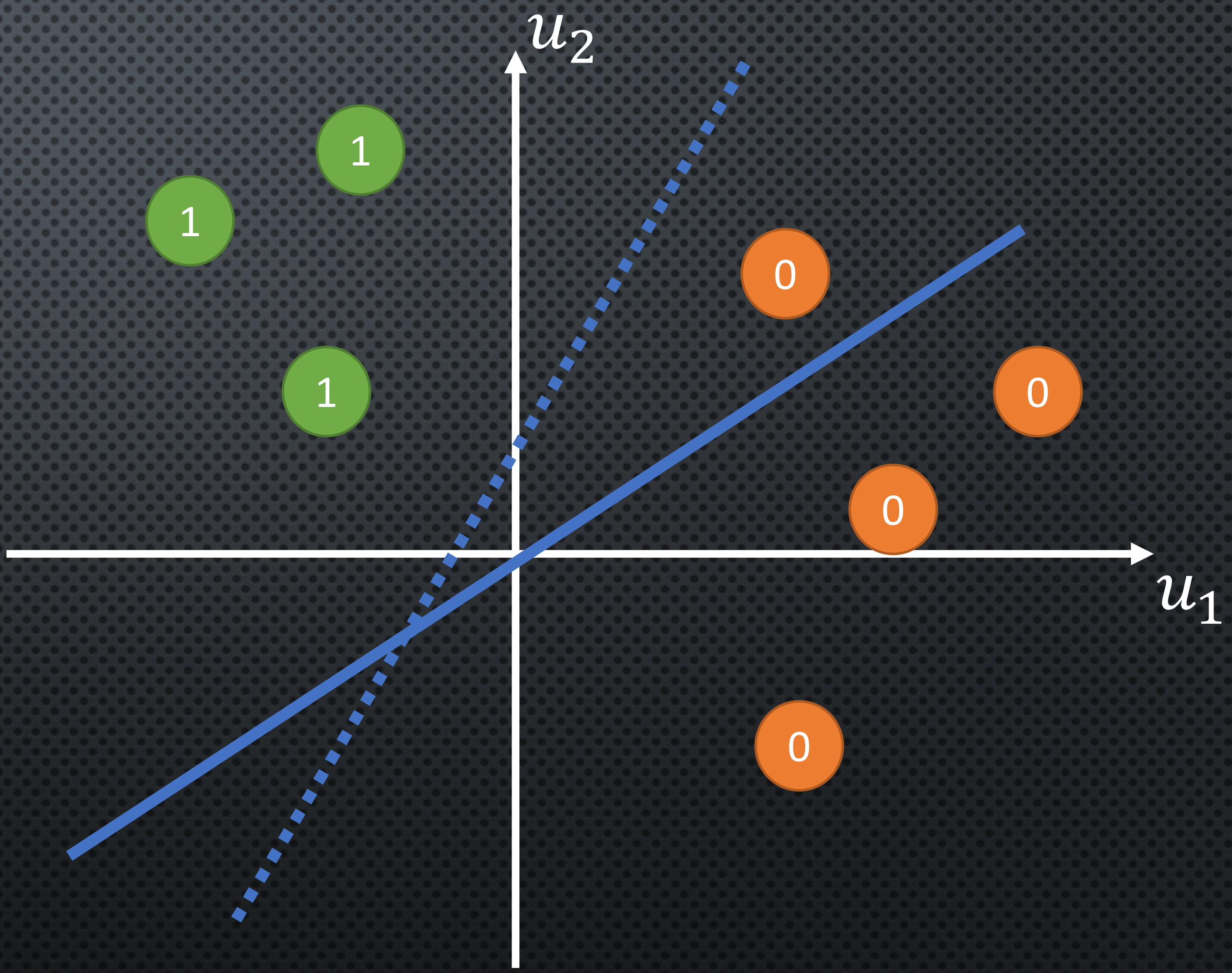


$$z = \sum_{i=1}^{n+1} \omega_i \cdot u_i$$

$$v = \frac{1}{1 + e^{-S(z-b)}}$$

Why shift the activation function?

- The blue line is called the decision boundary.
- Decision boundary separates inputs into different classes (for a classification problem). Here the classes are “0” and “1”, but it can also be labels such as “cat”, “dog” etc.
- By changing $w_1, w_2, \ldots, w_n, w_{n+1}$ by learning, the decision boundary can be shifted to adapt to new input data.

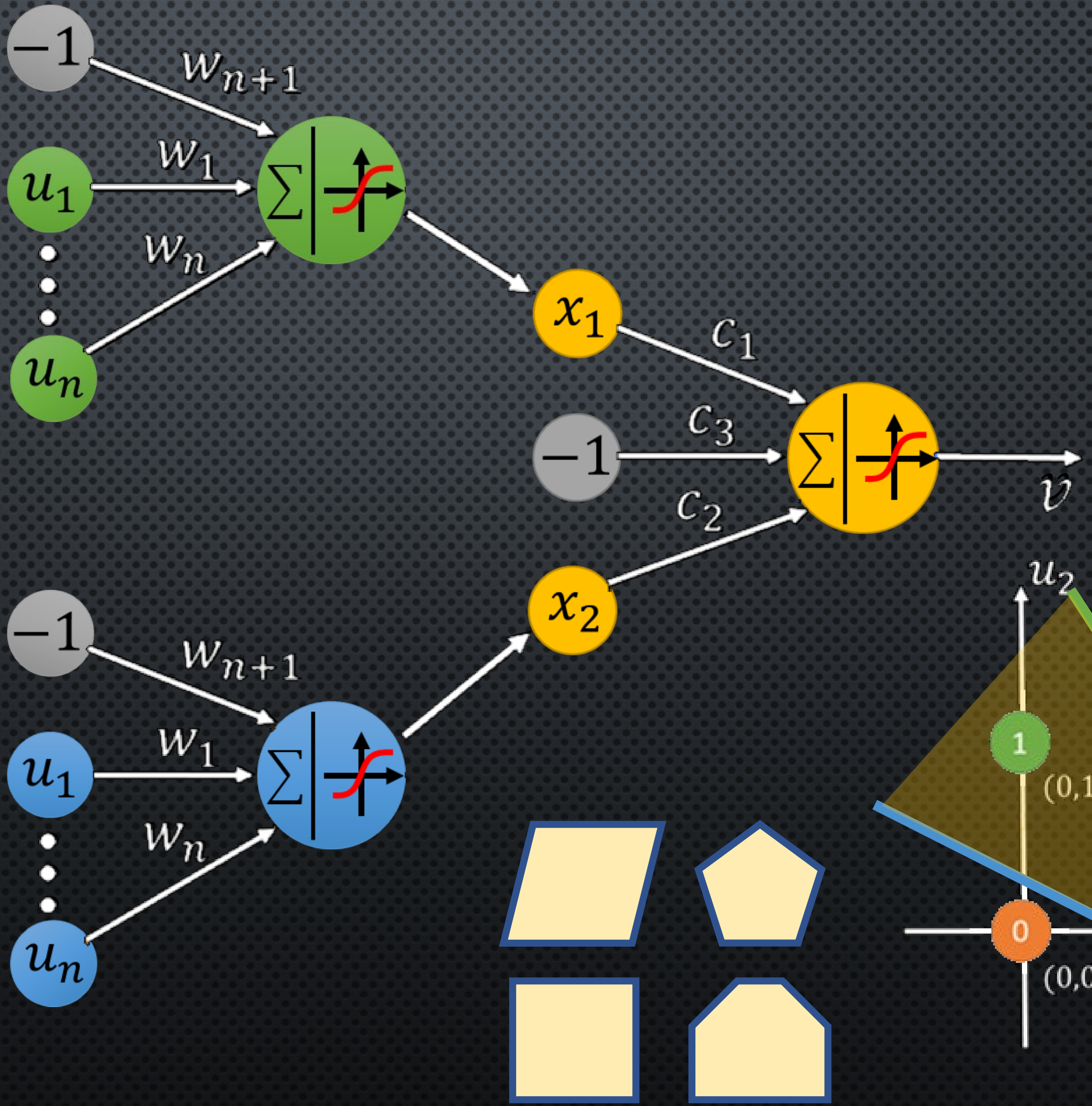


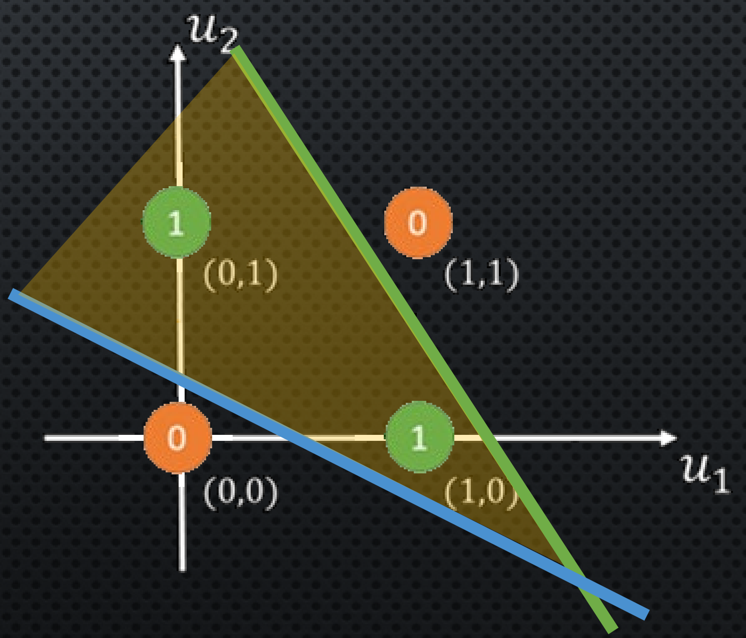
Equation for blue line:

$w_1 u_1 + w_2 u_2 \ldots + w_n u_n + b(-1) = (z - b)$

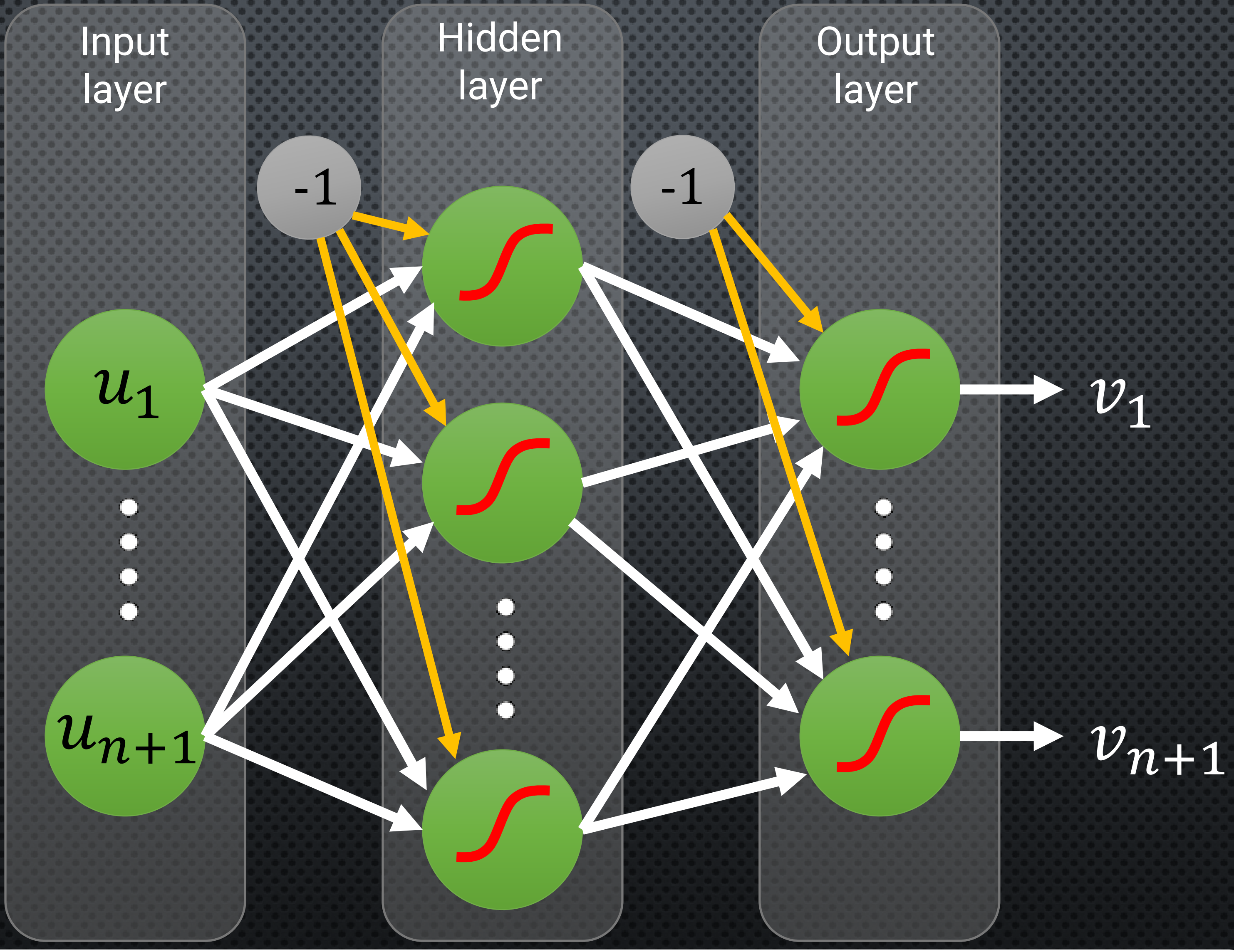
From decision boundaries to decision surfaces

- By adding a third perceptron as the next layer, we get a 2-dimensional decision surface.
- By adding more perceptrons in the first layer, we can draw more decision boundaries to enclose more complex decision surfaces.



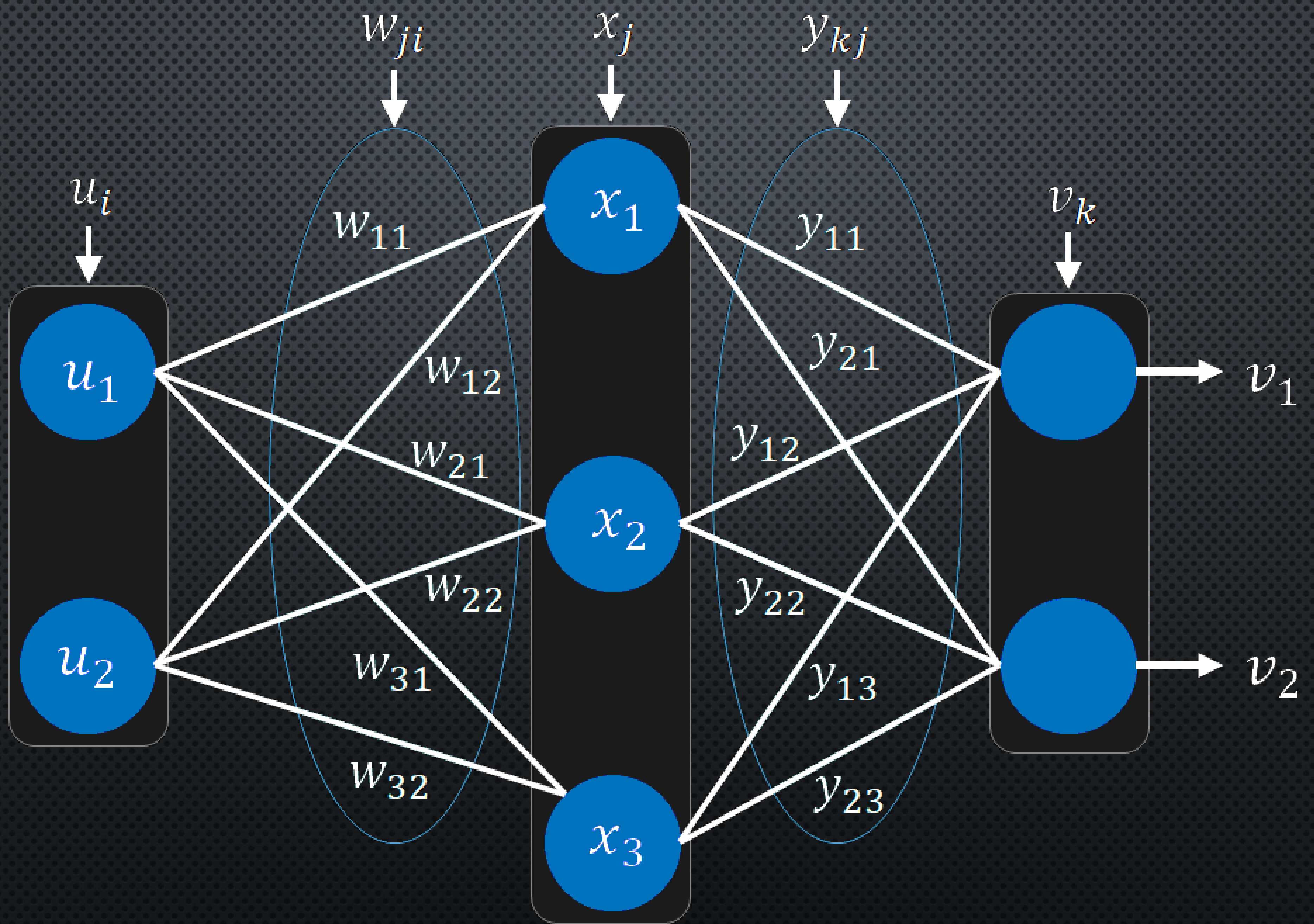


Multi-Layer Perceptron (MLP)



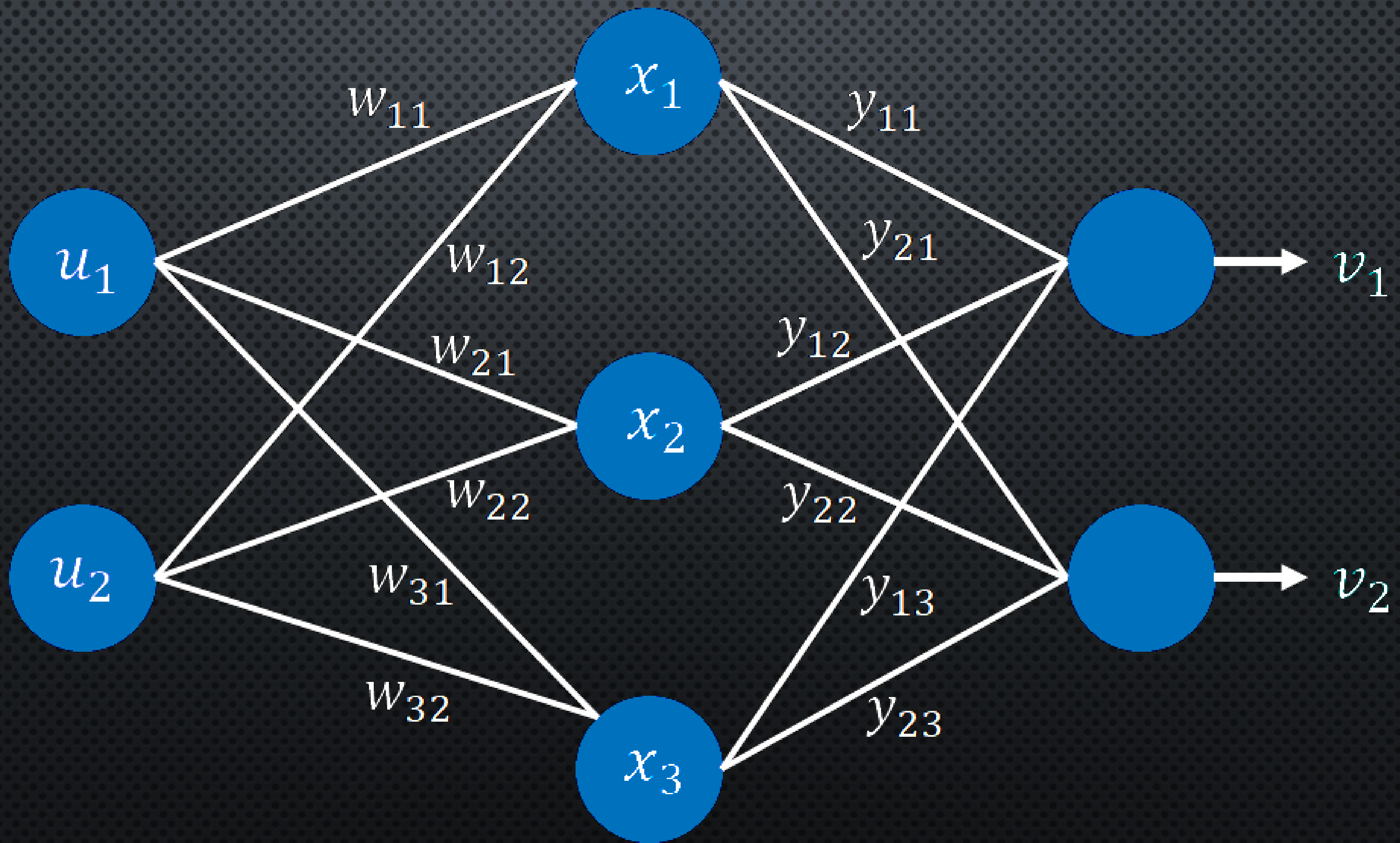
In deep learning neural networks, there is **more than one** hidden layer

Training a MLP: Notation



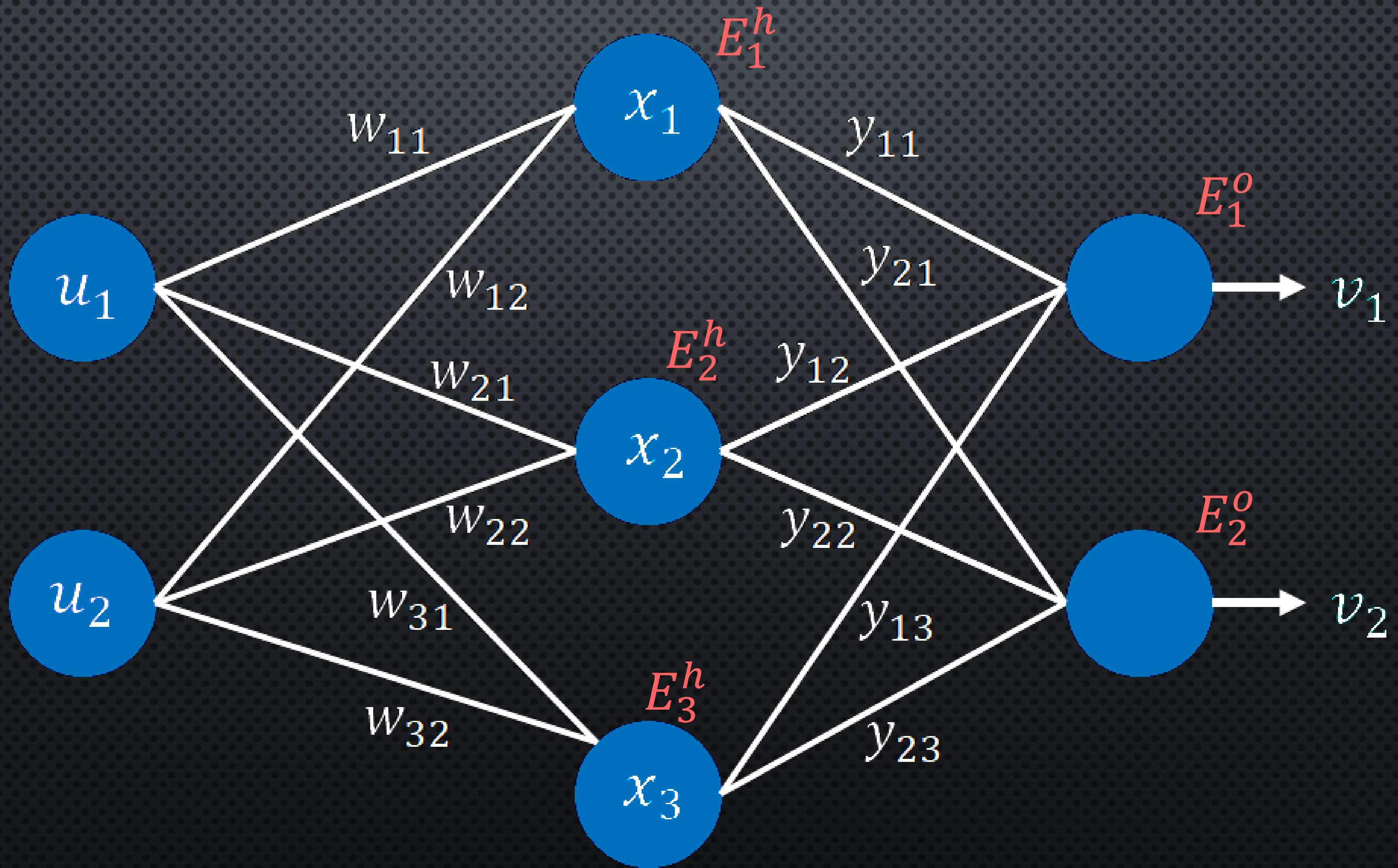
Step 1: Forward propagation

- Calculate output $\boldsymbol{x_j} = \sum_i u_i\, w_{ji}$ for all hidden neurons
- Calculate output $\boldsymbol{v_k} = \sum_j x_j\, y_{kj}$ for all output neurons



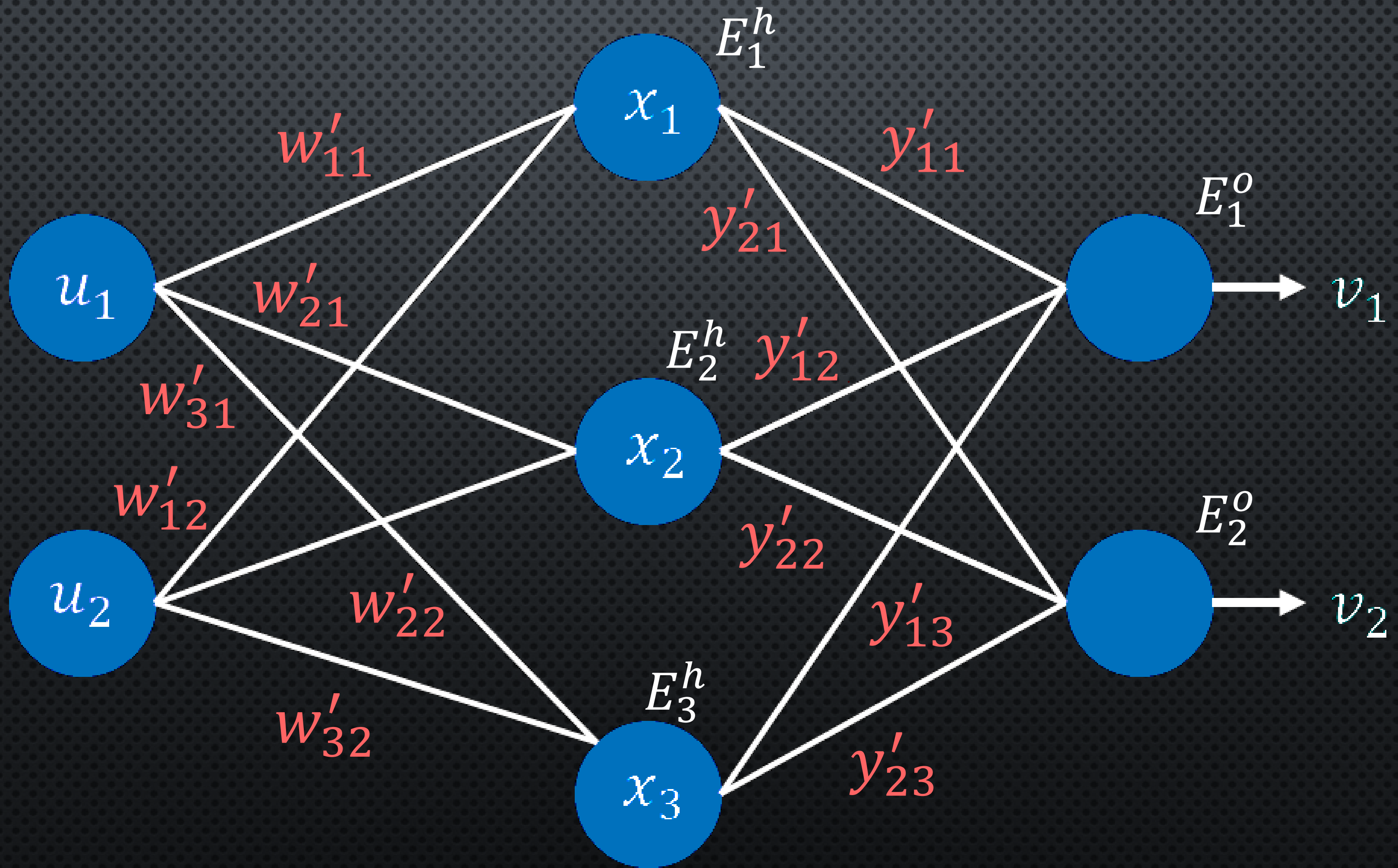
Step 2: Backpropagation

- Calculate error gradient for all output neurons, $\boldsymbol{E_k^o} = \boldsymbol{v_k}(1 - \boldsymbol{v_k})(\boldsymbol{t_k} - \boldsymbol{v_k})$
- Calculate error gradient for all hidden neurons, $\boldsymbol{E_j^h} = \boldsymbol{x_j}(1 - \boldsymbol{x_j}) \sum_k E_k^o \, y_{kj}$



Step 2: Backpropagation

- Update weights for the output neurons, $y'_{kj} = y_{kj} + \mu\, E^o_k\, x_j$
- Update weights for the hidden neurons, $w'_{ji} = w_{ji} + \mu\, E^h_j\, u_i$



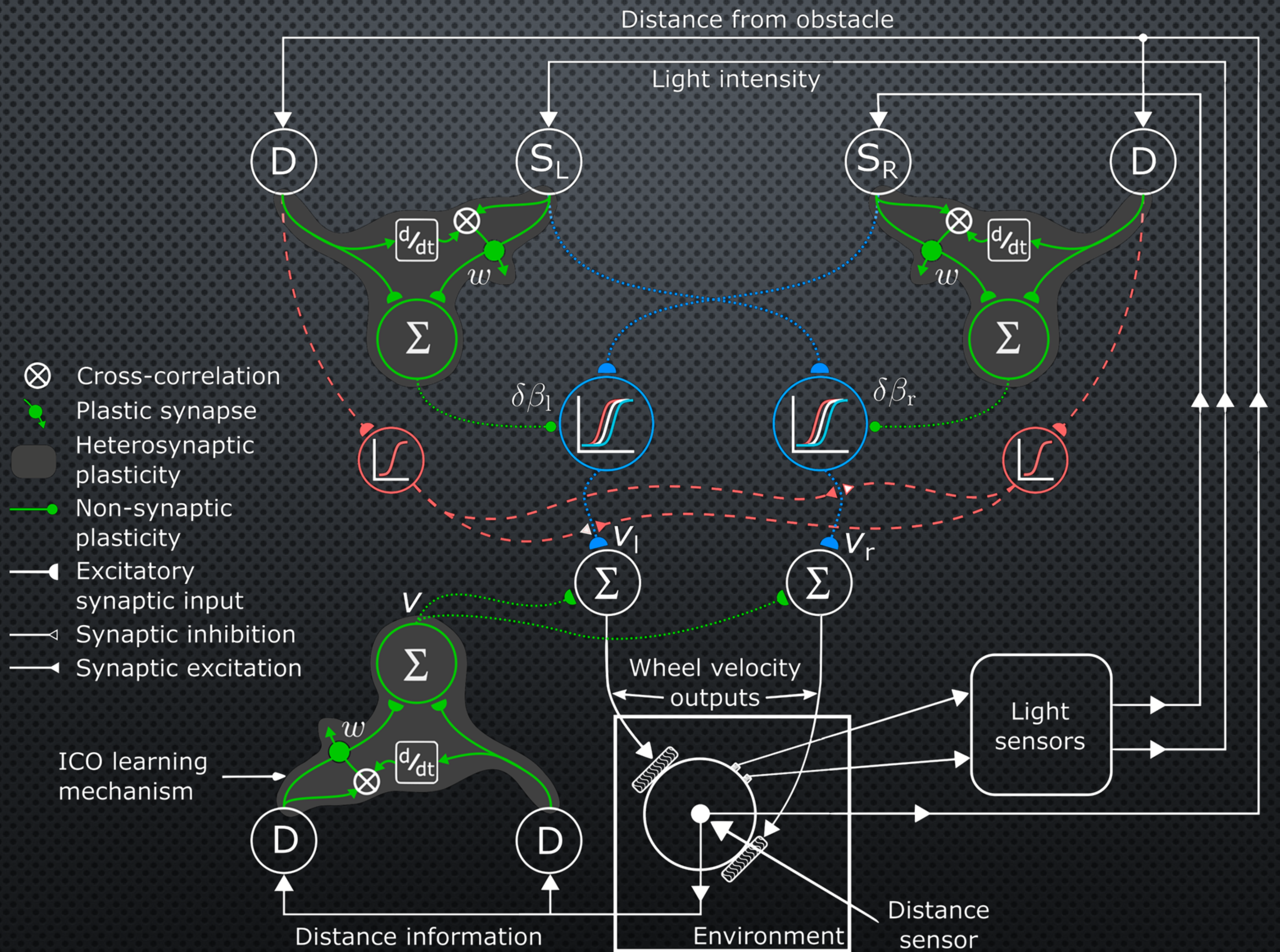
Training a MLP: backpropagation algorithm

- Divide dataset into two sets – training dataset (70% of total input data points) and testing dataset (30% of total input data points)
- Initialise all weights to random values between 0 and 1 (or -1 and +1)
- Step 1: Feed forward
  1. Randomly shuffle training dataset and select a (new) randomly chosen input data point
  2. Calculate output $\boldsymbol{x_j} = \sum_j \sum_i u_i\, w_{ji}$ for all hidden neurons, and $\boldsymbol{v_k} = \sum_k \sum_j x_j\, y_{kj}$ for all output neurons
- Step 2: Backpropagation
  1. Calculate error gradient for all output neurons, $\boldsymbol{E_k^o} = \boldsymbol{v_k}(1 - \boldsymbol{v_k})(\boldsymbol{t_k} - \boldsymbol{v_k})$
  2. Calculate error gradient for all hidden neurons, $\boldsymbol{E_j^h} = \boldsymbol{x_j}(1 - \boldsymbol{x_j}) \sum_k E_k^o\, y_{kj}$
  3. Update weights for the output neurons, $\boldsymbol{y'_{kj}} = \boldsymbol{y_{kj}} + \mu\, \boldsymbol{E_k^o}\, \boldsymbol{x_j}$
  4. Update weights for the hidden neurons, $\boldsymbol{w'_{ji}} = \boldsymbol{w_{ji}} + \mu\, \boldsymbol{E_j^h}\, \boldsymbol{u_i}$
- Repeat Step 1 and Step 2 until the error is very low or max. number of epochs is reached
- Test your trained network on testing dataset by repeating Step 1

Additional notes on training

- Training set: Used to adjust weights of the neural network
- Validation set: Used to minimize overfitting
- Testing set: Used only for testing the final solution
- Monte Carlo cross validation
  - Sub-sample data randomly into training and test sets (e.g., 70% and 30%)
- K-fold cross validation
  - Divide data into *k* subsets
  - Each time (in total *k* times) one of the subsets is used for testing and the rest *k-1* subsets are joined and used as a training set
- Leave-p-out cross validation
  - Use *p* data samples as training samples and the rest (*n-p*) data samples as test samples
  - Train and test $\frac{n!}{p! \cdot (n-p)!}$ times

Learning the activation function



Hungry for more?