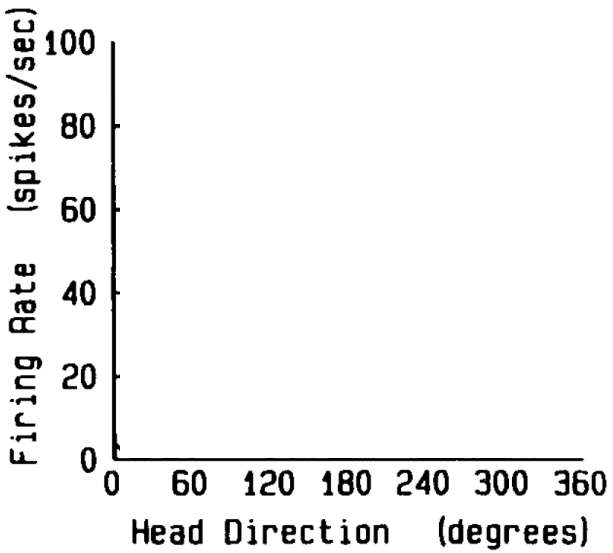
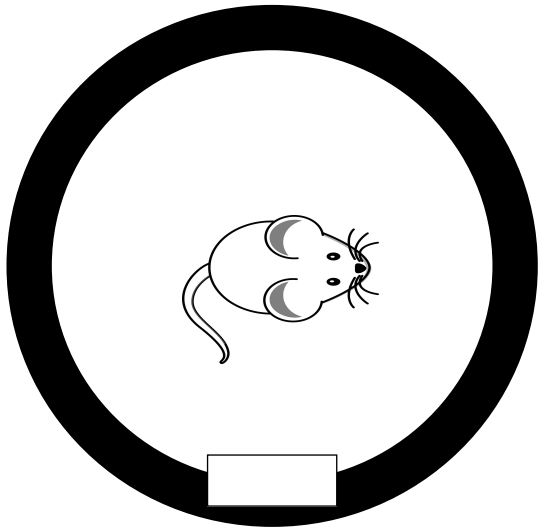
Compass in our head

– Head direction system –

Head Direction Cells

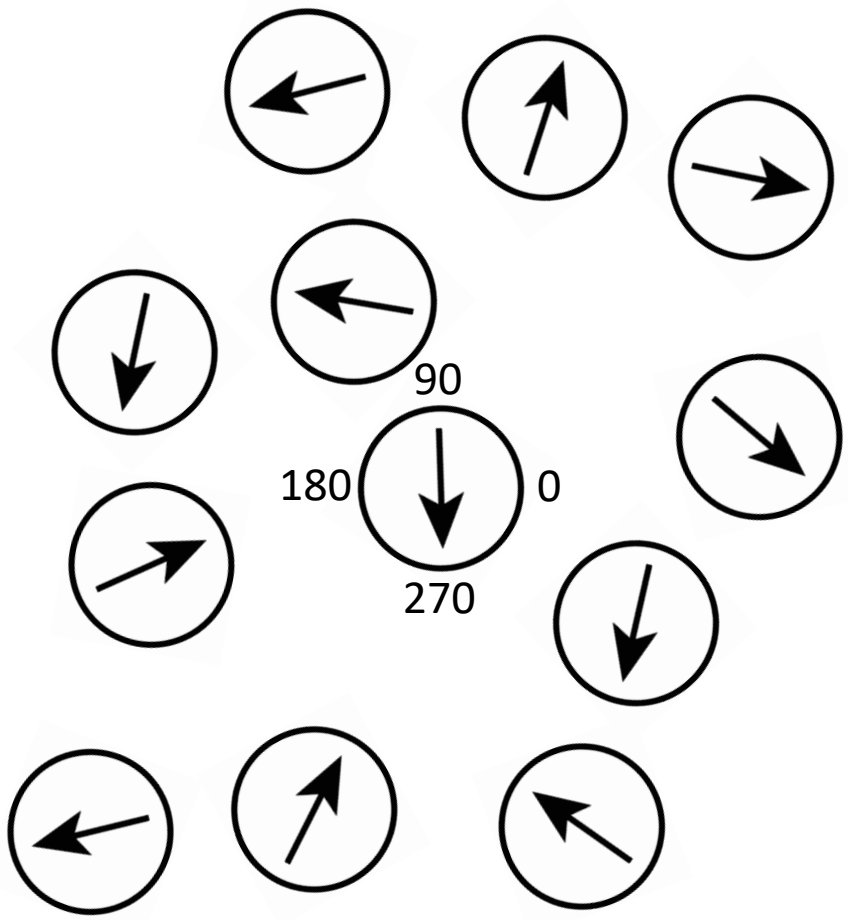
Single Neuron Tuning Curve

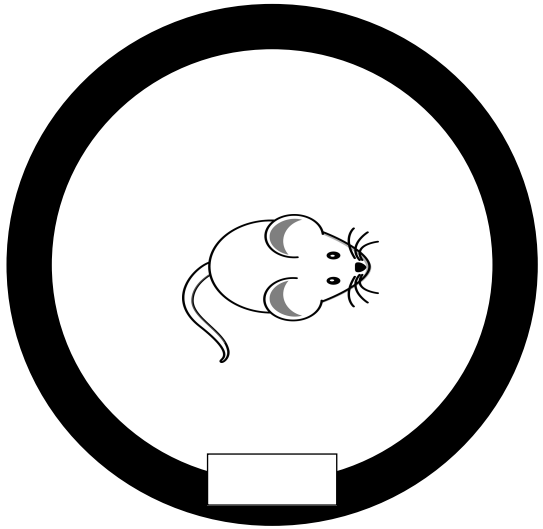




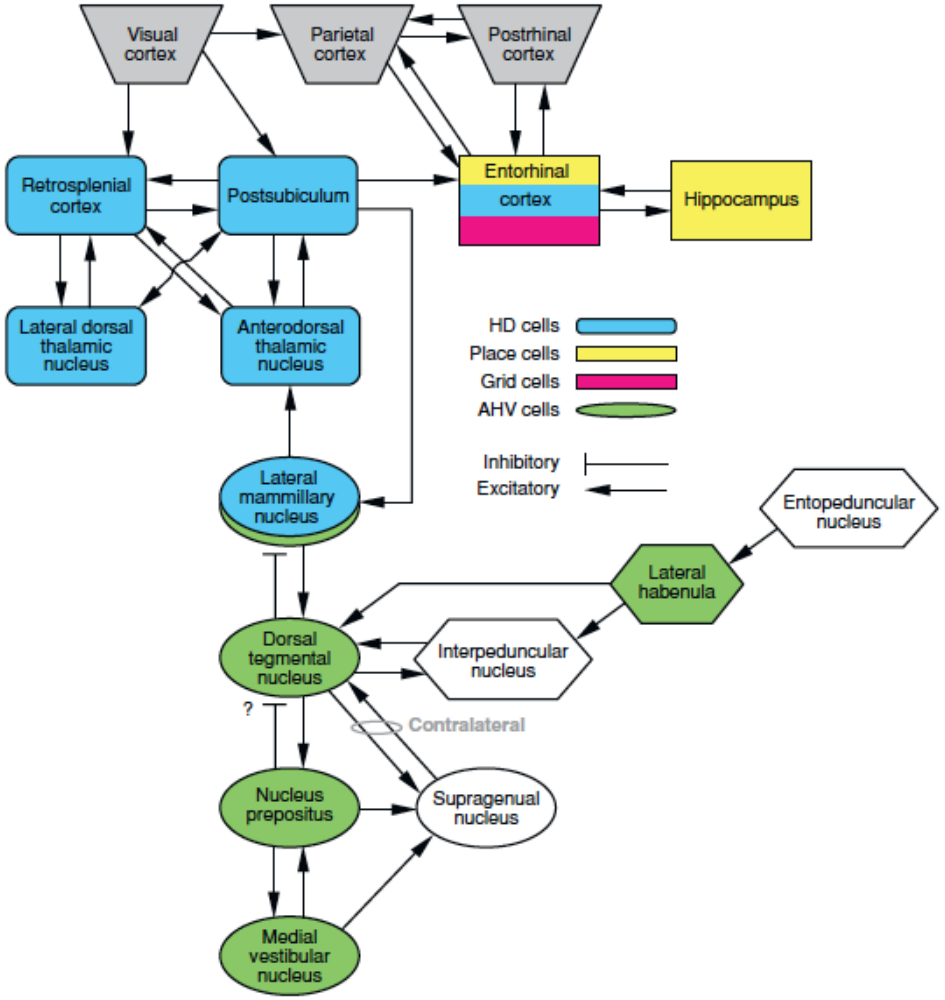
Taube et al. (1990) J. Neurosci.

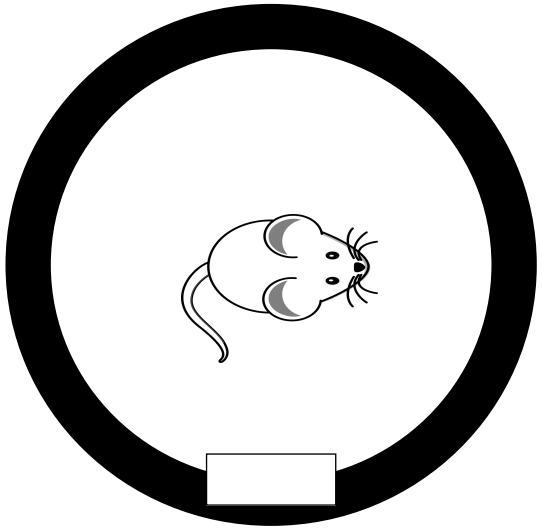
Head Direction Cells





Head Direction Cells





Taube (2007) Ann. Rev. Neurosci.

Baird, el Jundi et al.

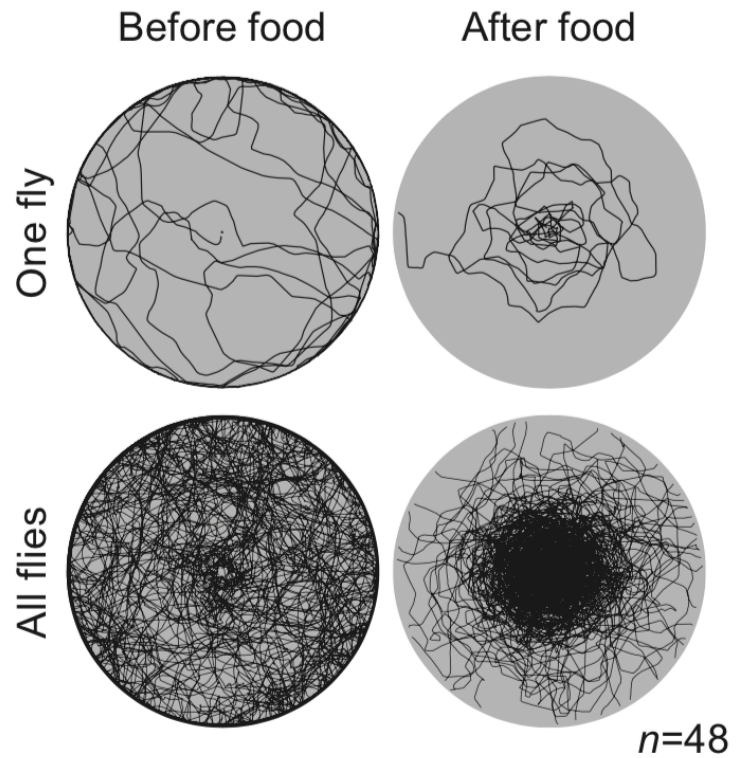
Reppert, Heinze et al.

von Frisch, Menzel, Srinivasan et al.

Condition: Oe-

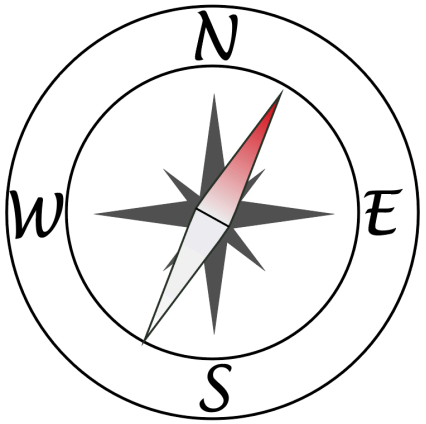
Fly: starved oe- female
Food source: yeast drop

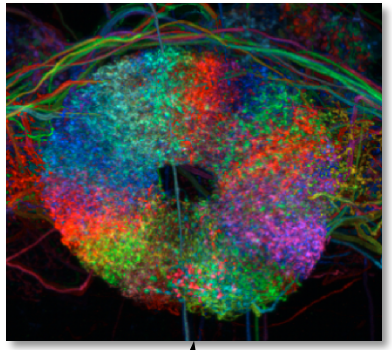
Walking trajectory before first food encounter

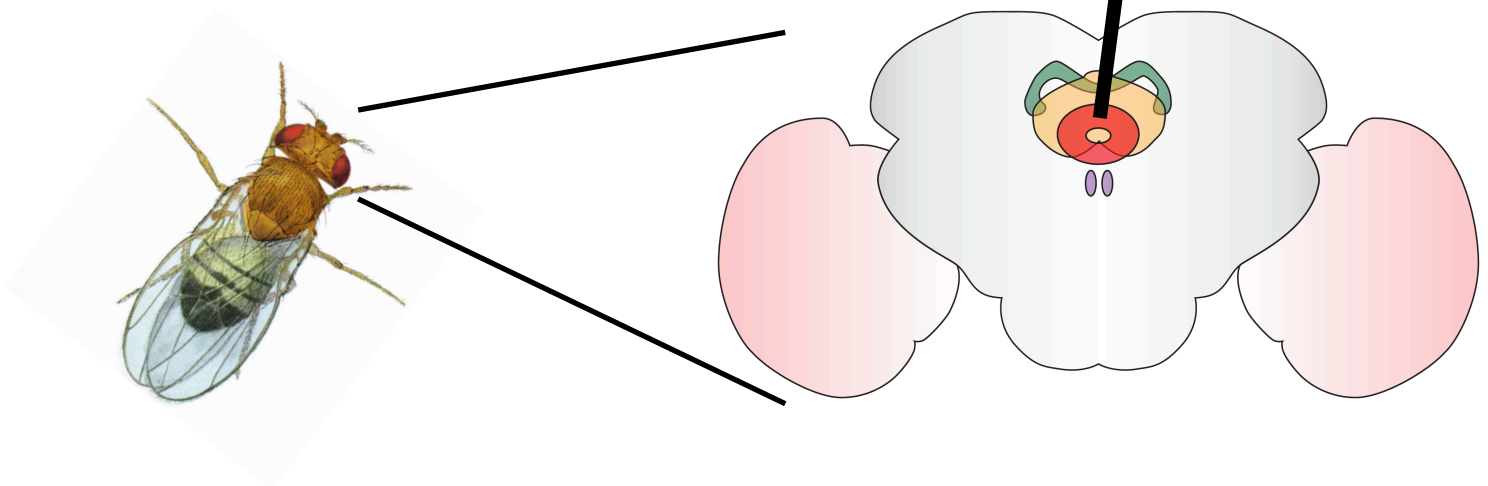


Kim & Dickinson (2017) *Current Biology*

Compass neurons

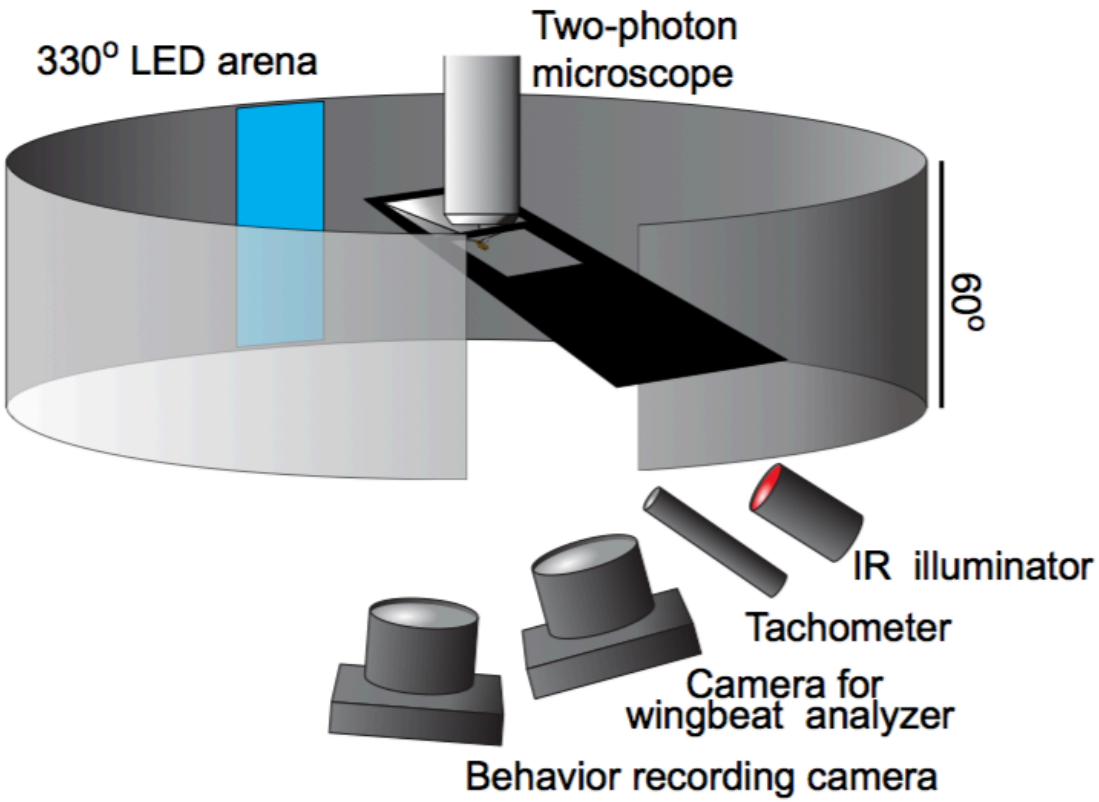


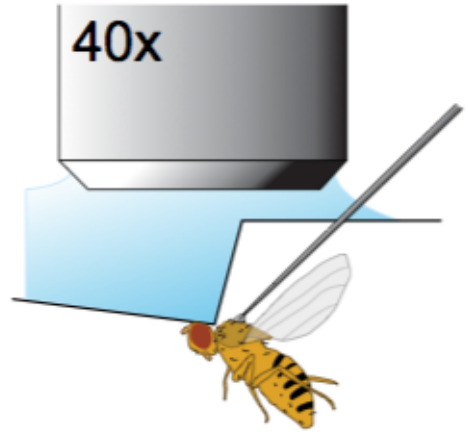


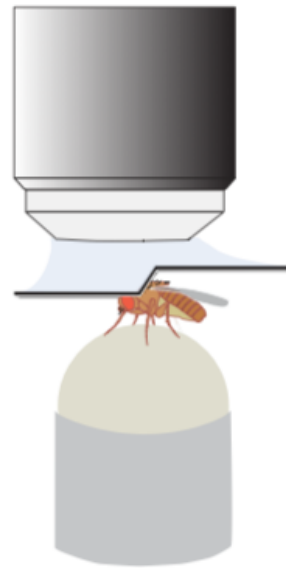


E-PG neuron (a.k.a. compass neurons) in Ellipsoid Body

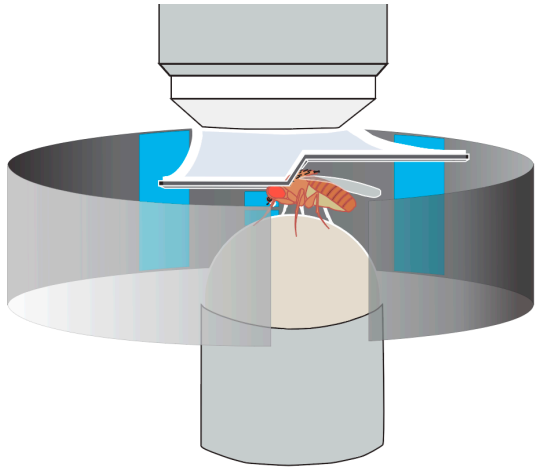
Jenett et al. (2012) *Cell Reports*
Wolff et al. (2015) *J Comp Neurol*


330° LED arena
Two-photon microscope
60°
IR illuminator
Tachometer
Camera for wingbeat analyzer
Behavior recording camera


40x



An unique representation of orientation



A fly is walking on an air-floated ball

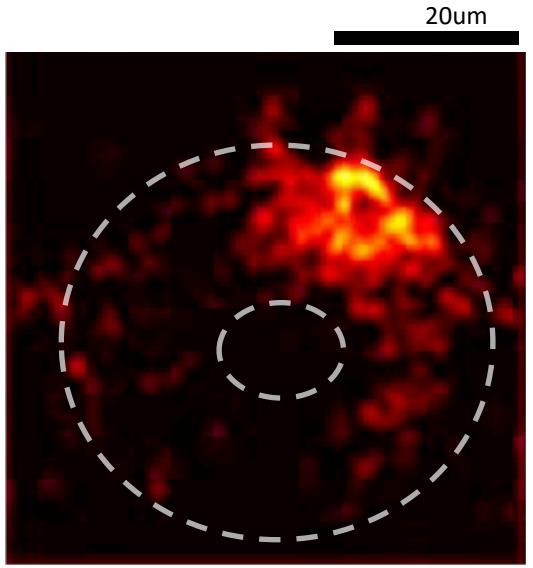
Visual arena

Visual scene on the VR screen that the fly is watching

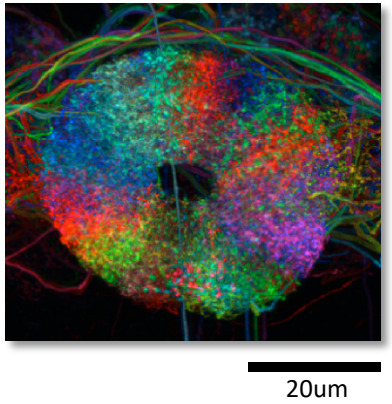
GCaMP is expressed in the entire population of compass neurons

Tethered fly

The rotation of the ball is measured to infer the fly's intention of turning



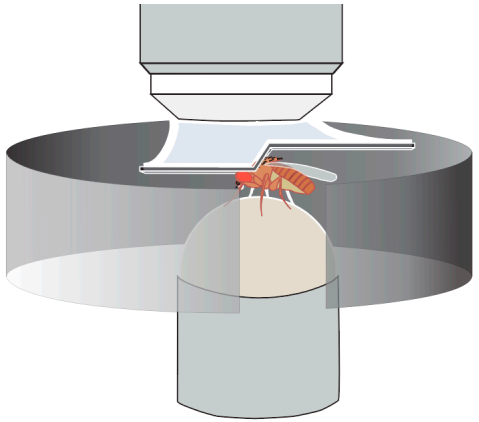
Calcium imaging of compass neurons with GCaMP

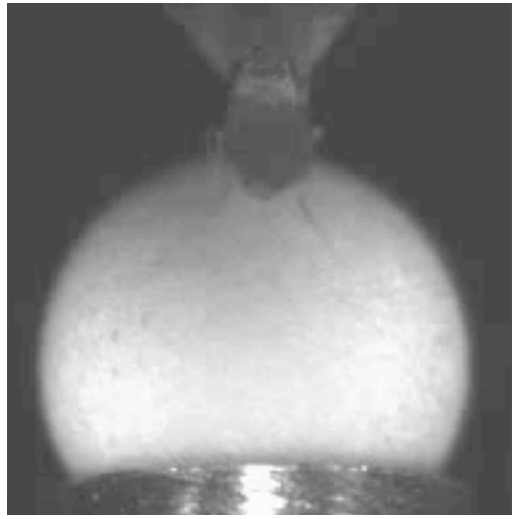


We are looking are the neural activity of the entire population of fly compass neurons

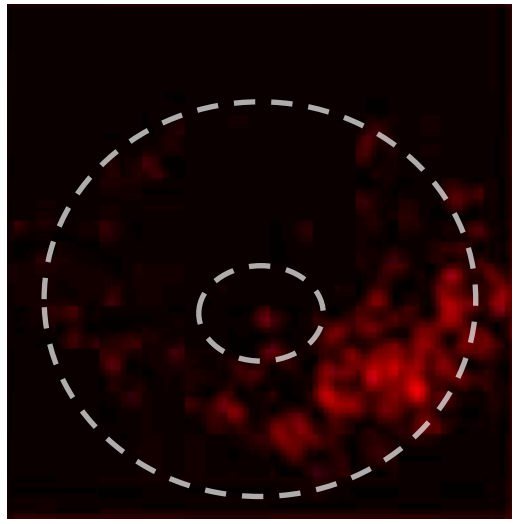
Seelig & Jayaraman (2015) *Nature*

Representation is maintained in darkness

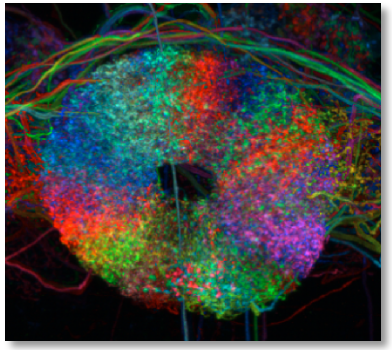




The fly is in a complete darkness.

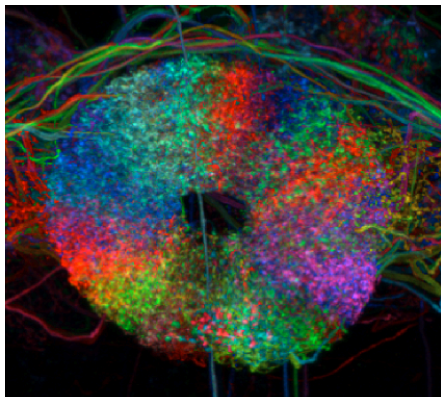


20um

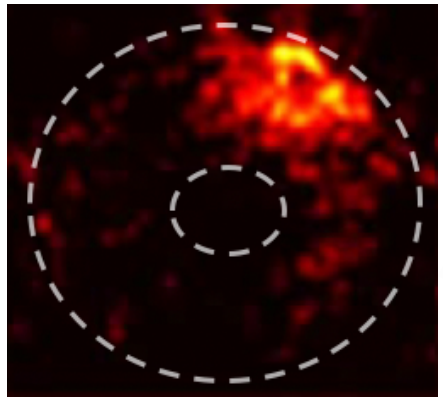


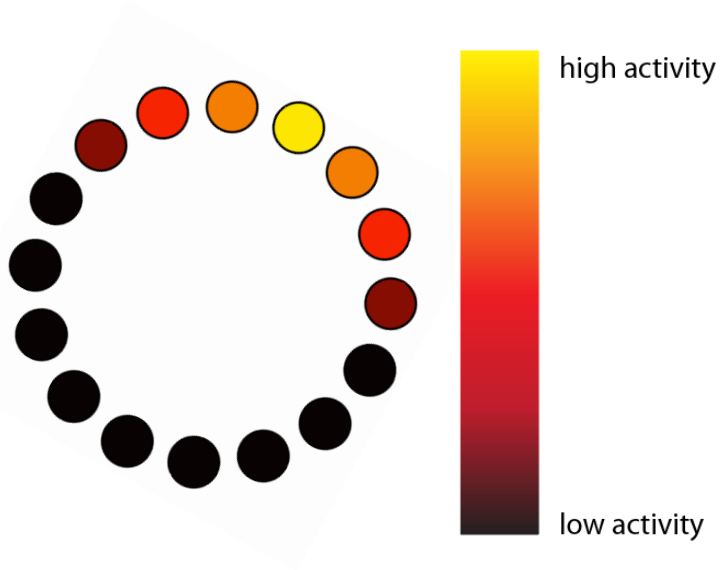
20um

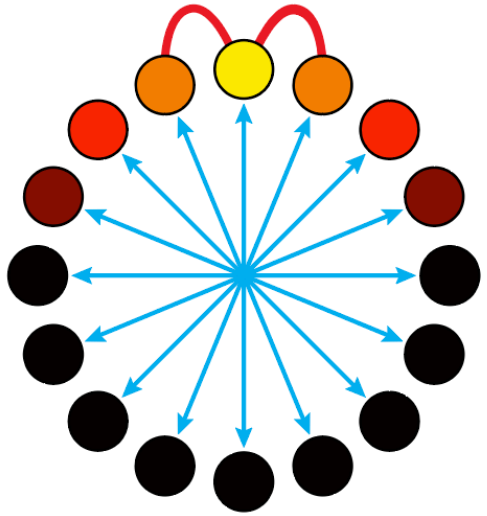
Seelig & Jayaraman (2015) *Nature*



20um

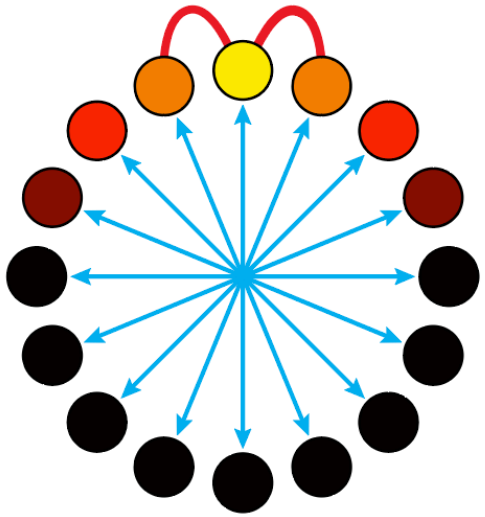






Ring Attractor

$$\tau \frac{\partial f_n}{\partial t} = -f_n + \left[\alpha f_n + D(f_{n+1} + f_{n-1} - 2f_n) - \beta \sum_{m=0}^{N-1} f_m + 1 \right]_+$$



Ring Attractor

$$\tau \frac{\partial f_n}{\partial t} = -f_n + \left[\alpha f_n + D(f_{n+1} + f_{n-1} - 2f_n) - \beta \sum_{m=0}^{N-1} f_m + 1 \right]_+$$

f_n is the activity of a compass neuron with an index, n

$N = 32$ is the number of compass neurons (i.e, there are 32 similar equations)

$[x]_+$ is a rectification function (that is, if $x < 0$, it becomes zero).

$\tau = 0.05$ is a decay time constant (if there is no input, the neuron's activity decays)

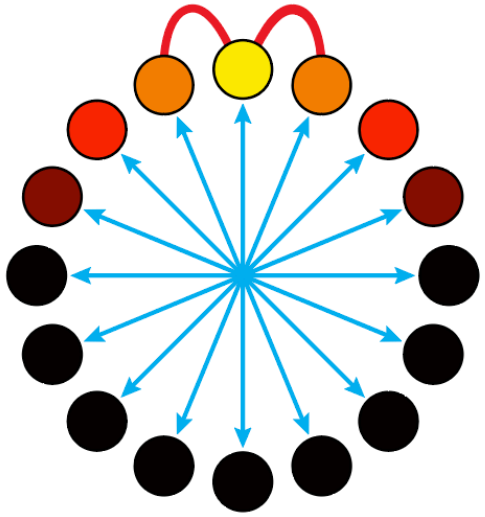
$\alpha = 2.62$ is the strength of self-recurrent connection

$\beta = 1.96$ is the strength of uniform inhibition

$D = 5.19$ is the strength of recurrent excitation between neighboring compass neurons

The constant term 1 is a uniform excitatory drive that generates activity in the system.

The initial value of f_n can be set `yinit = np.random.random(size=(32))*0.05`



Ring Attractor

$$\tau \frac{\partial f_n}{\partial t} = -f_n + \left[\alpha f_n + D(f_{n+1} + f_{n-1} - 2f_n) - \beta \sum_{m=0}^{N-1} f_m + 1 \right]_+$$

```
import numpy as np
from scipy.integrate import solve_ivp

def RingAttractorODEs(t,f):

    tau = 0.05
    alpha = 2.6833
    beta = 1.9835
    D = 5.187
    n_neurons = 32;

    f0, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, ..., f31 = f
```

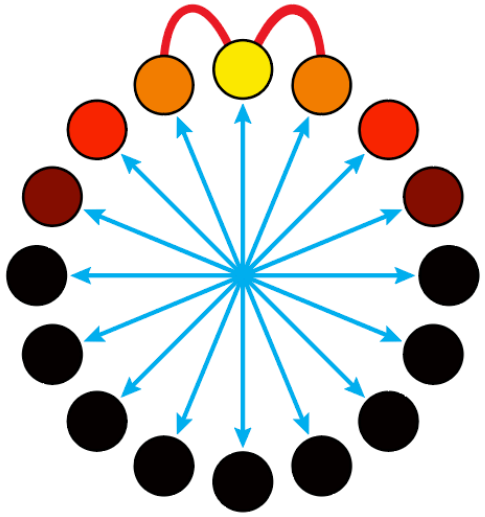
```
alpha*f   + D*(np.roll(f,-1) + np.roll(f,1) - 2*f)- beta*f.sum()   + 1
```

f: array([100, 200, 300, 400, 500, 600])

f_{n+1}: array([200, 300, 400, 500, 600, 100])

f_{n-1}: array([600, 100, 200, 300, 400, 500])

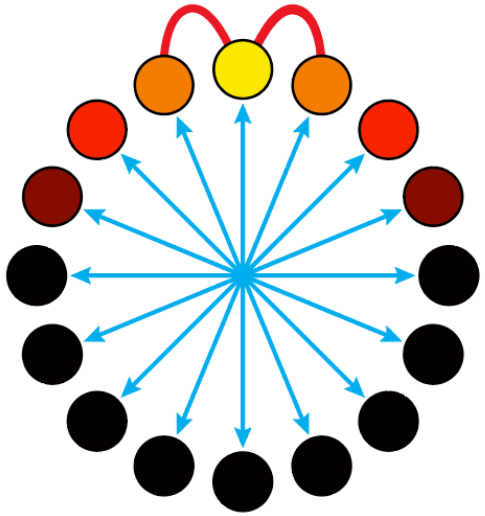
```
f_np1 = f[  list(range(1,n_neurons)) + [0]    ]
f_nm1 = f[  [-1] + list(range(n_neurons-1))   ]

D*(f_np1+f_nm1-2*f)
```



Ring Attractor

$$\tau \frac{\partial f_n}{\partial t} = -f_n + \left[\alpha f_n + D(f_{n+1} + f_{n-1} - 2f_n) - \beta \sum_{m=0}^{N-1} f_m + 1 \right]_+$$

```
import numpy as np
from scipy.integrate import solve_ivp

def RingAttractorODEs(t,f):

    tau = 0.05
    alpha = 2.6183
    beta = 1.9635
    D = 5.1876
    n_neurons = 32;

    tmp = alpha*f + D*(np.roll(f,-1) + np.roll(f,1) - 2*f) - beta*f.sum()  + 1
    tmp[tmp<0] = 0

    dfdt = (  -f + tmp )/tau

    return dfdt
```



Ring Attractor

$$\tau \frac{\partial f_n}{\partial t} = -f_n + \left[\alpha f_n + D(f_{n+1} + f_{n-1} - 2f_n) - \beta \sum_{m=0}^{N-1} f_m + 1 \right]_+$$

```
import numpy as np
from scipy.integrate import solve_ivp

def RingAttractorODEs(t,f):
        ......

tspan = np.linspace(0, 1, 100)
f_init = np.random.random(size=(32,))*0.08

sol = solve_ivp(RingAttractorODEs,
                [tspan[0], tspan[-1]], f_init, t_eval=tspan)
```

```
import matplotlib.pyplot as plt
plt.figure(1, figsize=(12,10))
plt.imshow(sol.y, extent=[0,tspan[-1],32.5,0.5], aspect=tspan[-1]/32/8)
```