Decoding/ Machine learning

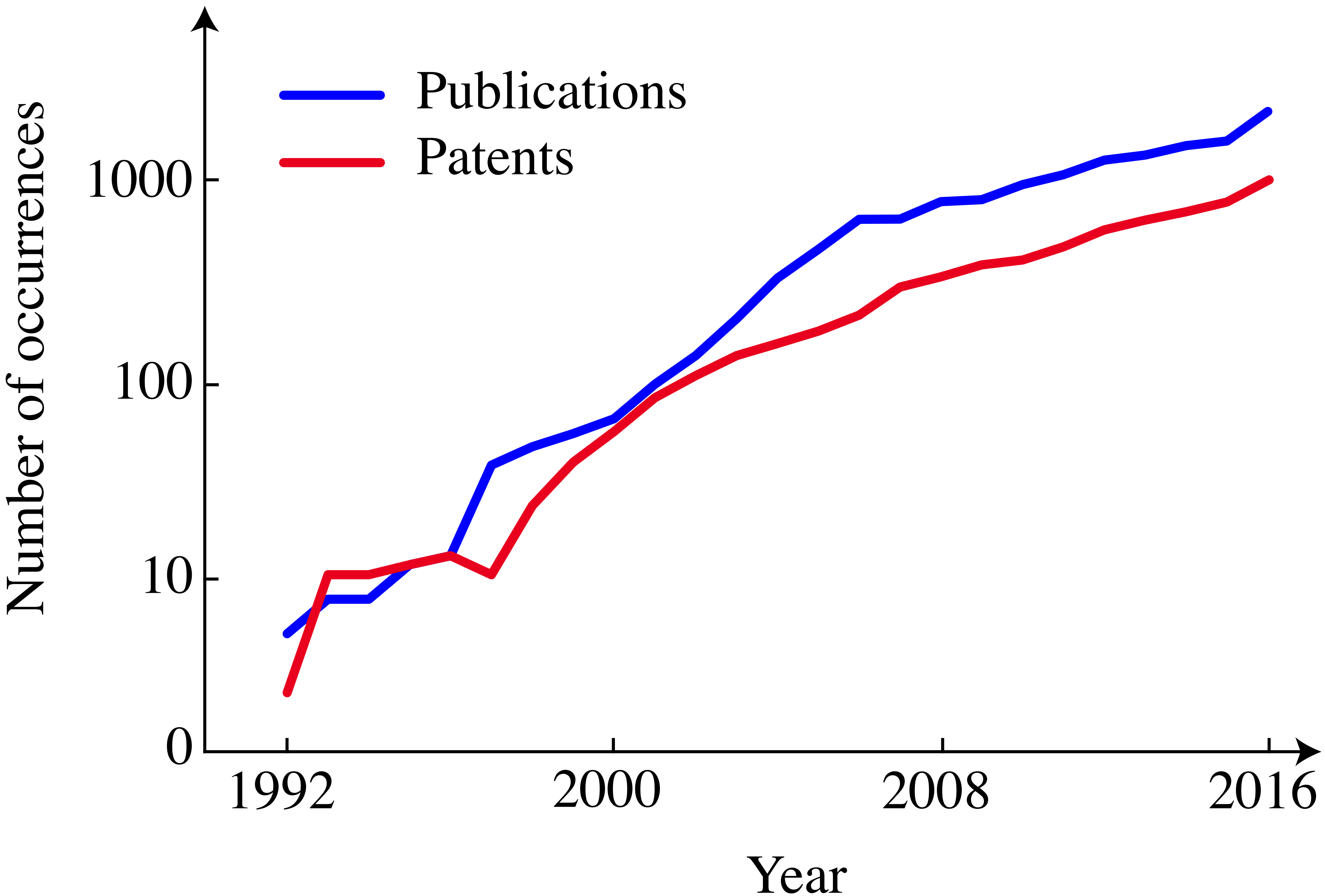
@kordinglab

Shameless plug: Please read *10 simple rules for structuring papers*

Outline

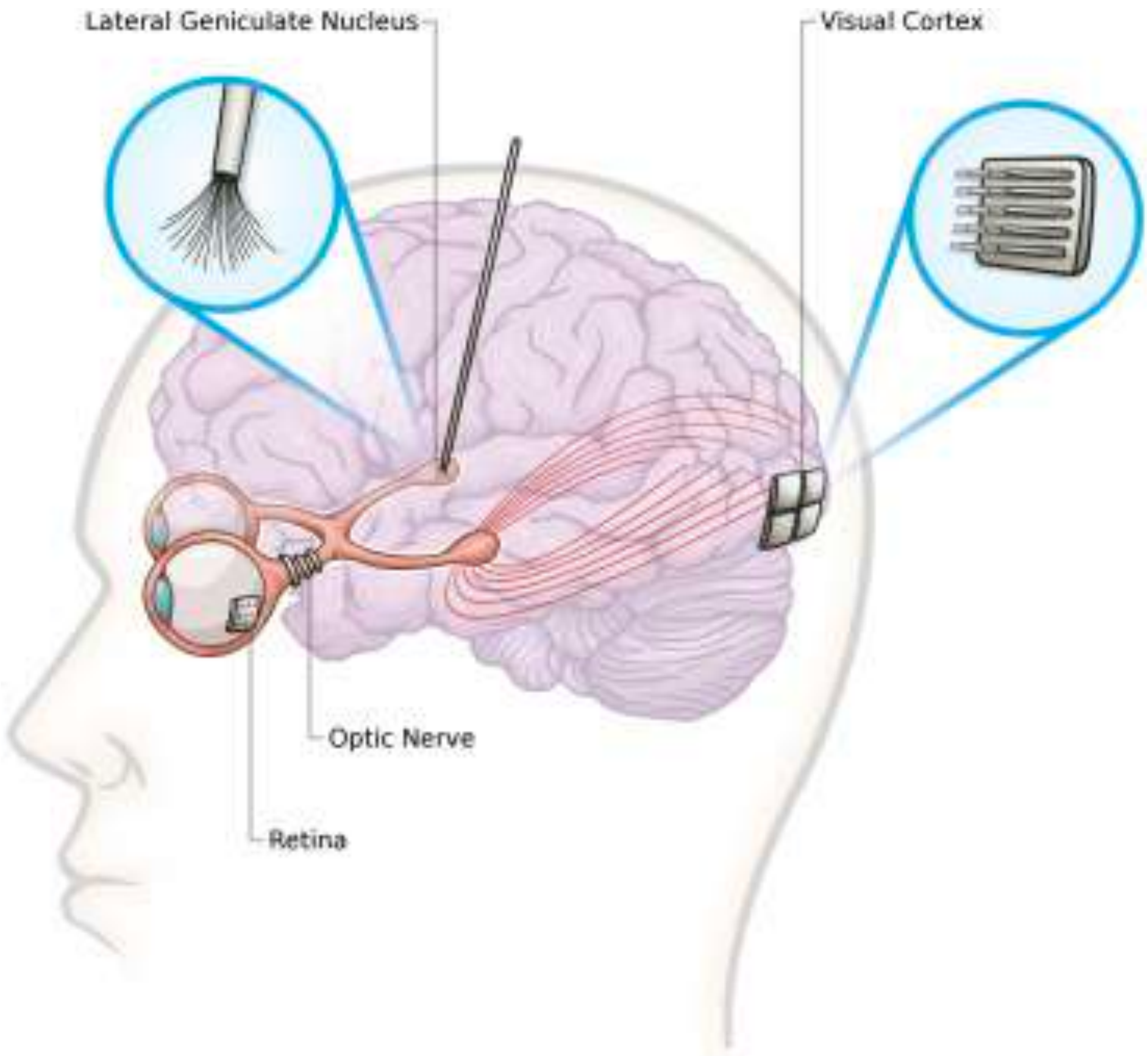
- 0) Why decoding/ML
- I) Overfitting
- II) Crossvalidation
- III) Regularization
- IV) RNNs ftw
- V) Which methods to use and when

O: ML is getting popular in biomedical science

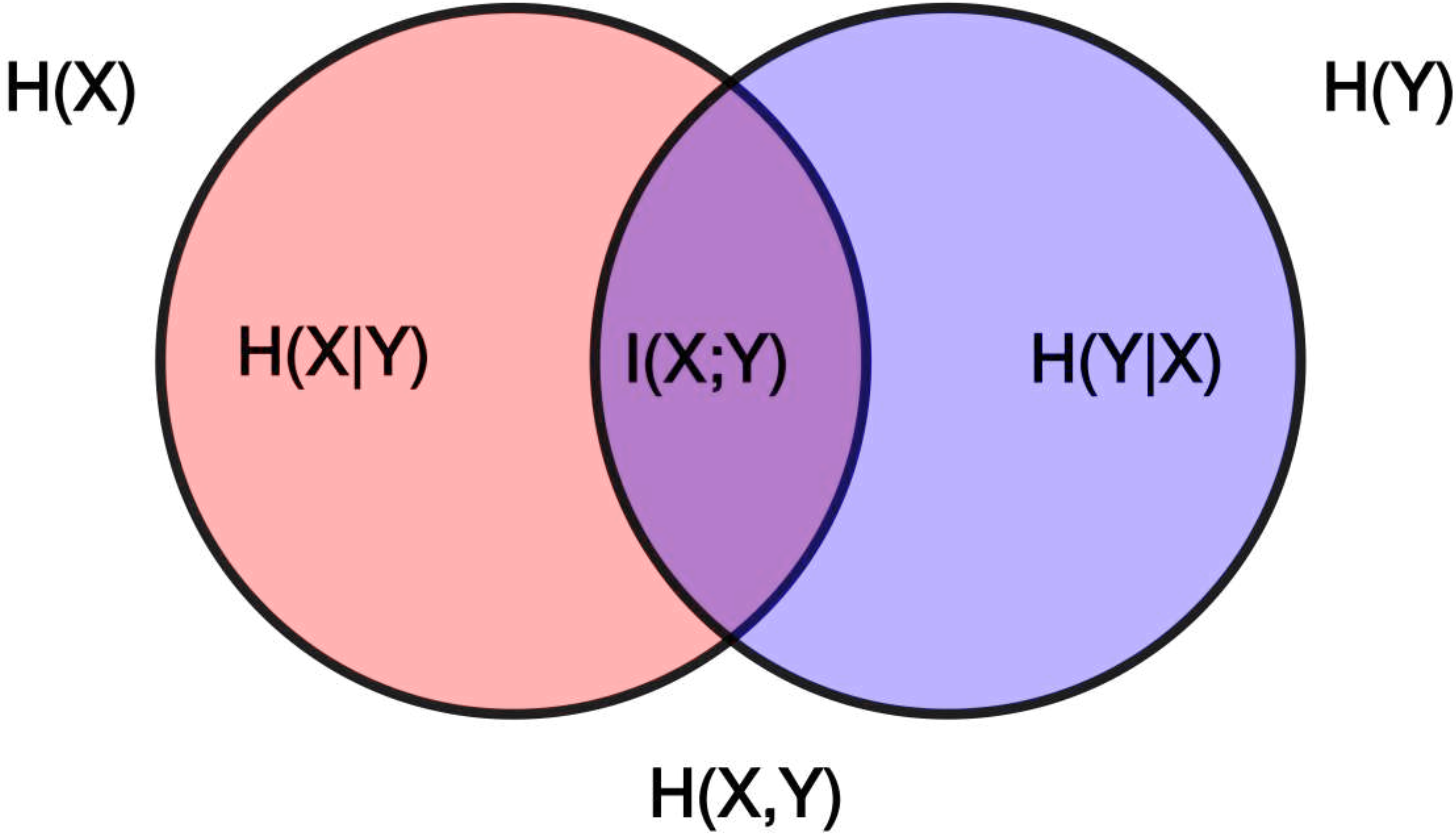


A) Solve engineering problems

Encoding: Cure blindness

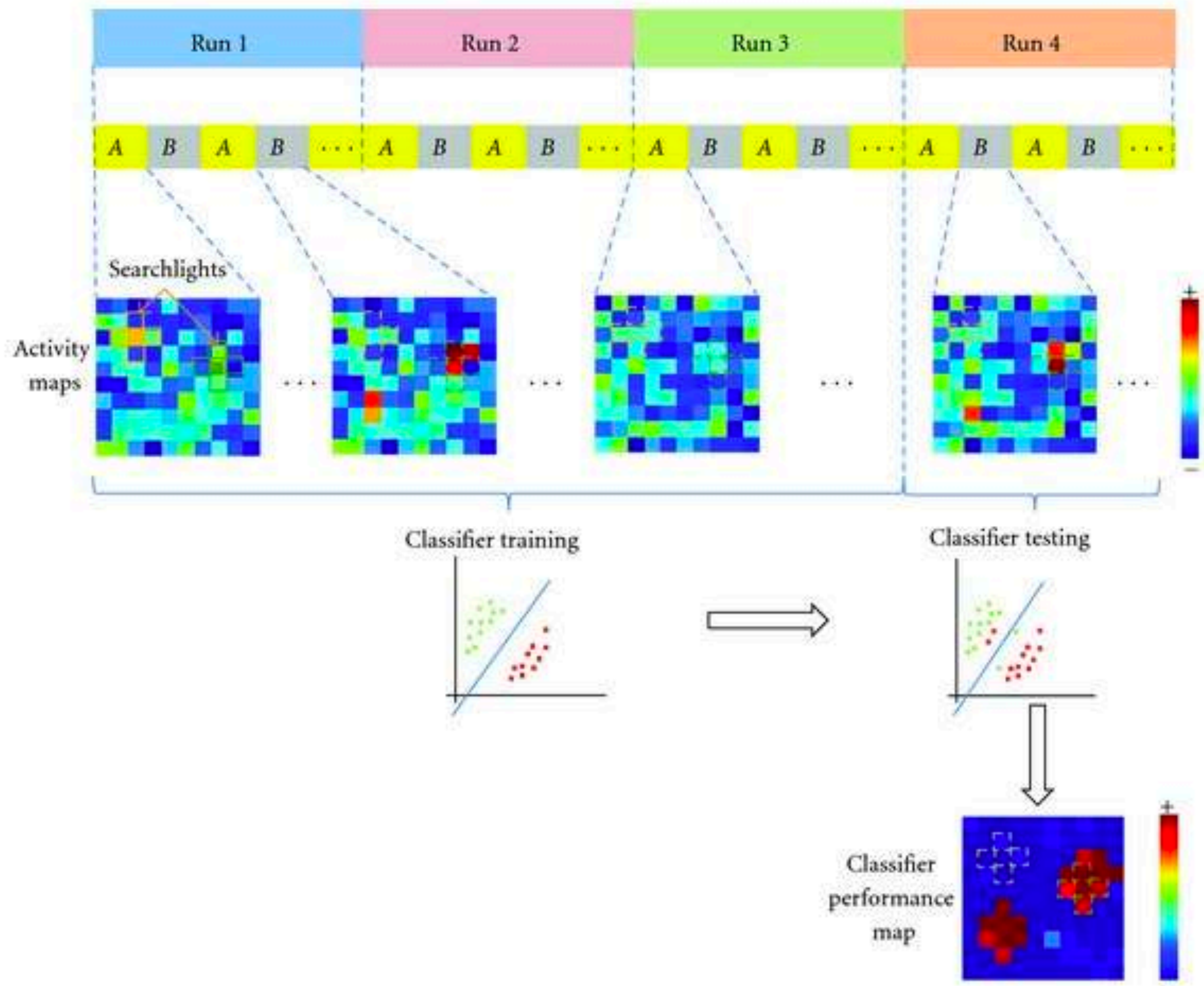


B) Understand data



e.g. Bialek

Example: MVPA



From: Mahmoudi et al

C) Provide a benchmark

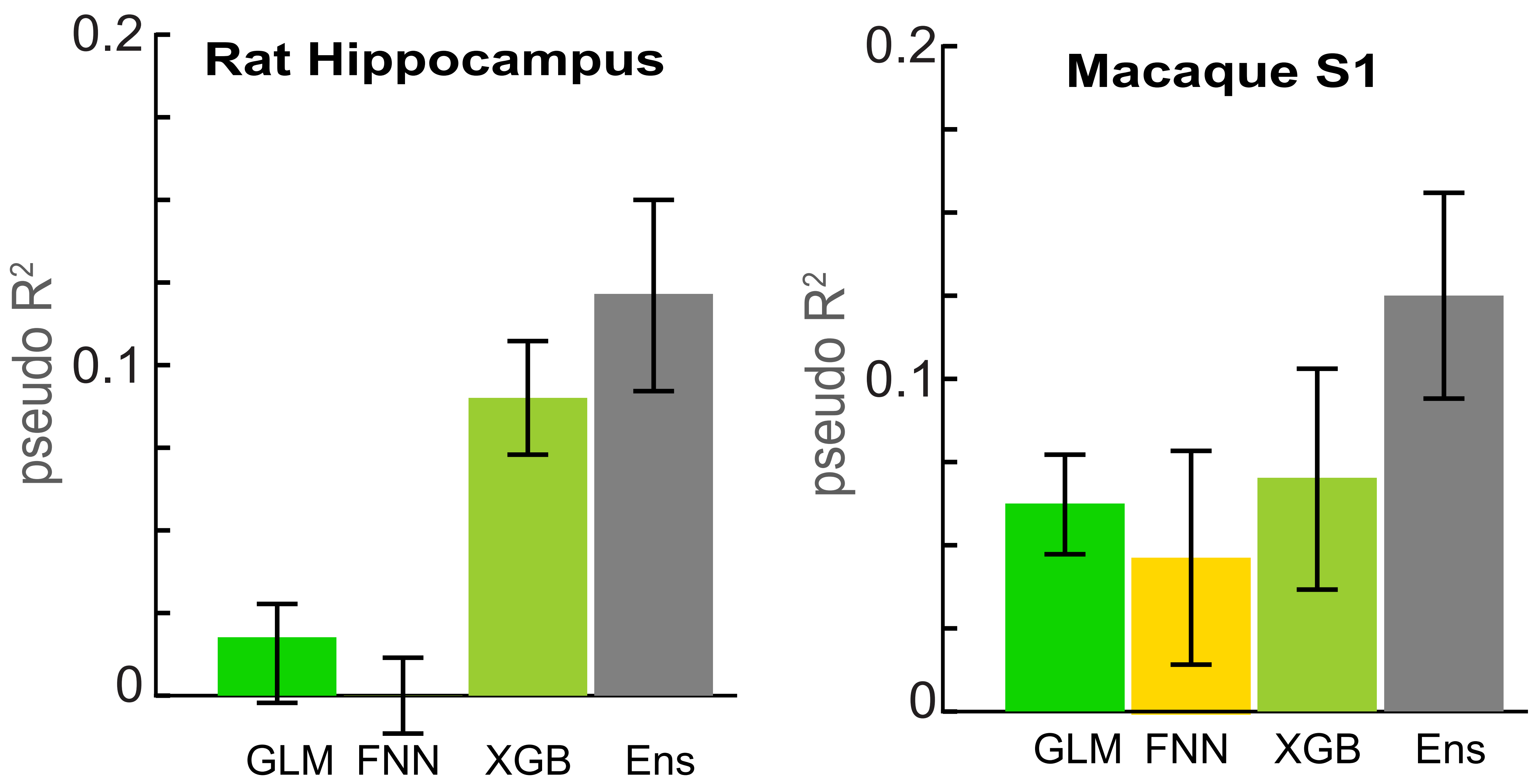
Machine learning

Your Model

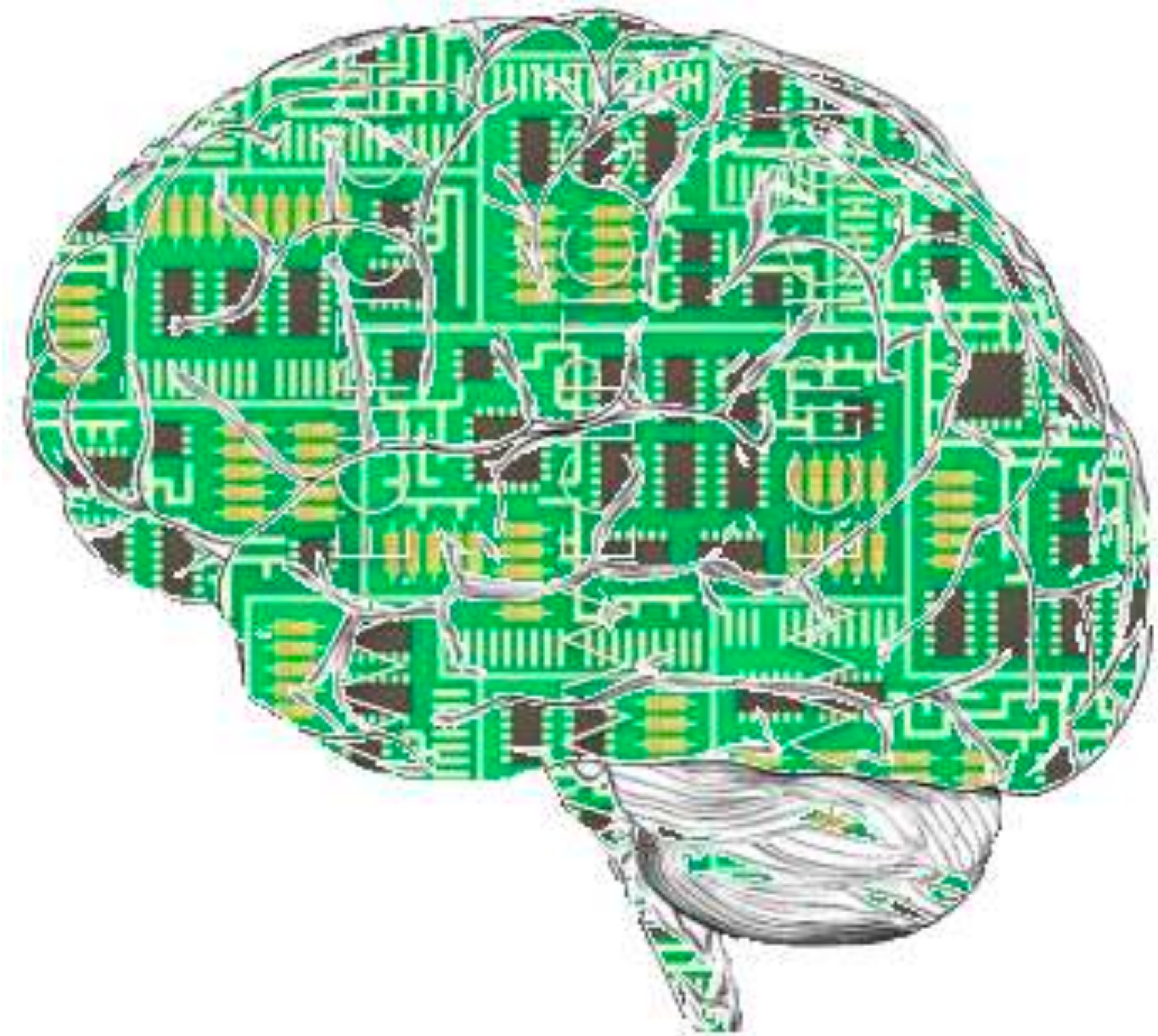
Being better than another model does not make a model true.

See Jonas and Kording, Could a neuroscientist understand a microprocessor 2017

How to think of GLMs



D) Model for brain

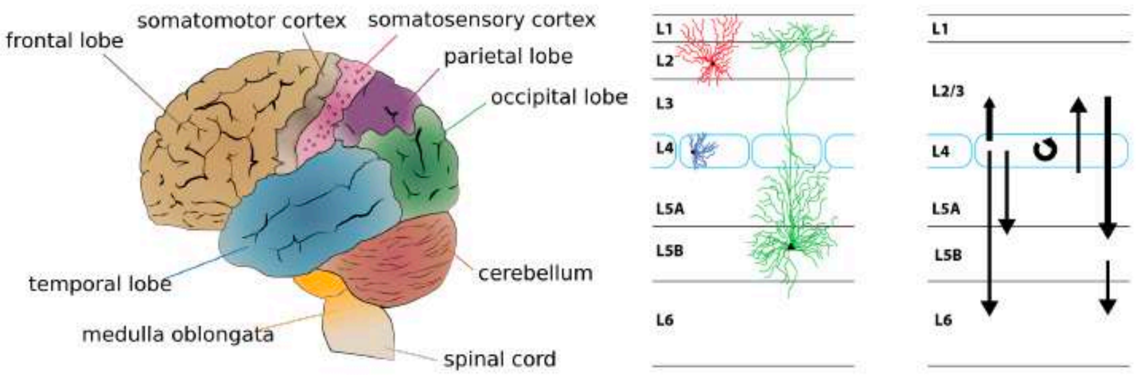


see Marblestone, Wayne, Kording, 2017

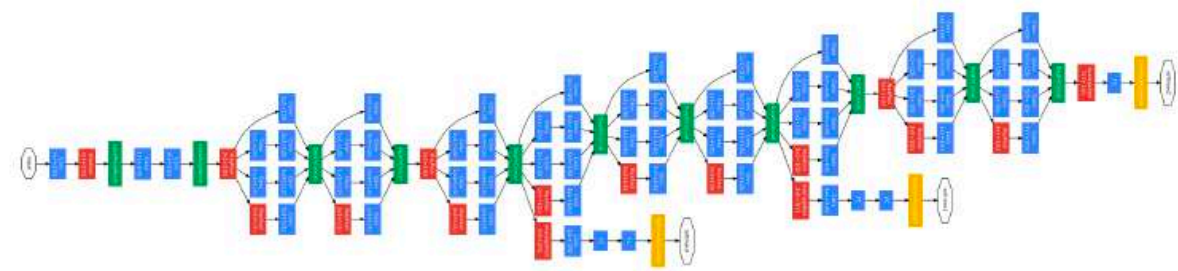
Systems Neuroscience

Machine Learning

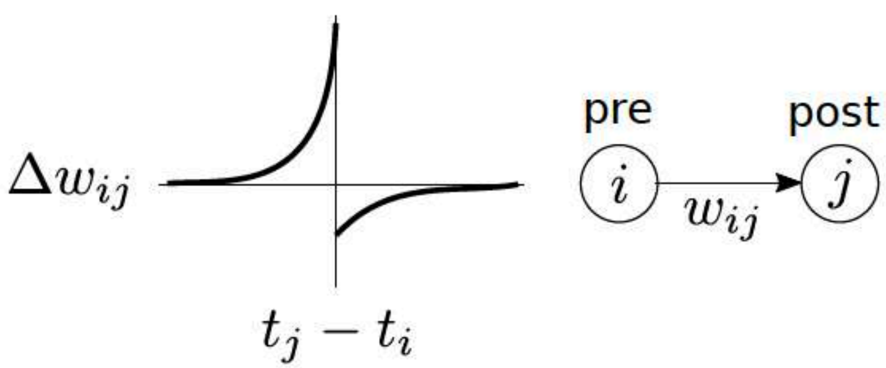
Anatomy:



Architecture:



Plasticity Rules:



Learning Rules:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{x_i}; \mathbf{w})$$

Tutorial

Always simulate data first

- Big things, whole organisms
- Medium things, groups of connected neurons
- Small things, say linear generators

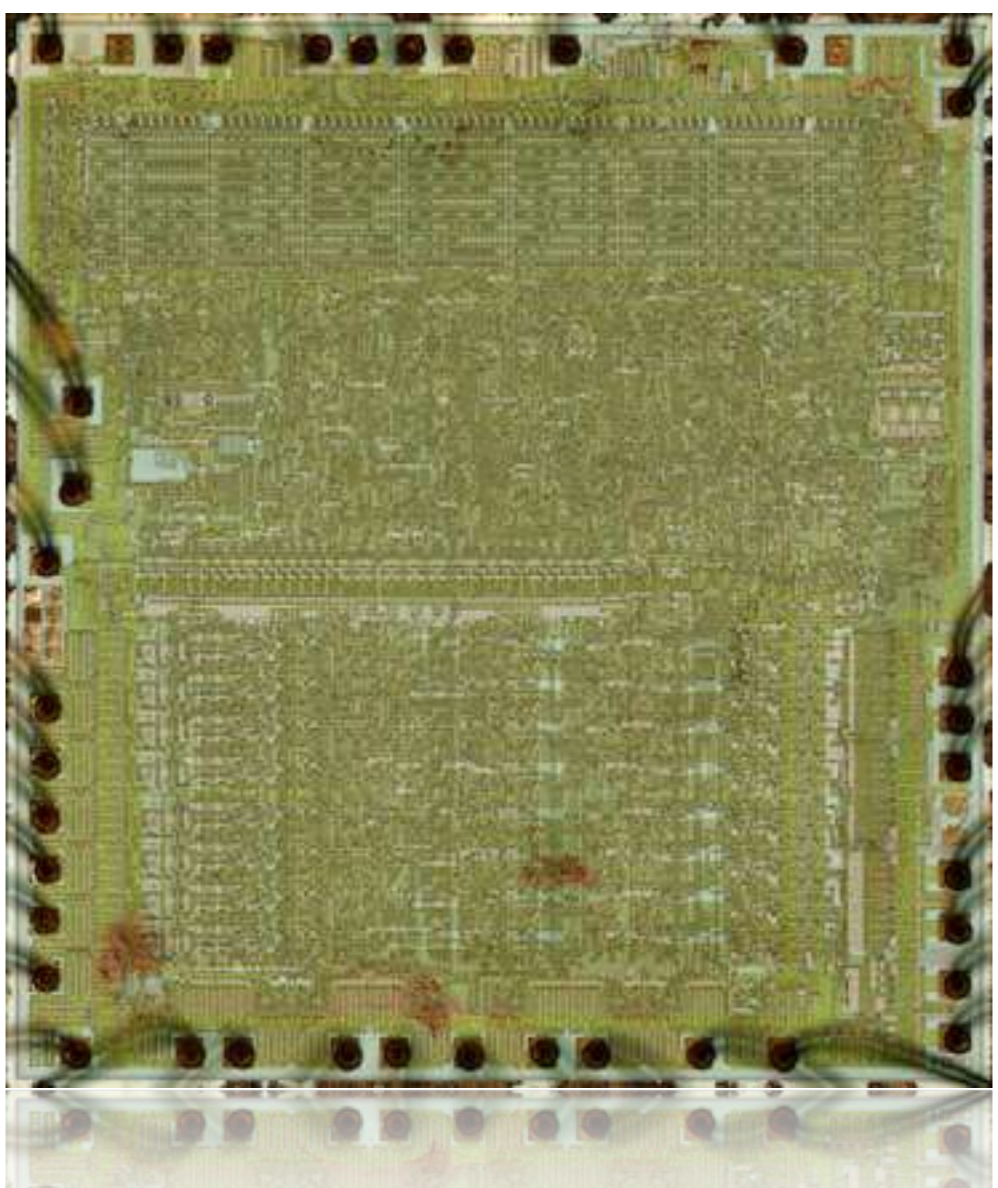
Big: organism

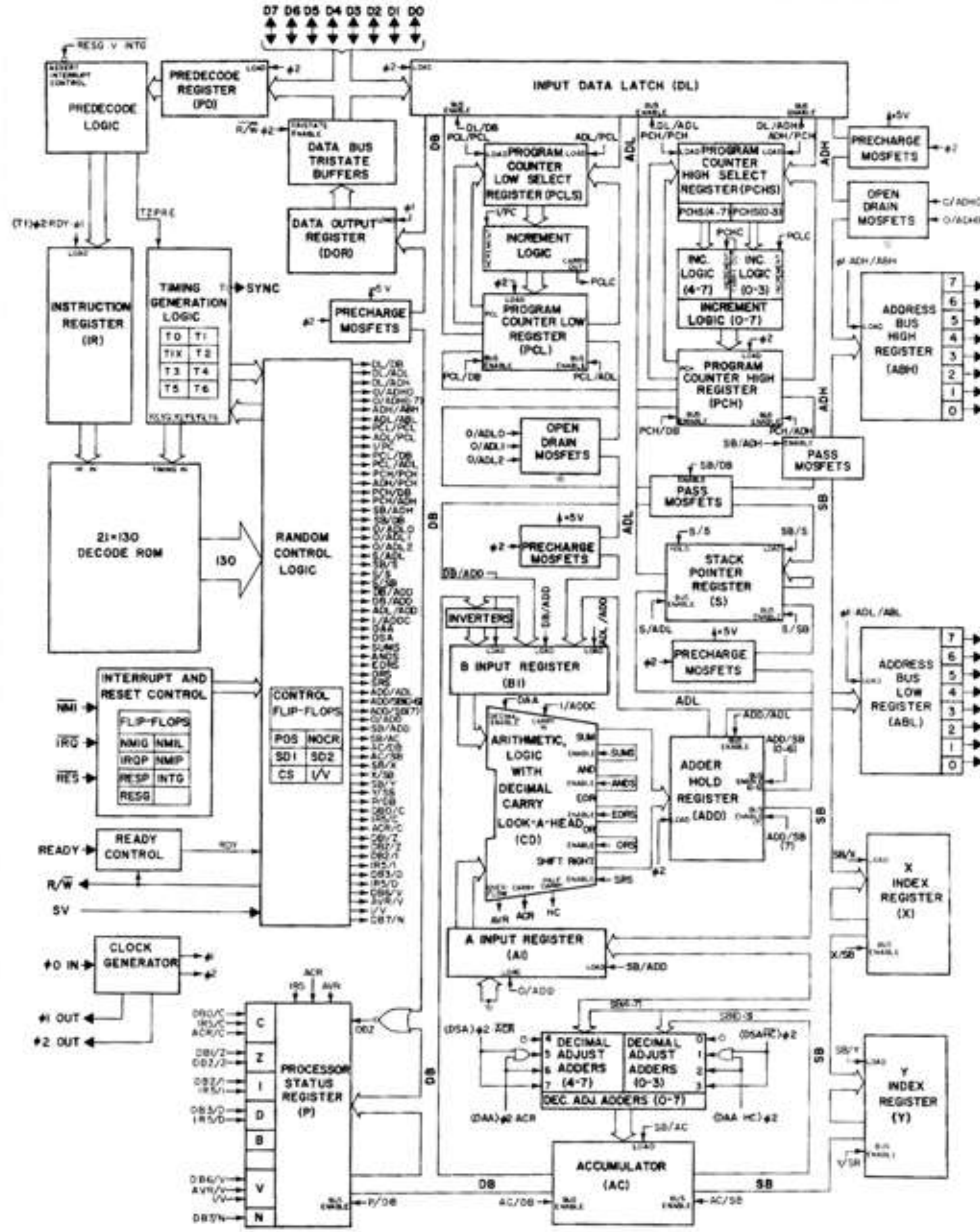
- Is neuroscience on the right path?

What is understanding?

- Fix whatever is wrong
- Simulate it
- Marr levels
  - Computational
  - Algorithmic
  - Mechanistic

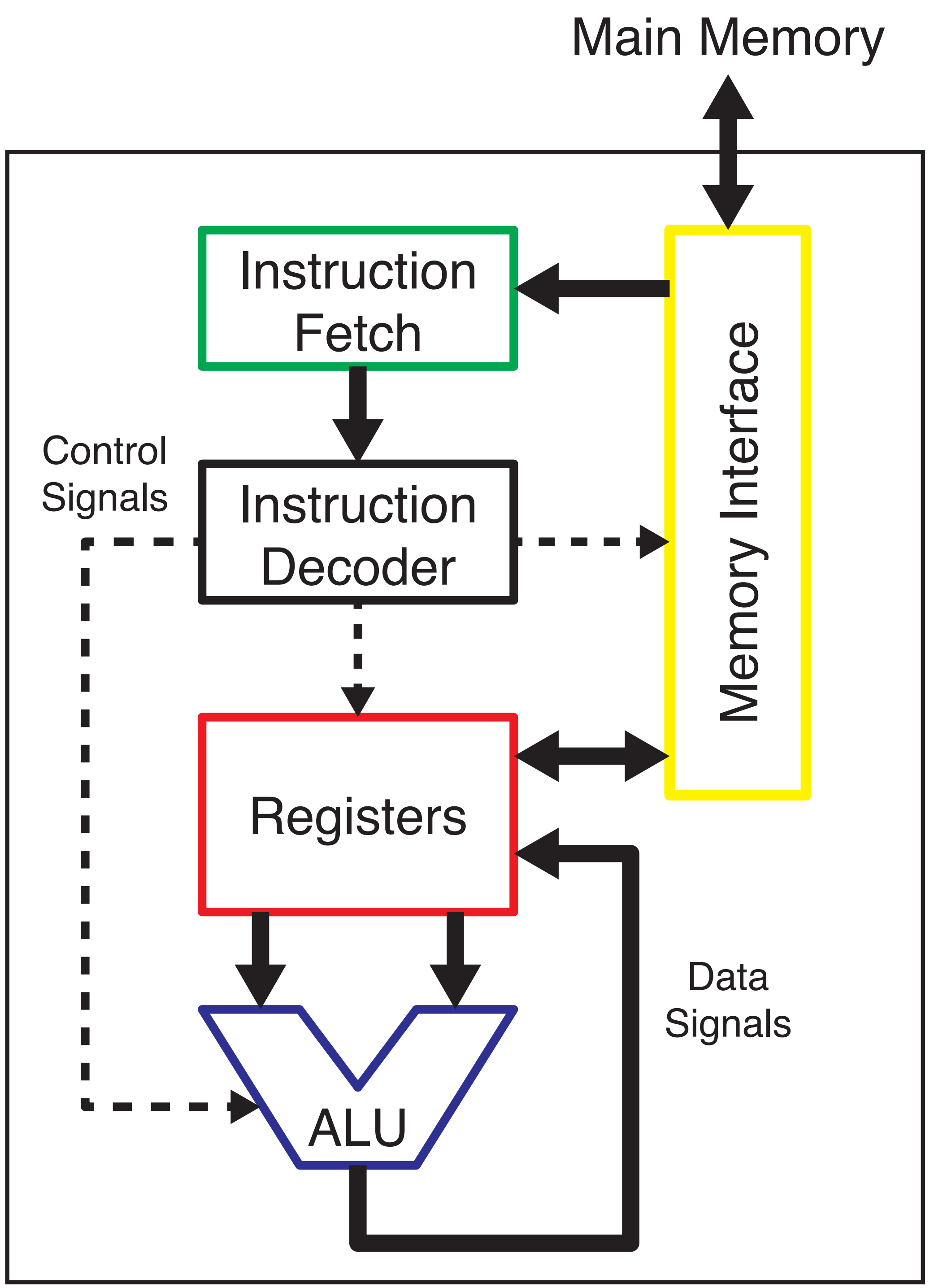
MOS 6502





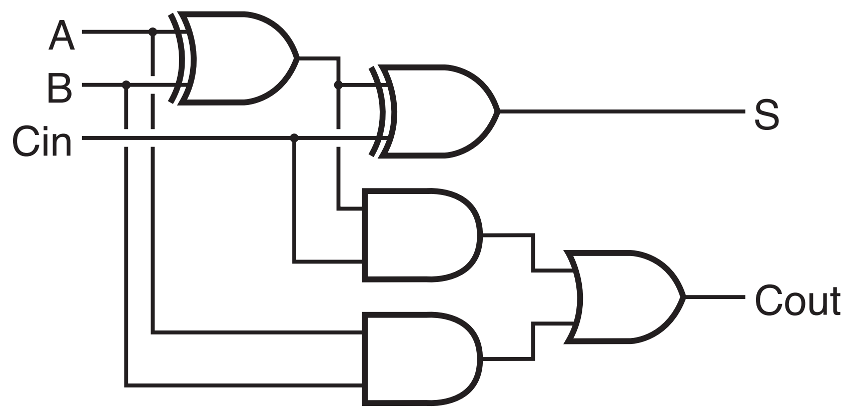
Courtesy http://visual6502.org

How it actually works

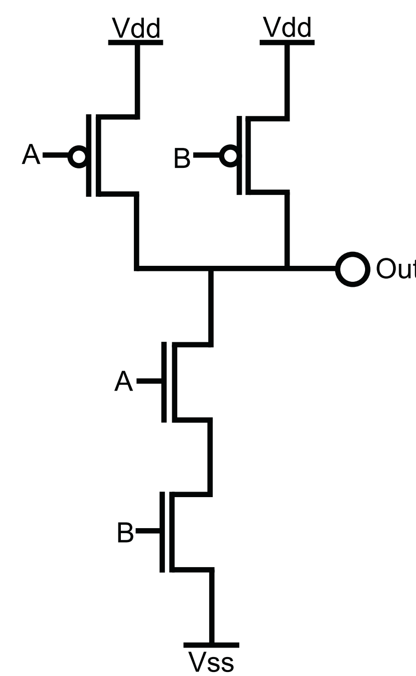


Multi scale

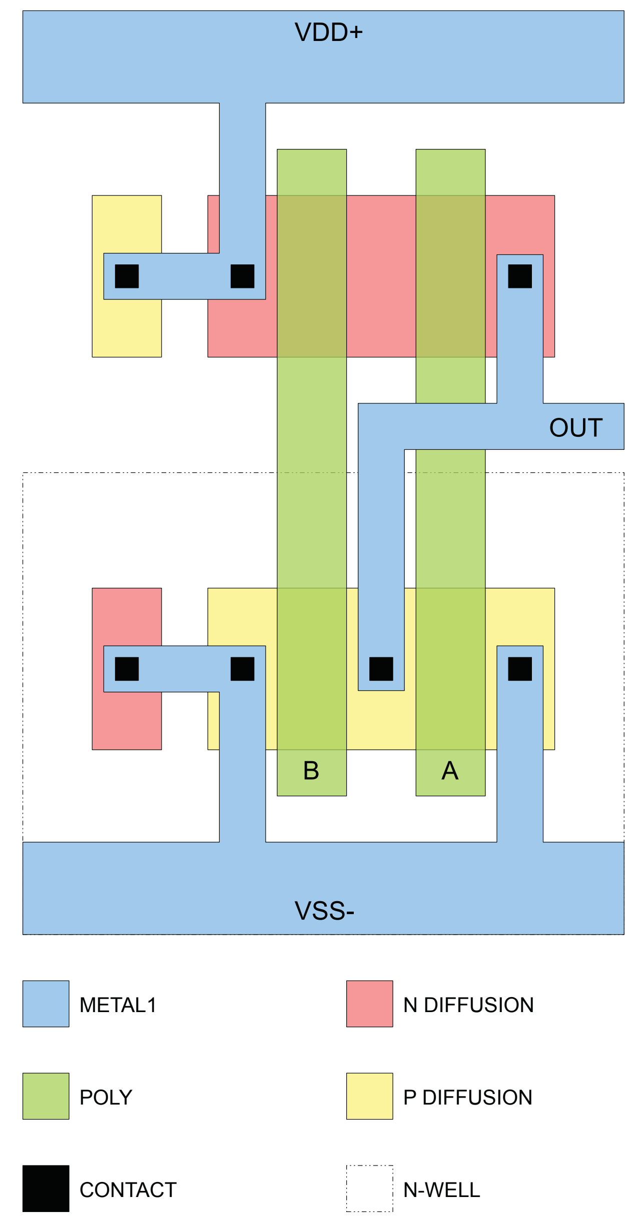
1-bit Adder



AND gate



AND gate (silicon)



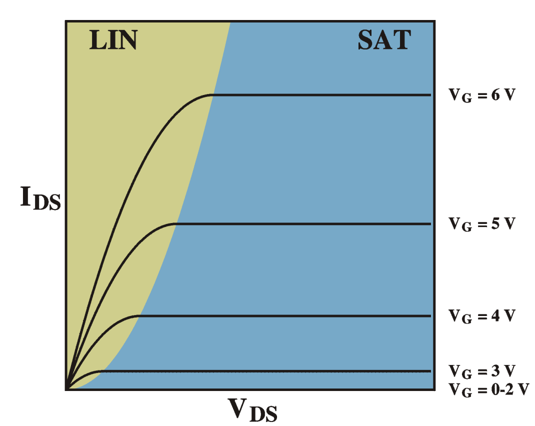
logic gate primitives

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

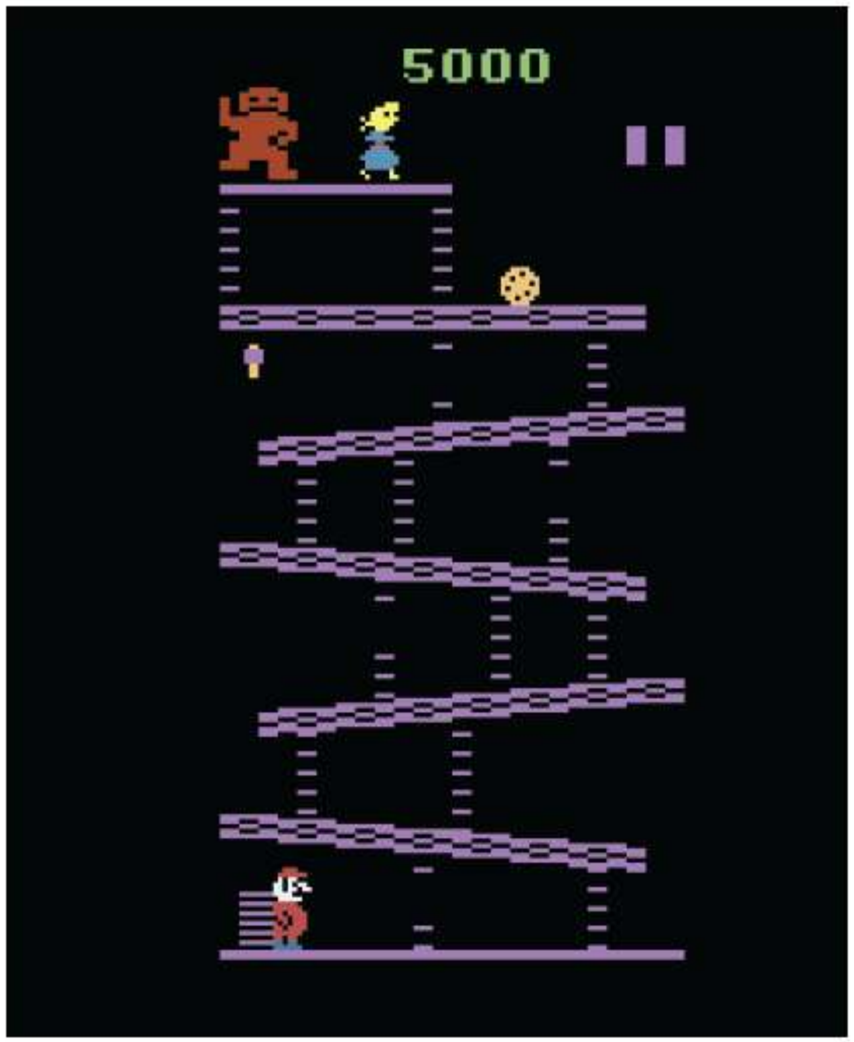
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

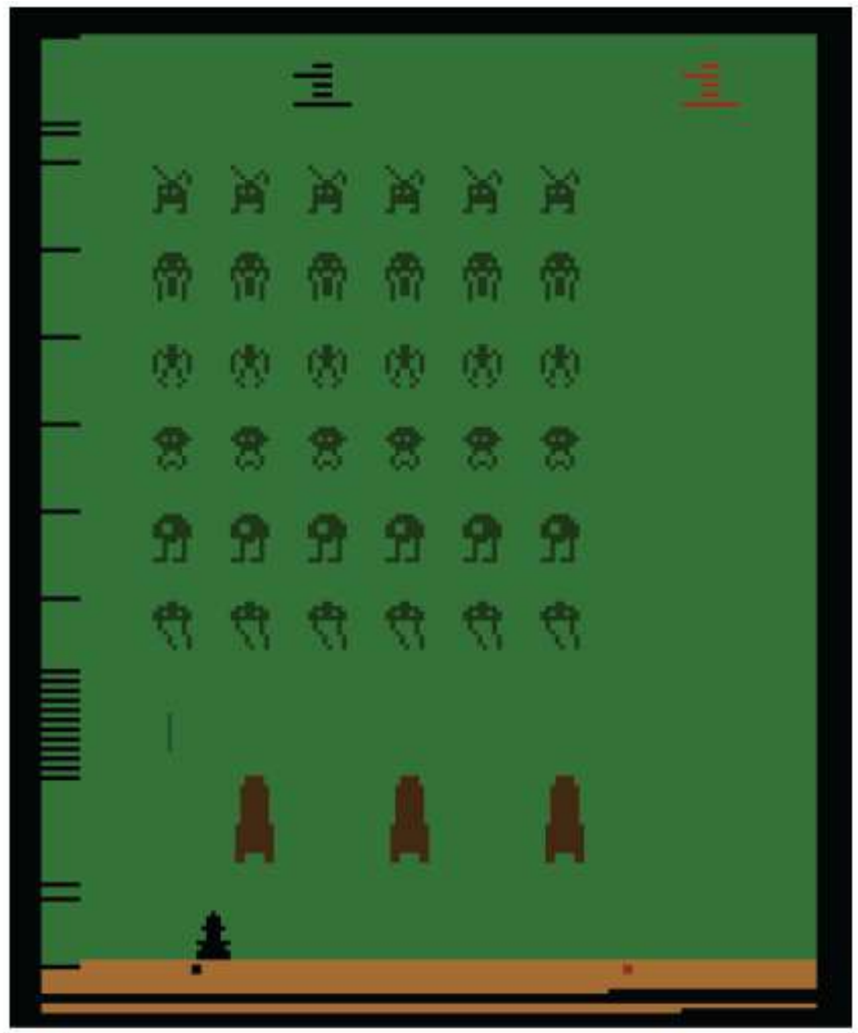
I/V for single gate



3 Behaviors



a. Donkey Kong (DK)

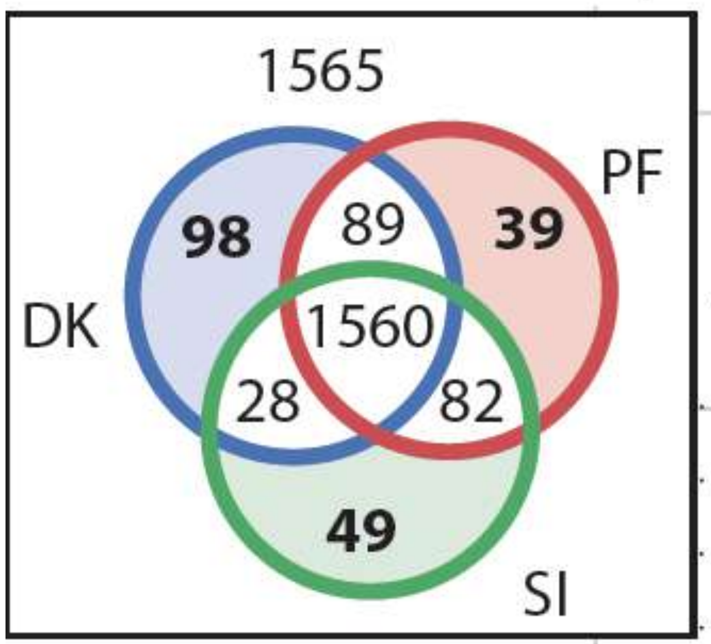


b. Space Invaders (SI)

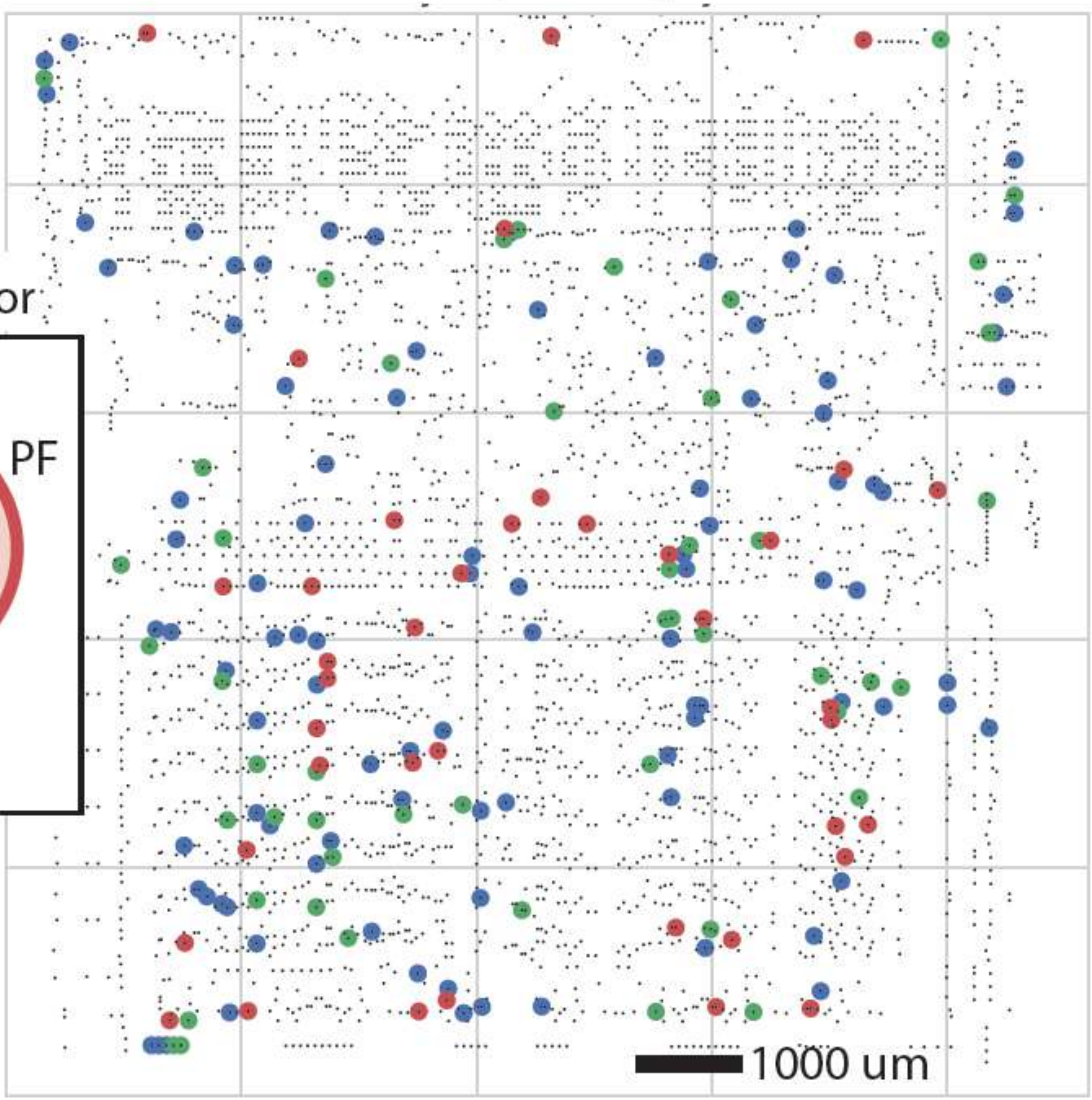
c. Pitfall (PF)

Lesion studies

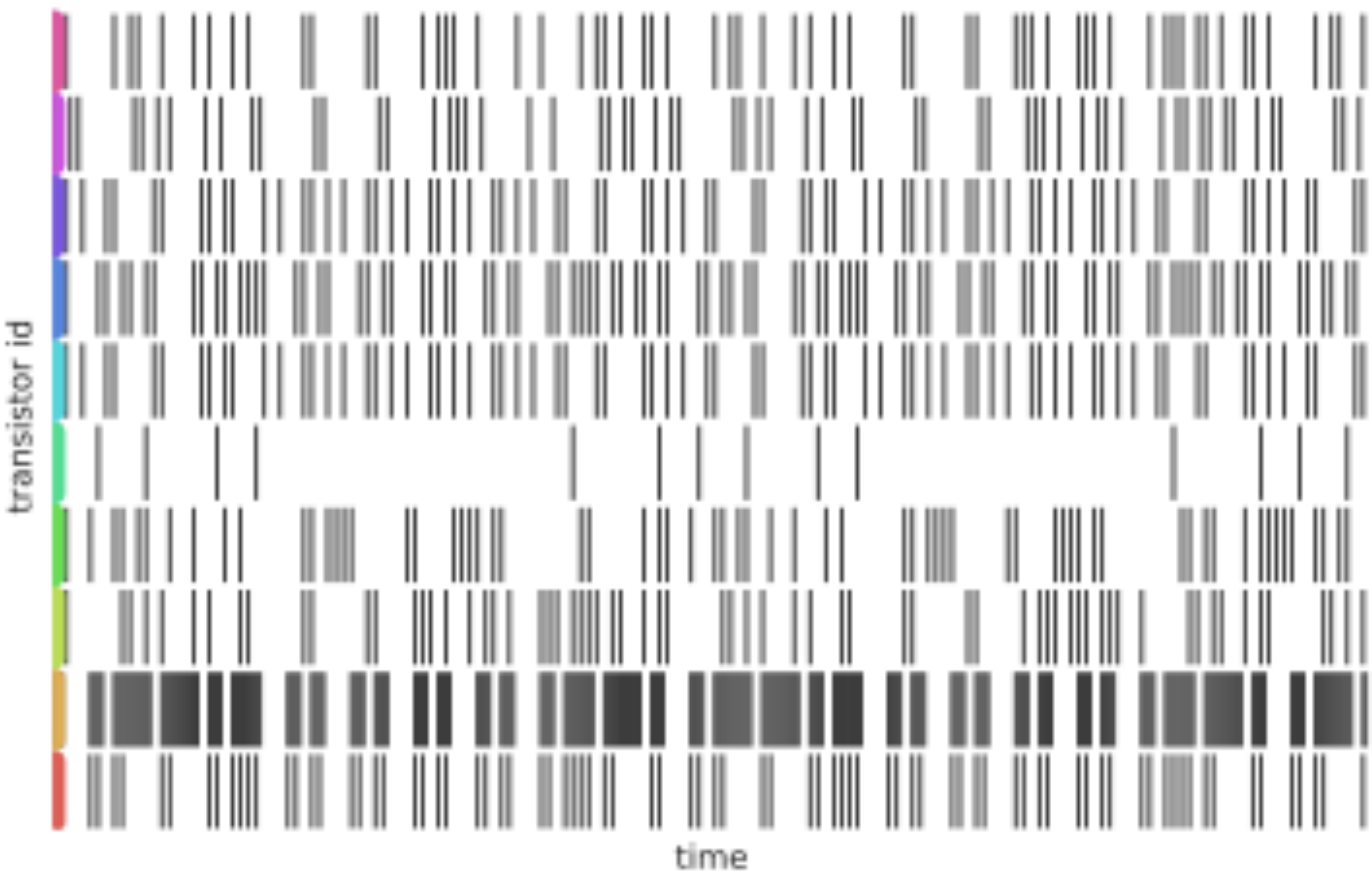
Lesion site vs behavior



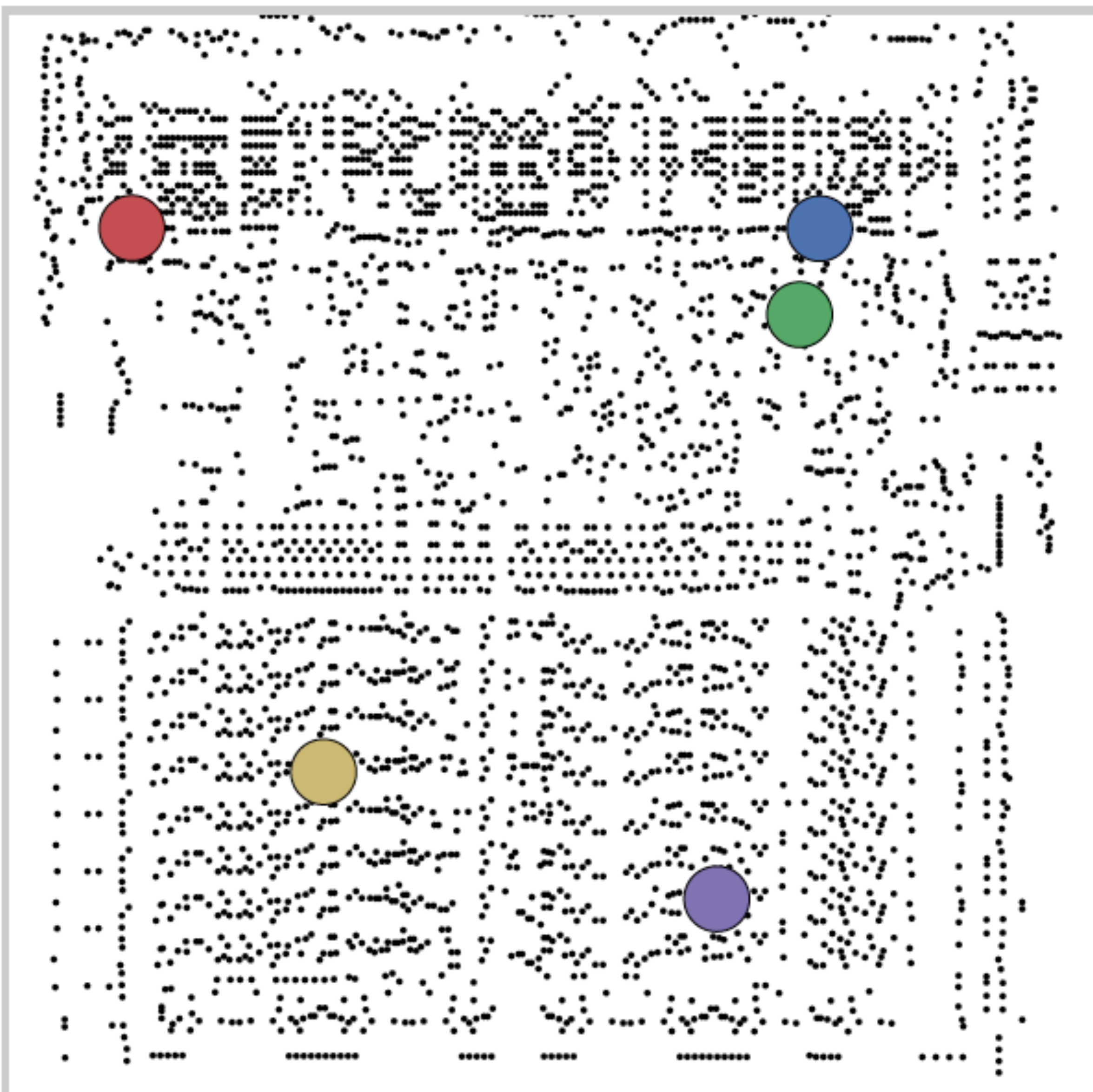
Lesions which impact single behavior

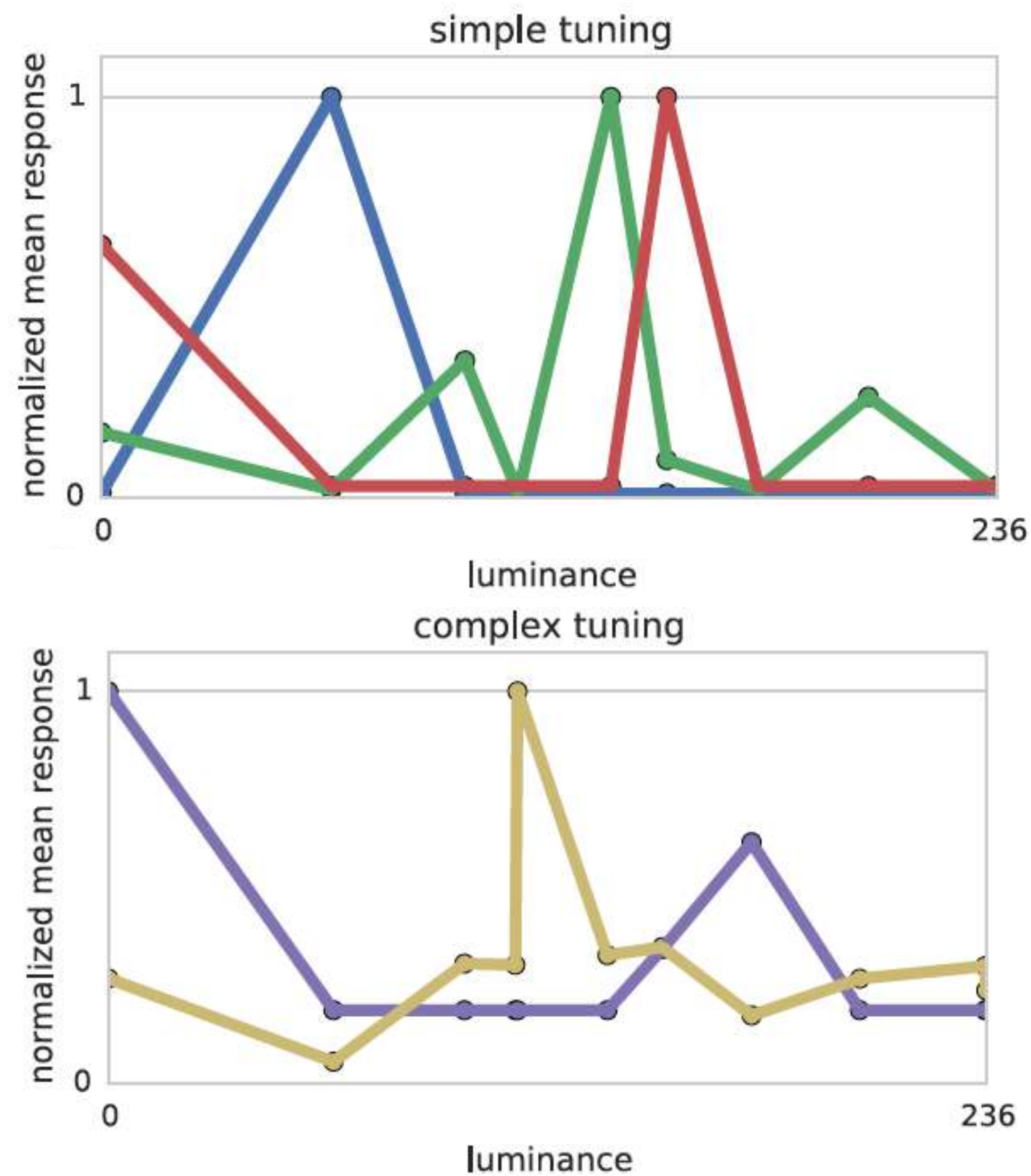


“Spike data”

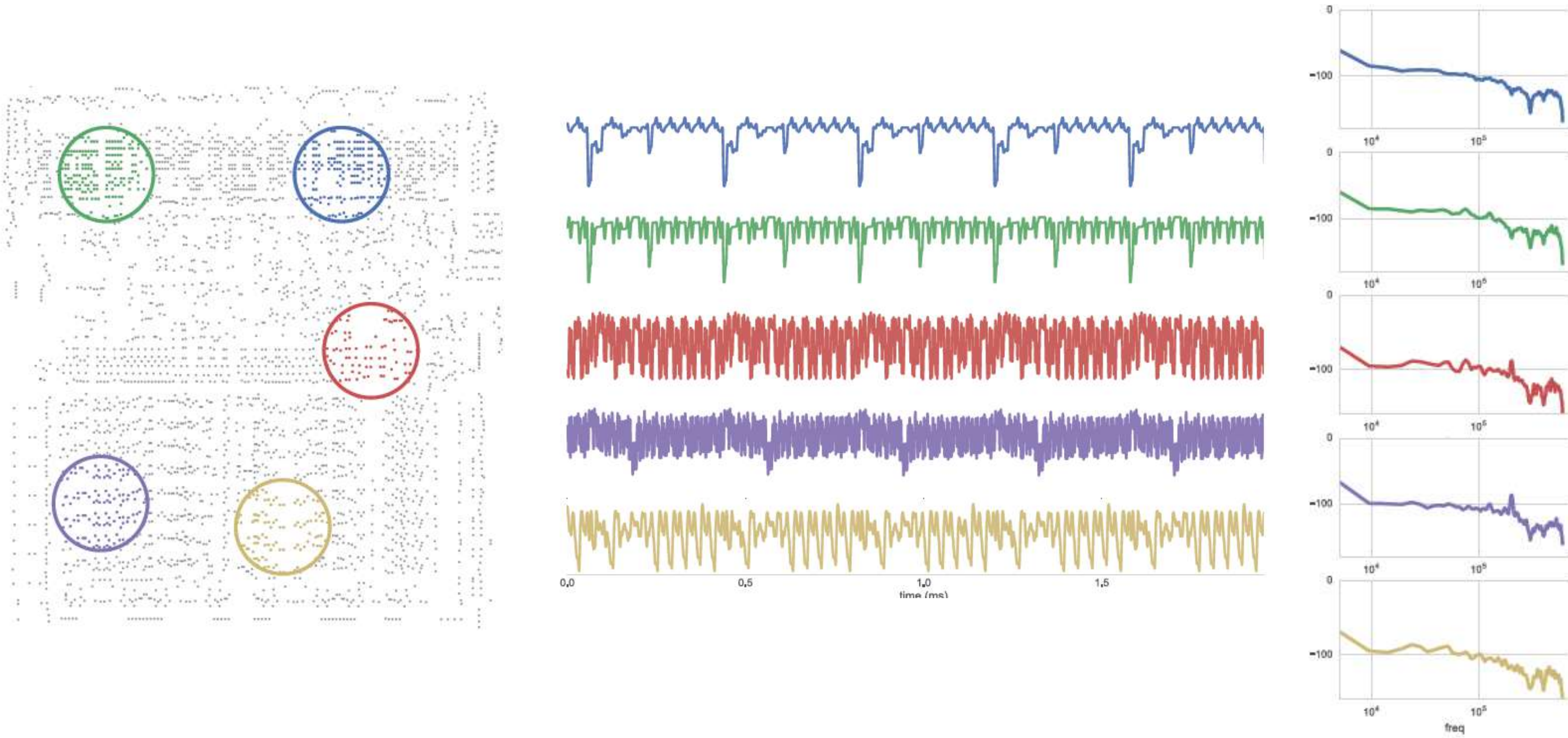


Tuning curves

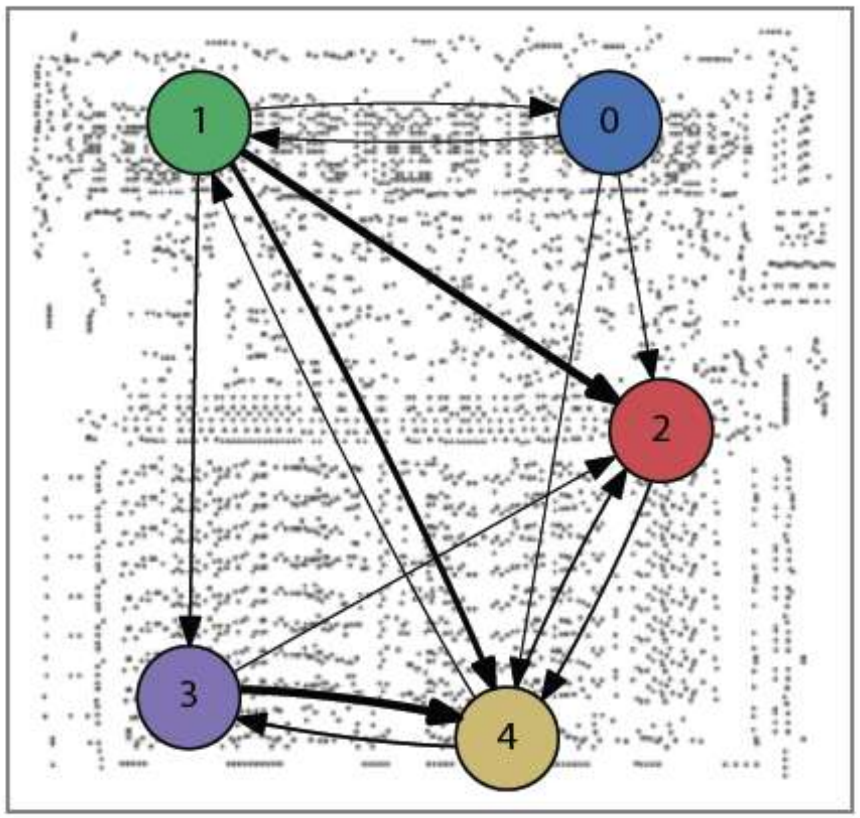




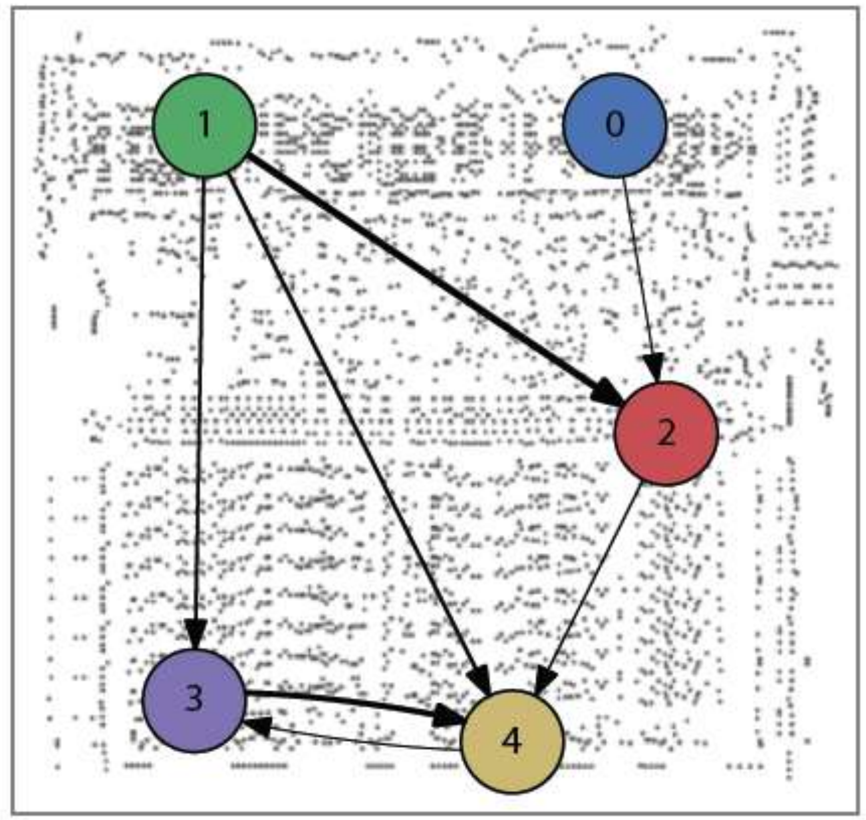
LFPs and power law spectra



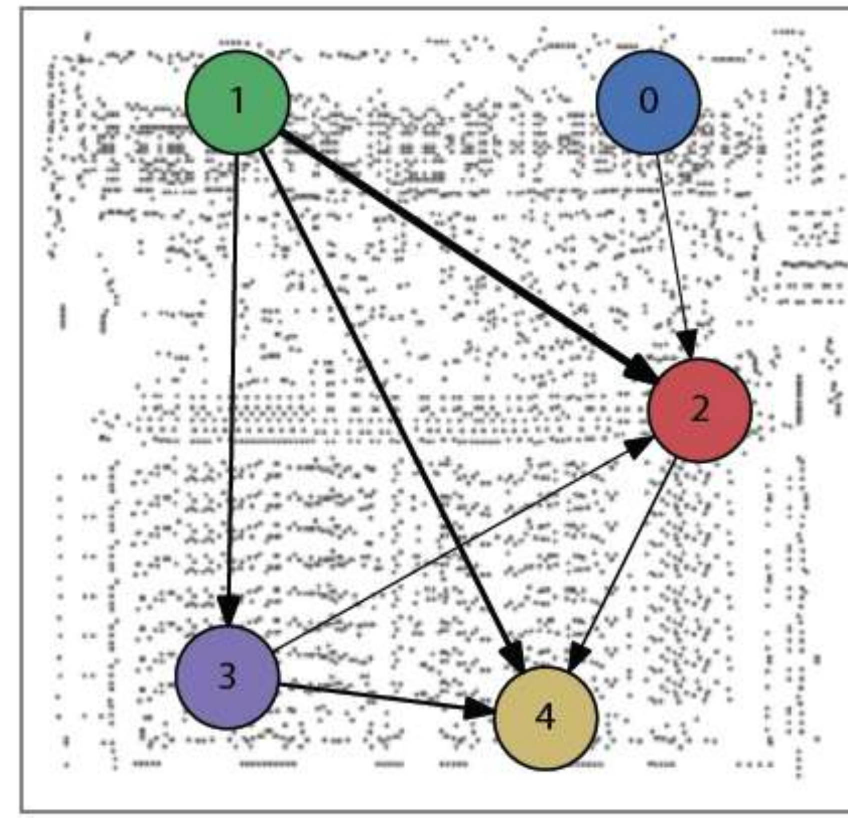
Granger causality



a. Donkey Kong

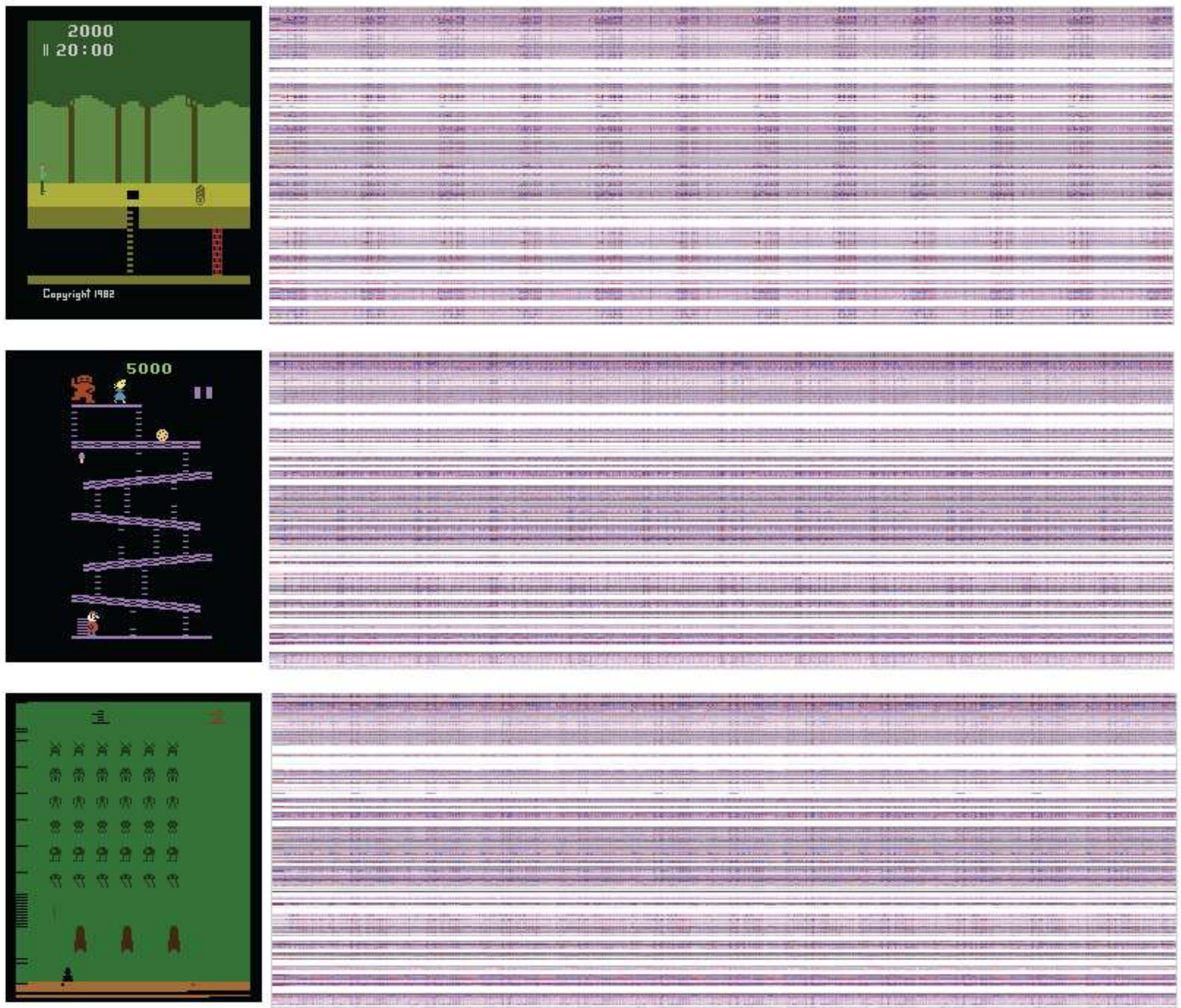


b. Space Invaders



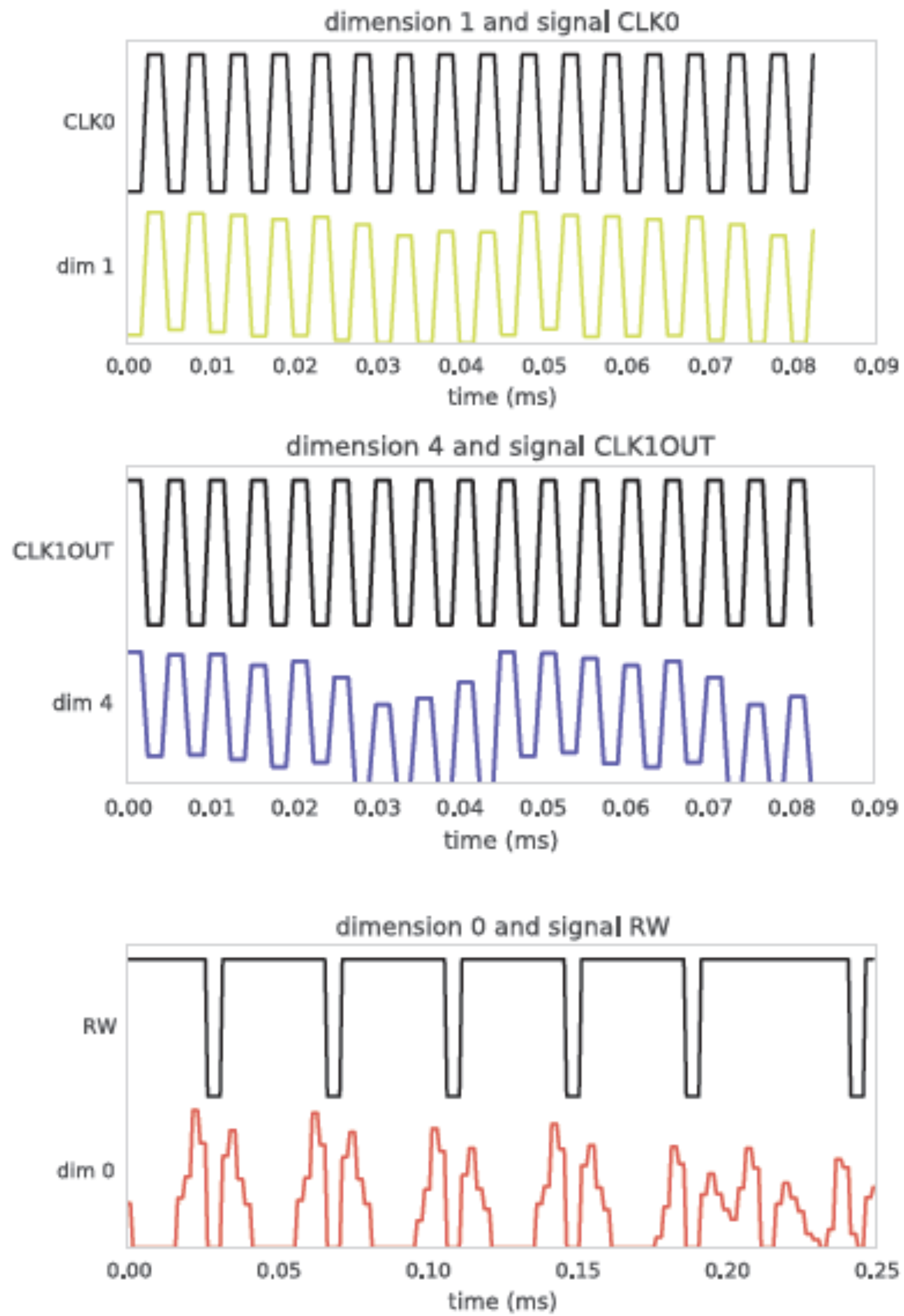
c. Pitfall

Whole chip



time

Nonnegative matrix factorization finds something



Medium: Simulate a trivial causal system

$$x_{t+1} = Ax_t + \epsilon$$

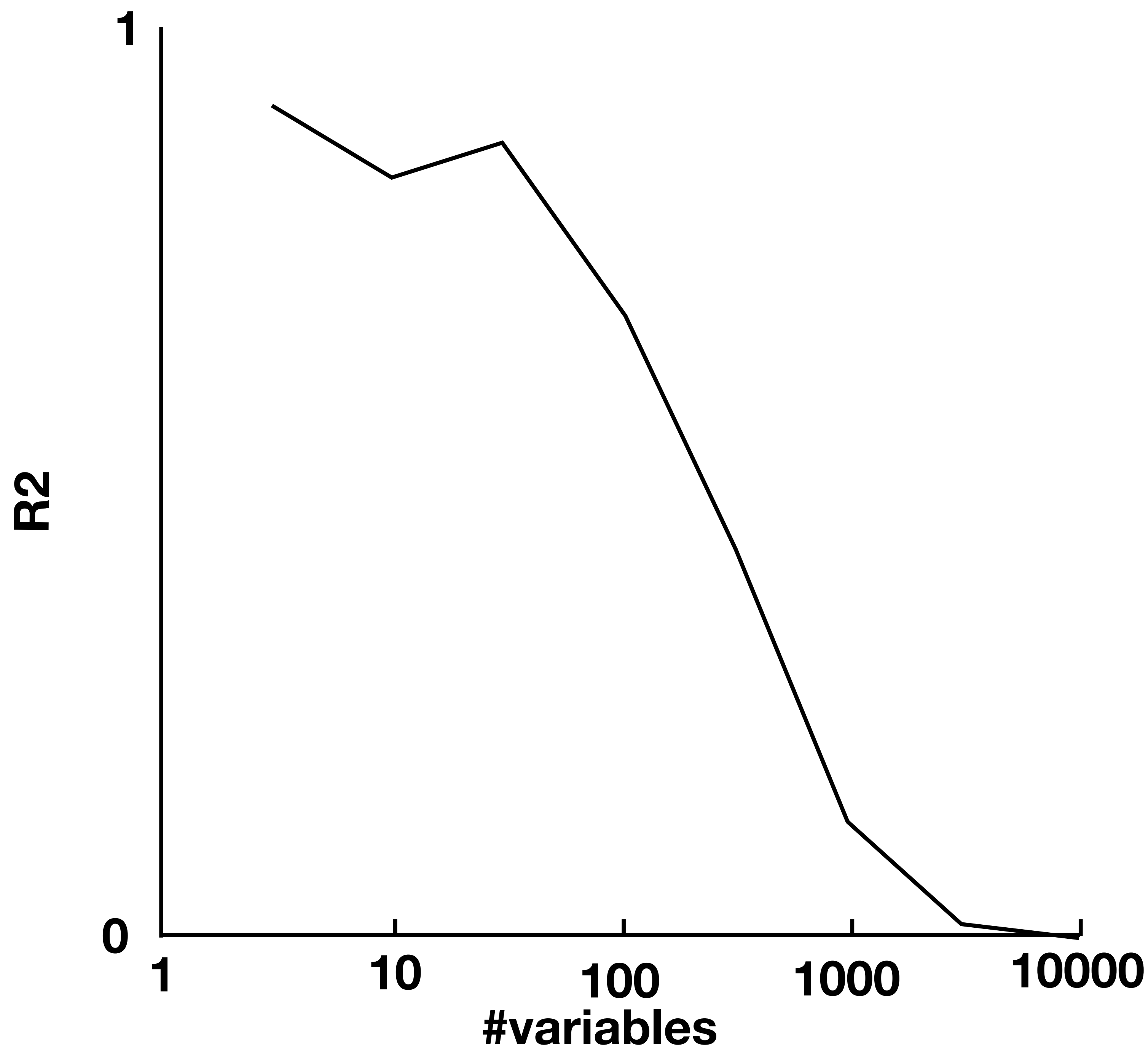
Where

$$\epsilon \sim \mathcal{N}(0, \Sigma)$$

$$\Sigma = diag(nL)$$

Choose A: sparse binary (p=.1), largest SV=.99

Delayed Correlation vs Causation



0) Simulate neural activities

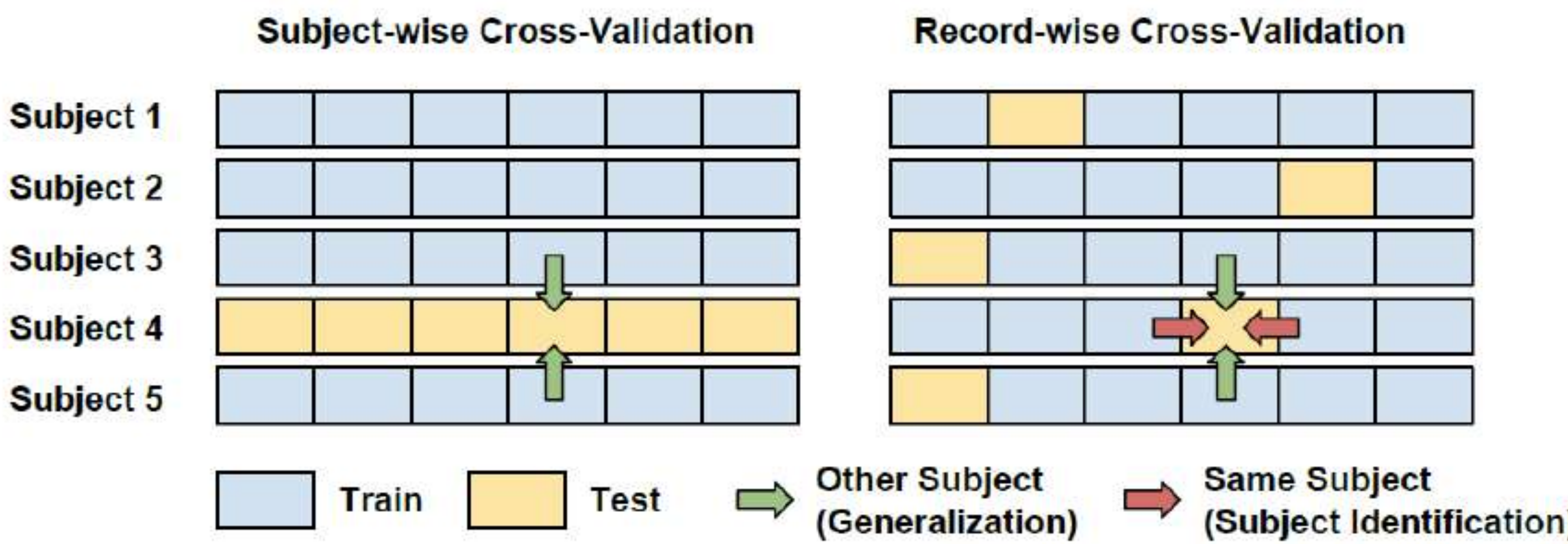
CODE

1) Overfitting

CODE

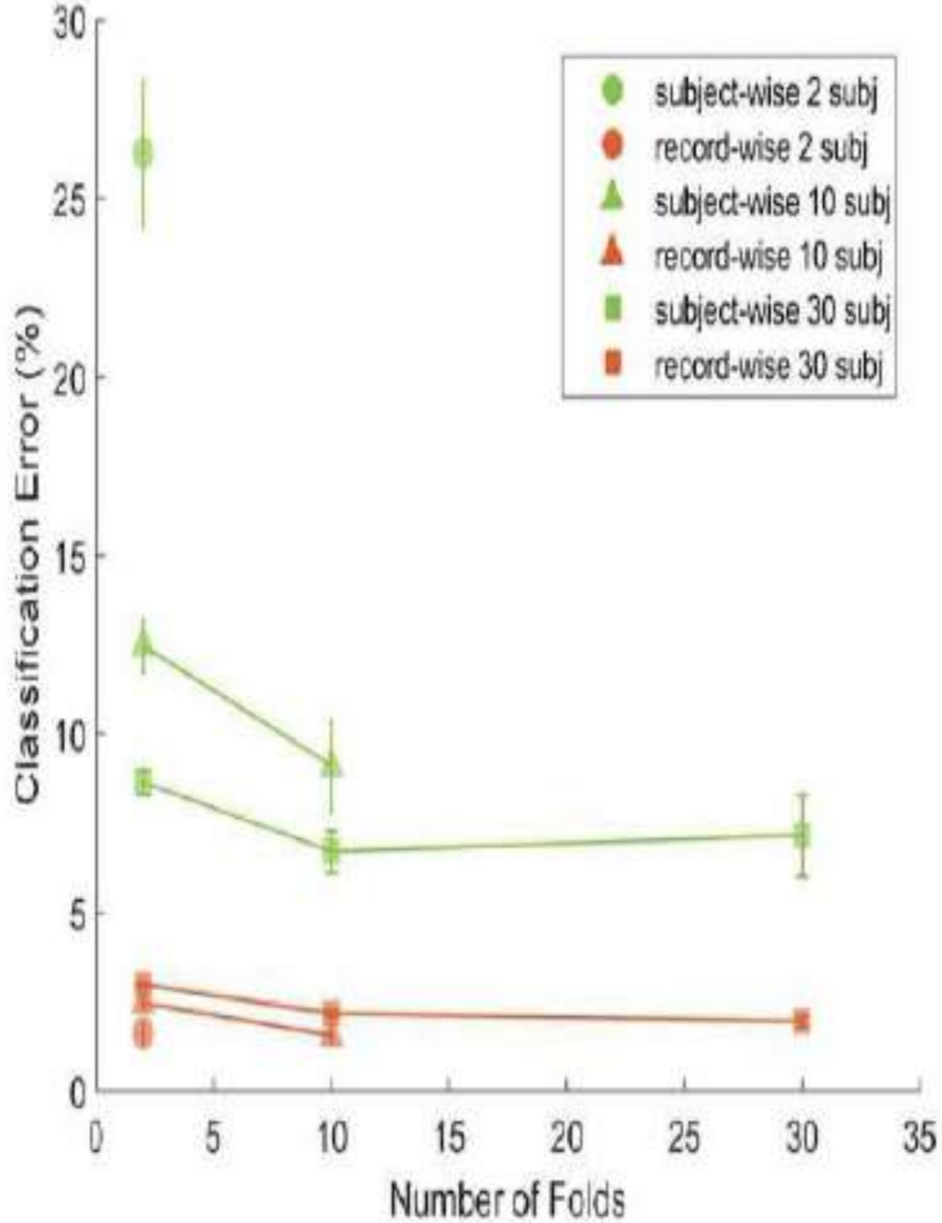
2) Crossvalidation

(2) Wrong way of assessing Quality e.g. bad crossvalidation

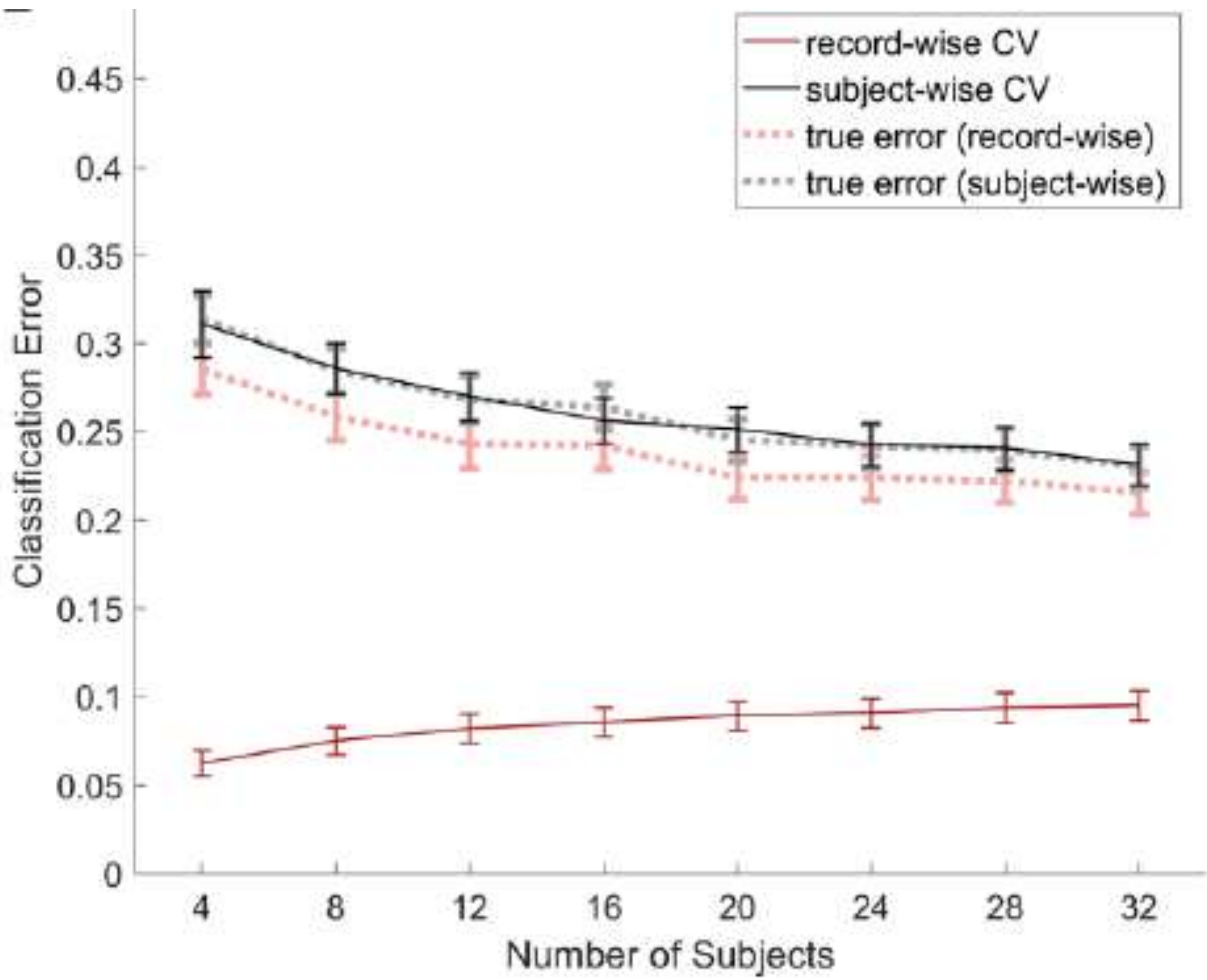


With Lonini, Saeb, Mohr, Jayaraman

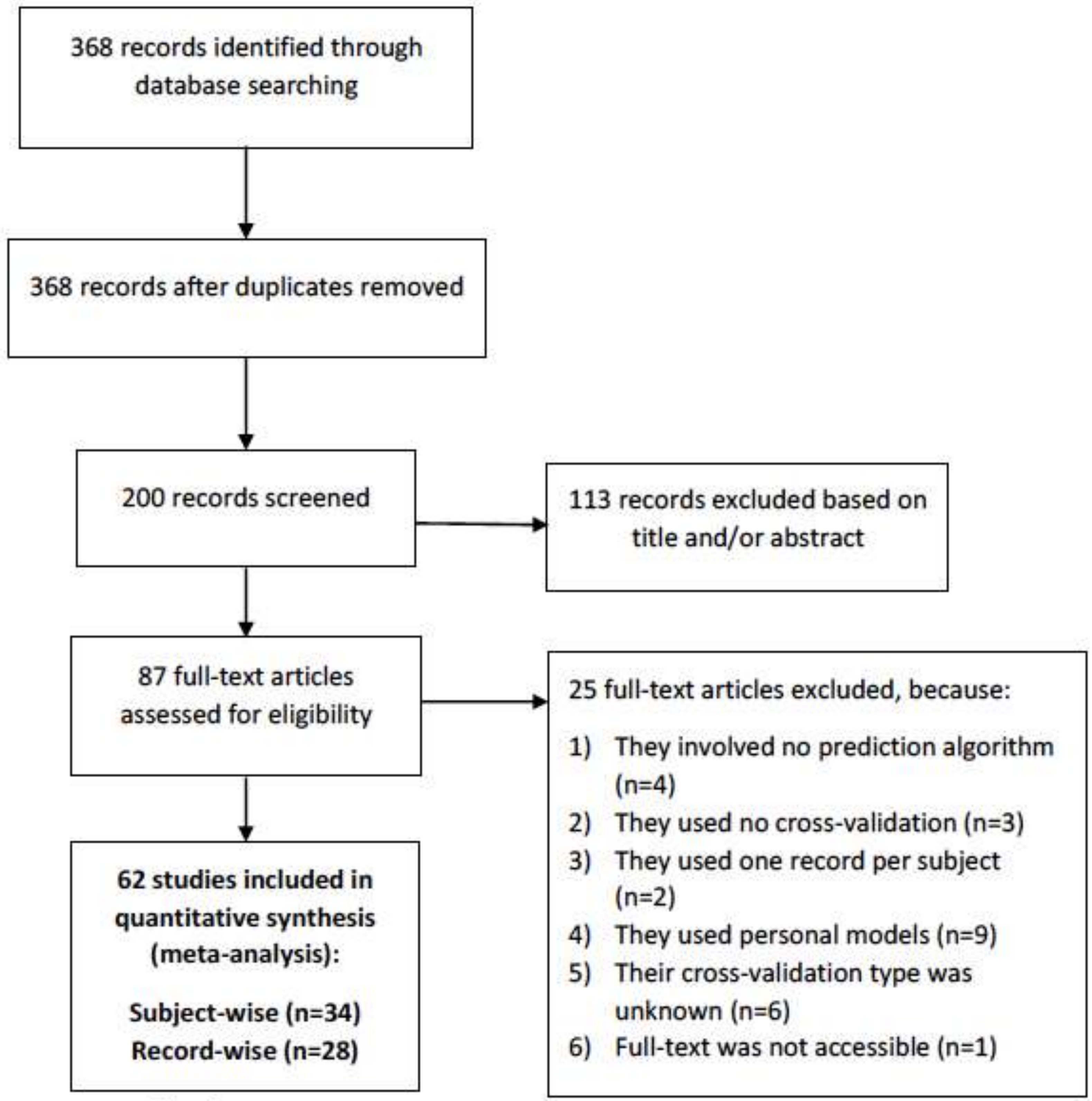
Cheating works



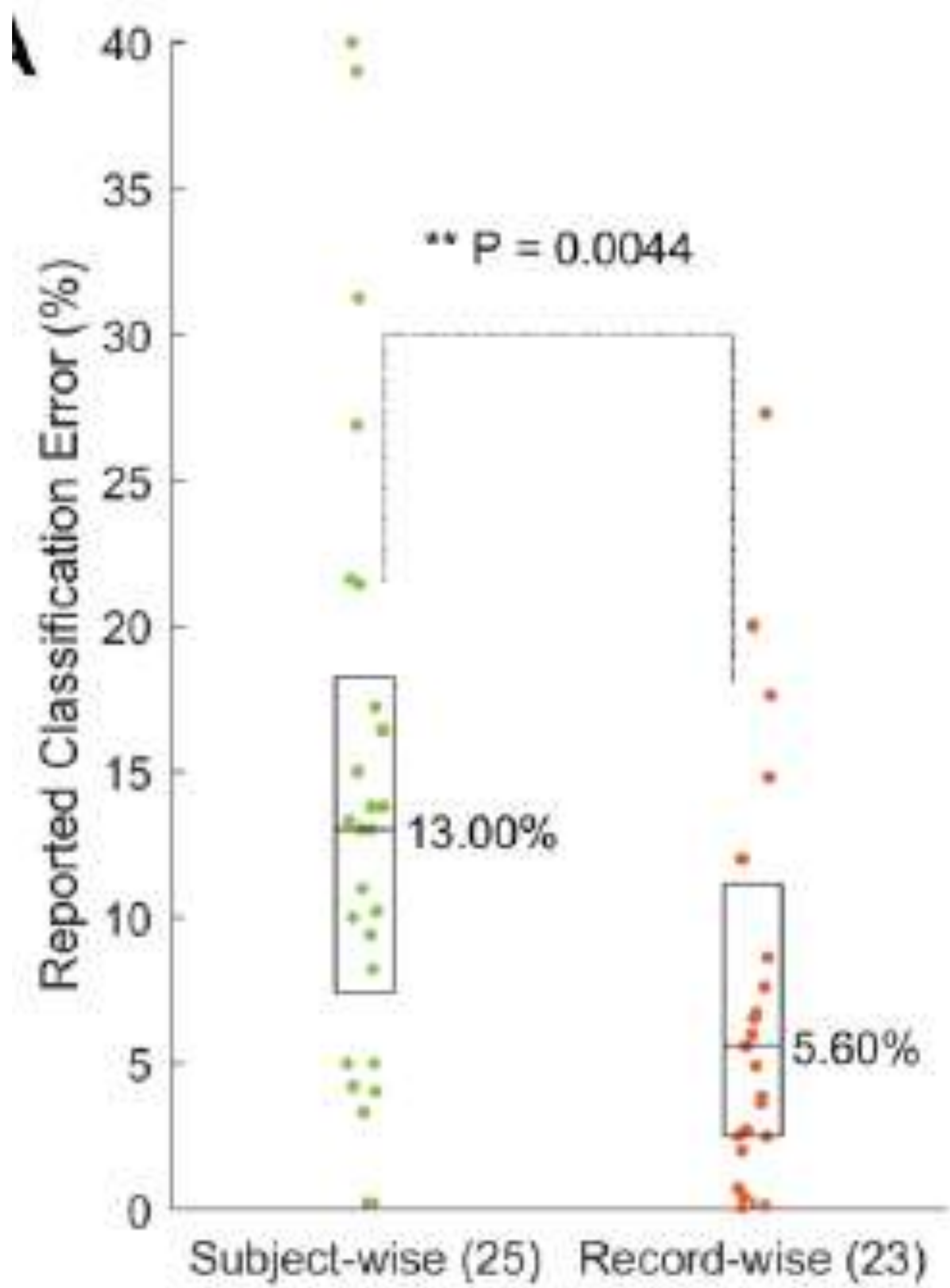
Massive overconfidence



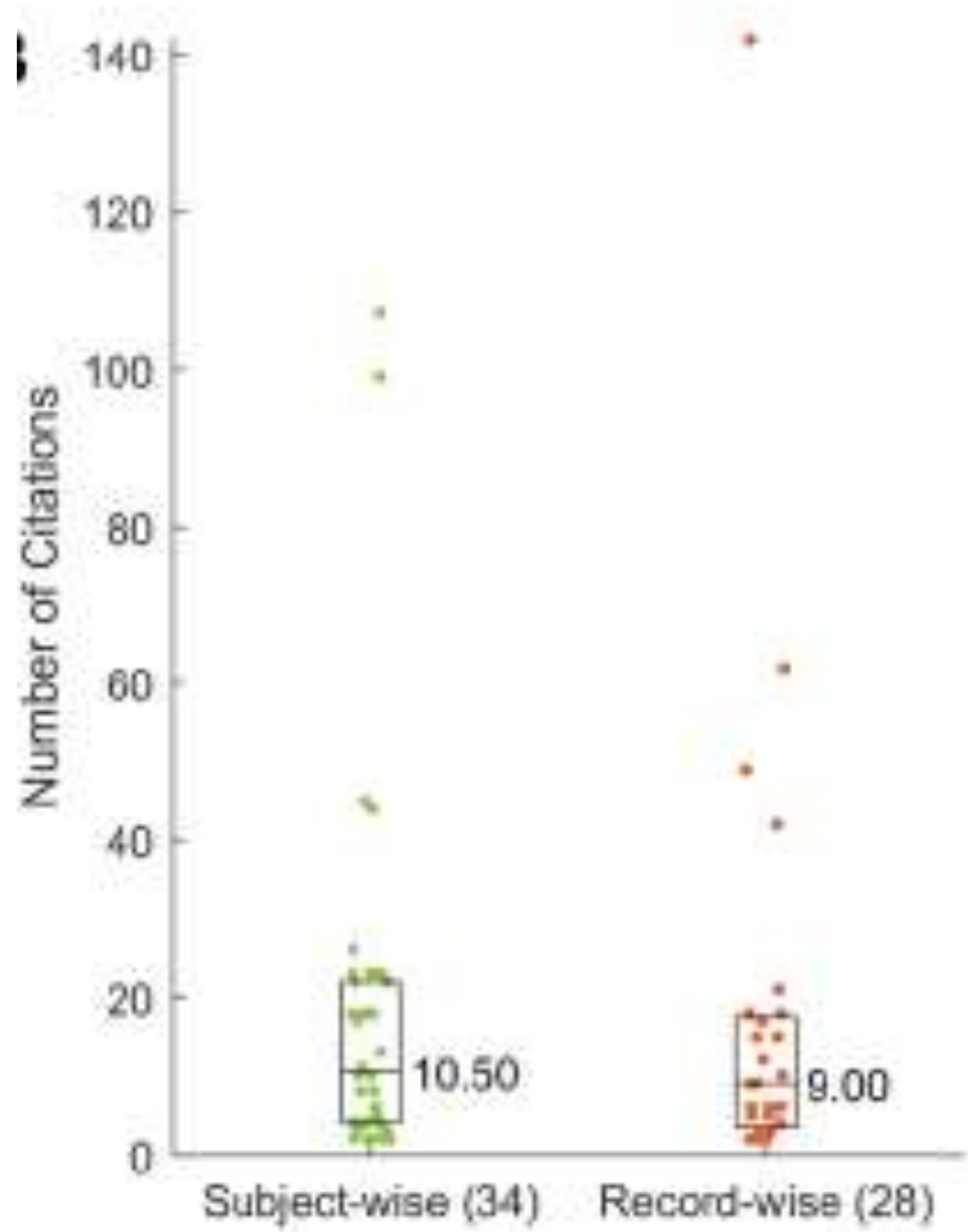
Literature review



Cheating helps



No one cares



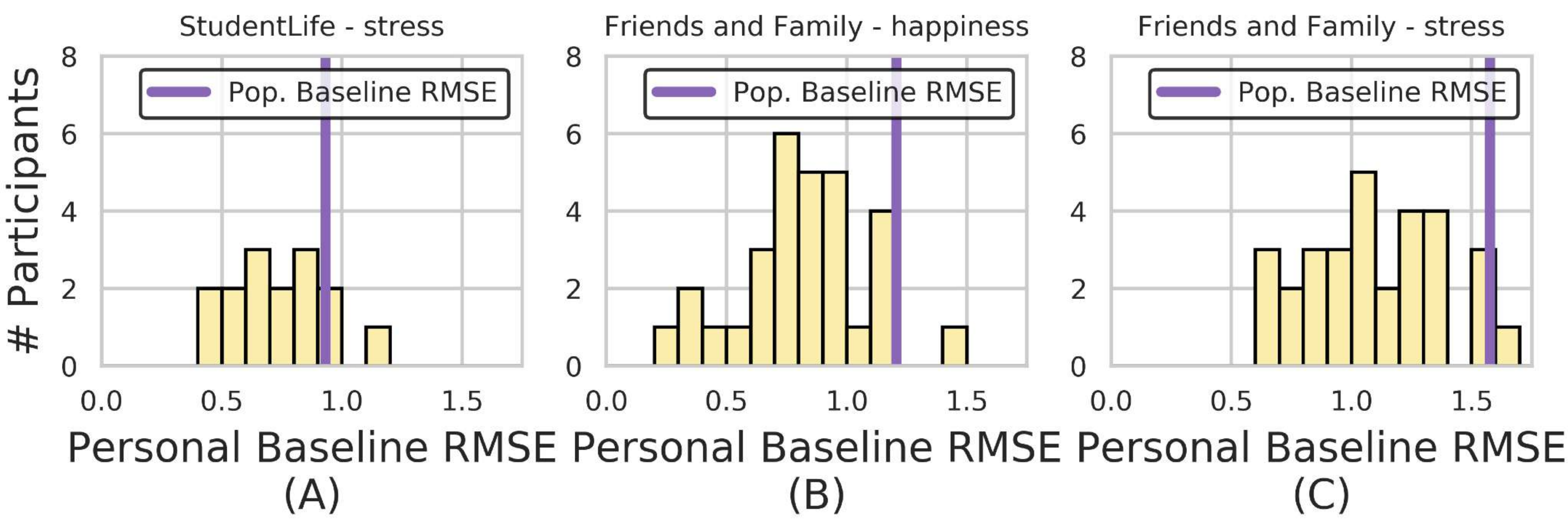
Number of Citations
140
120
100
80
60
40
20
0
10.50
9.00
Subject-wise (34)
Record-wise (28)

Relatedly: Wrong way of comparing e.g. personal baselines

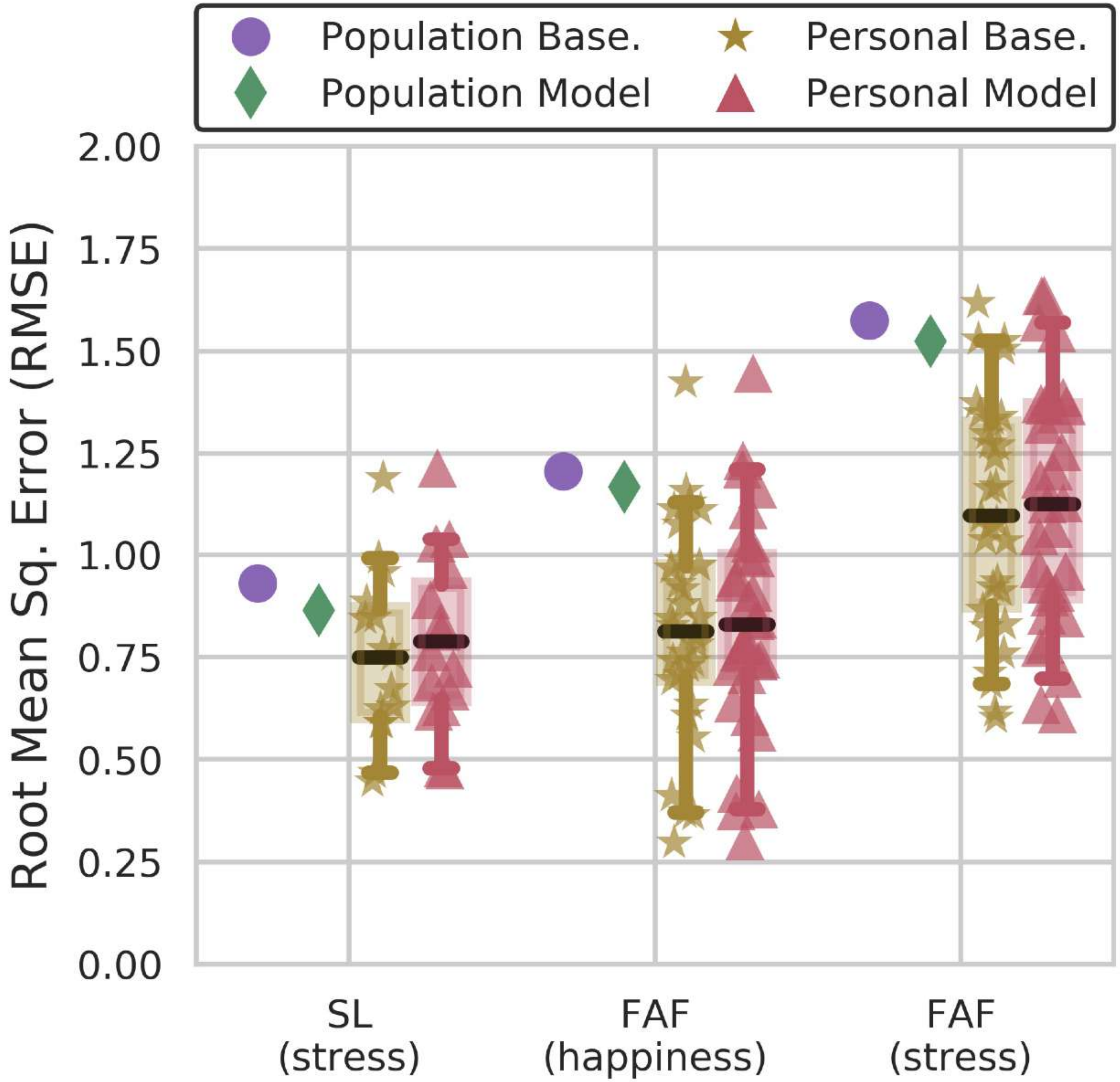
- Variance explained

W Orianna Demasi, Ben Recht

Personal vs group baselines



Machine learning often does not help

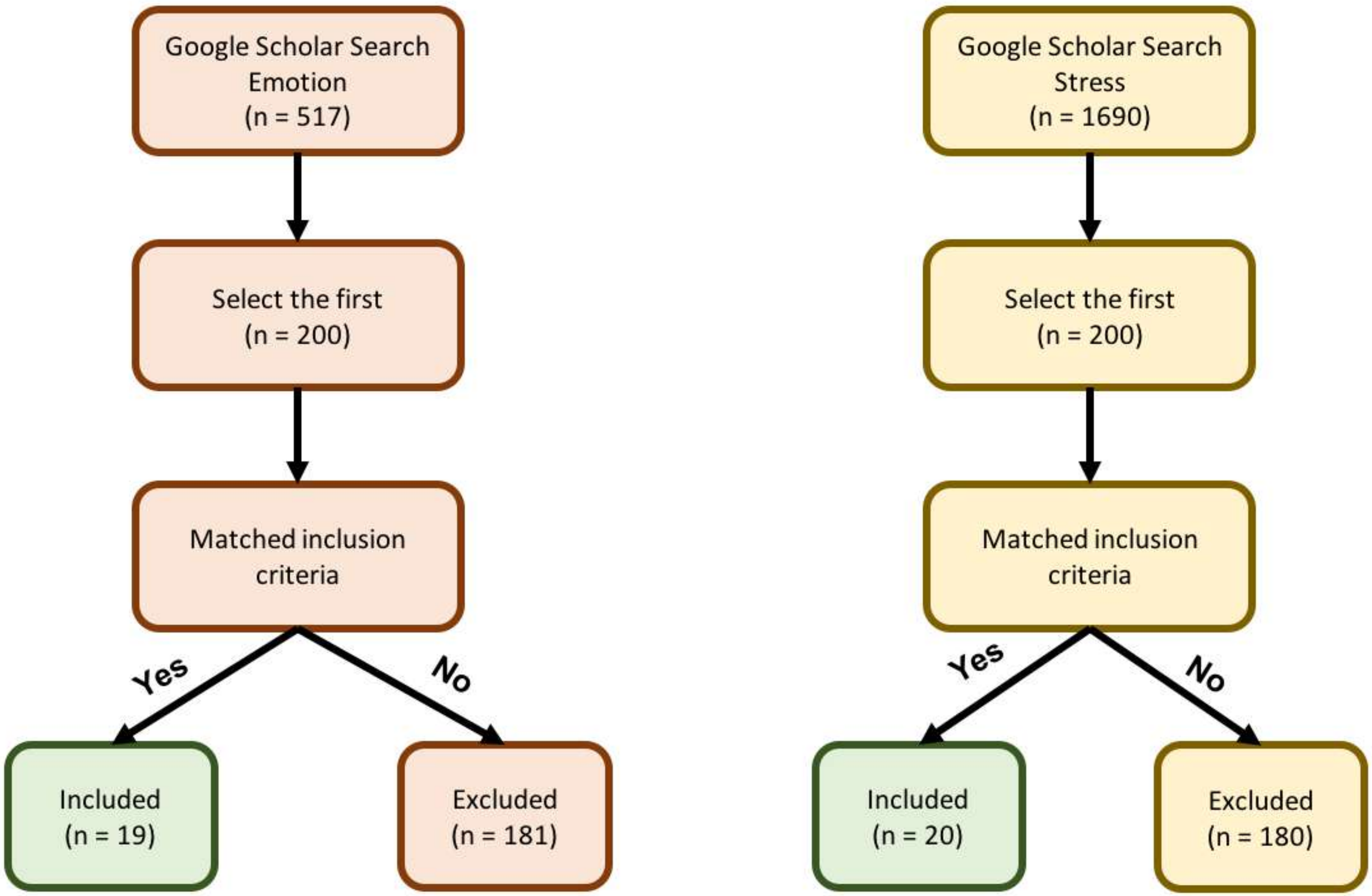


User lift

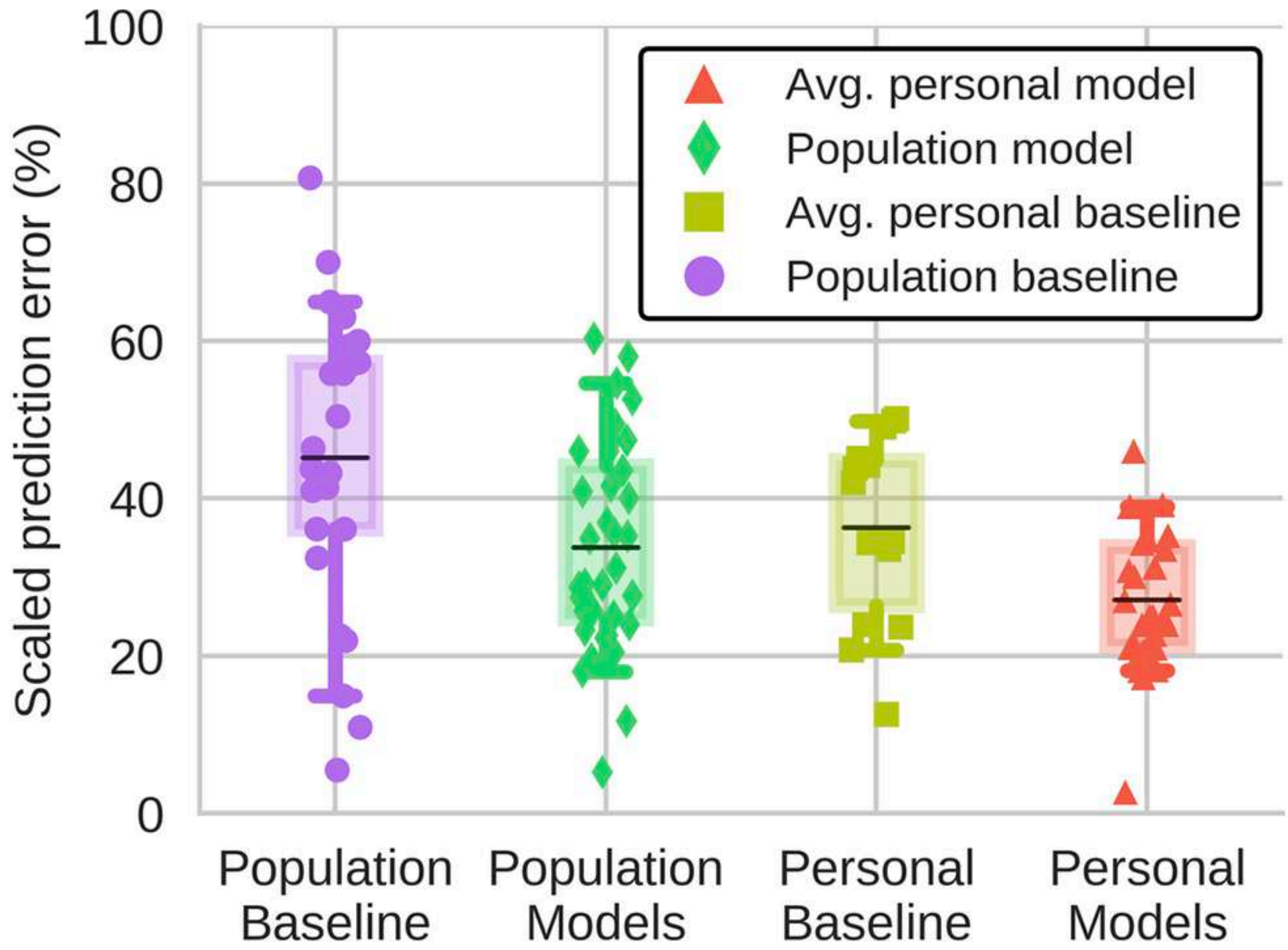
Dataset	Problem	Model	Avg. Personal Baseline Error	Avg. Personal Model Error	Avg. User Lift (Error)	p-value
SL—Stress	binary	Log.Reg.	29.19%	29.09%	**0.10**	.481
FaF—Happiness	binary	SVM(rbf)	16.51%	18.67%	**-2.17**	.967
FaF—Stress	binary	SVM(rbf)	25.17%	23.35%	**1.82**	.240
SL—Stress	regression	Elastic Net	0.75	0.78	**-0.03**	.988
FaF—Happiness	regression	Elastic Net	0.81	0.83	**-0.02**	.999
FaF—Stress	regression	Elastic Net	1.10	1.13	**-0.03**	1.000

https://doi.org/10.1371/journal.pone.0184604.t001

Literature review

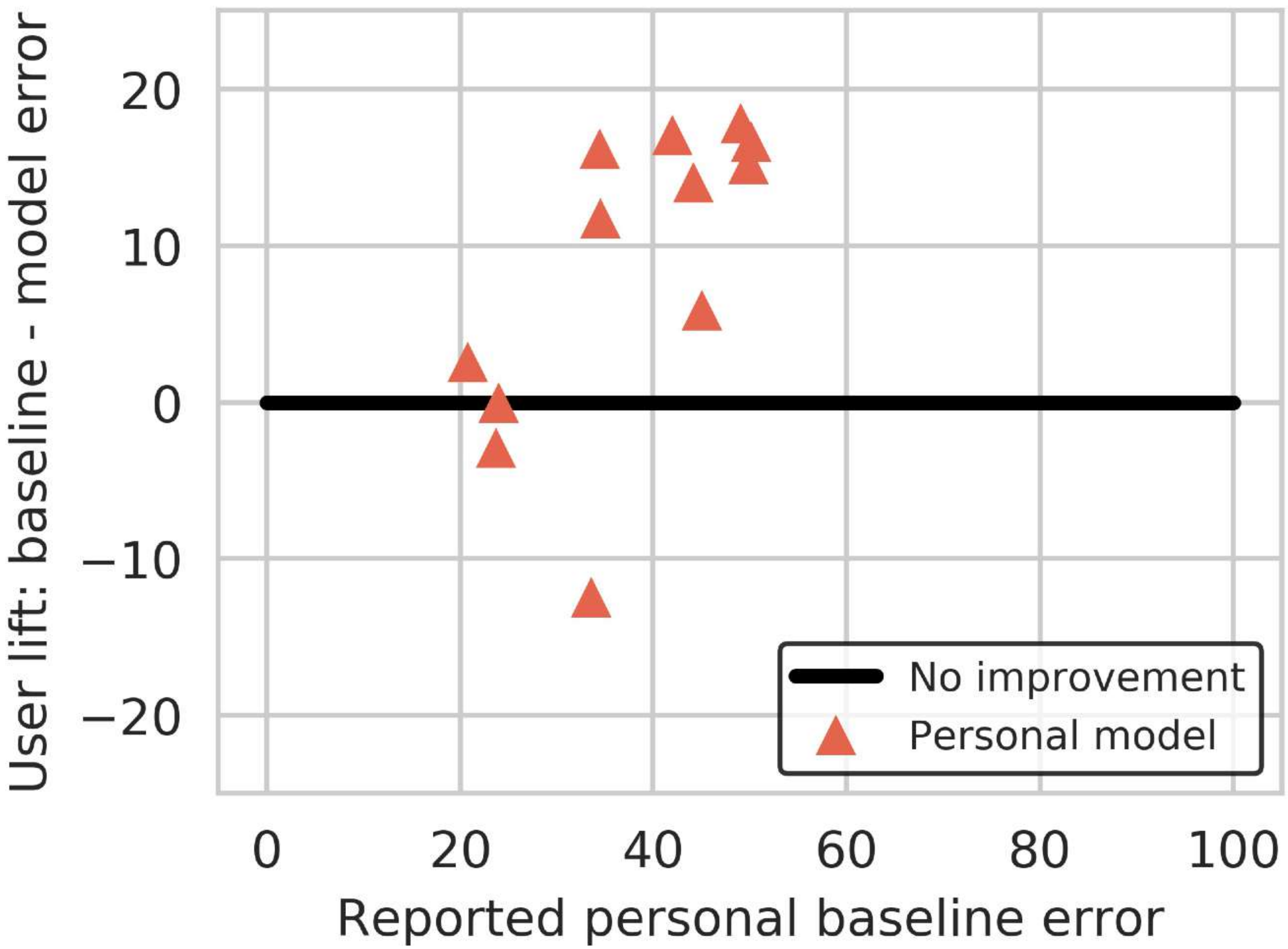


Machine learning often does not help



(B)

Does ML even help?



Let us talk about train/validation/test

CODE

3) Regularization

- Blackboard

Code

4) With RNNs

Naive Bayes

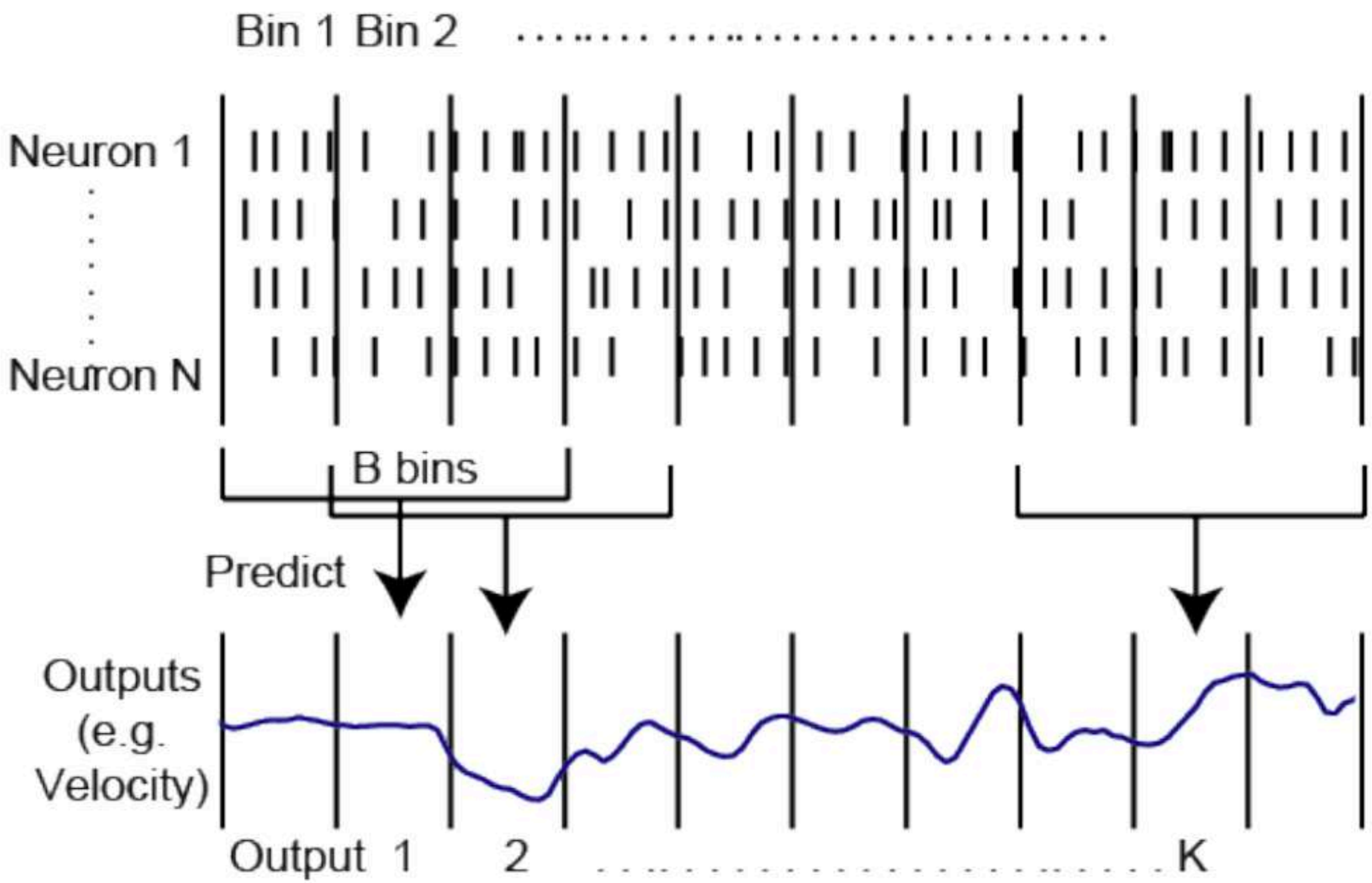
CODE

Now lets take some time to go through the python version

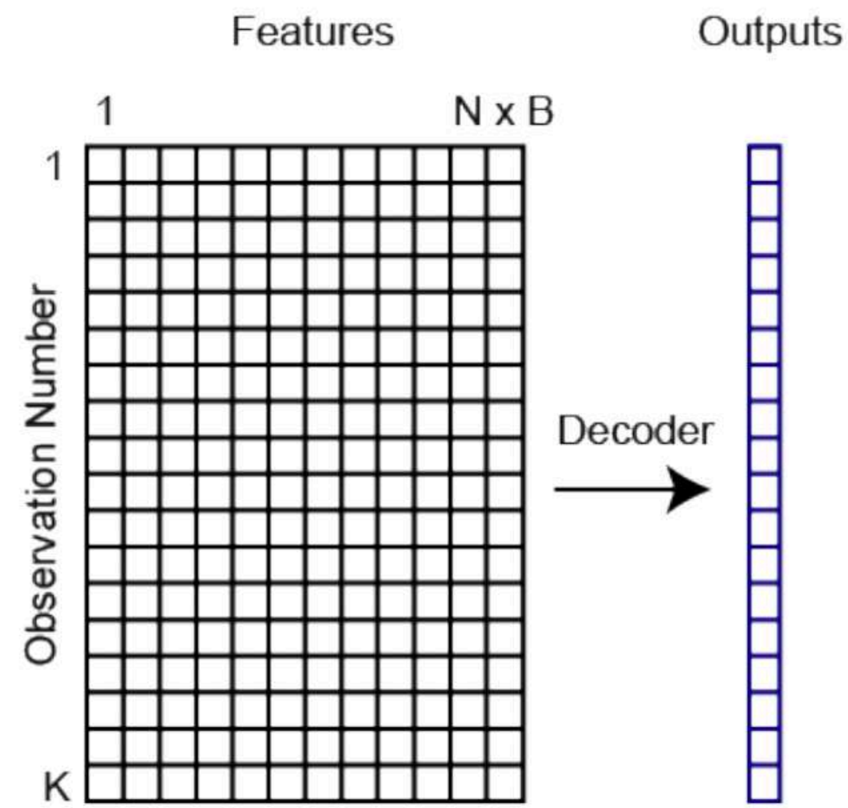
5) Comparison on real data

- Glaser, Choudhury, Perich, Miller, Kording

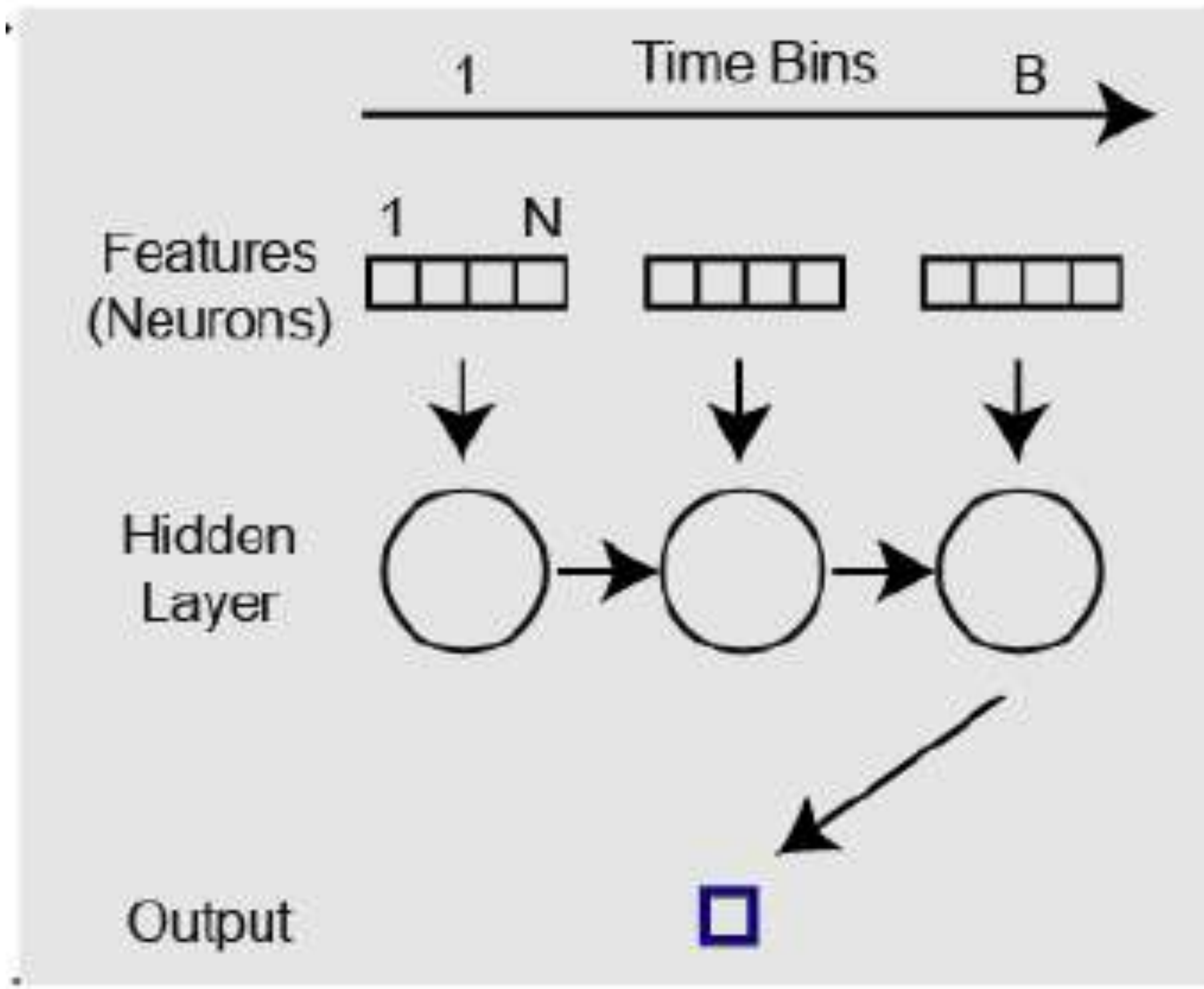
Dealing with time

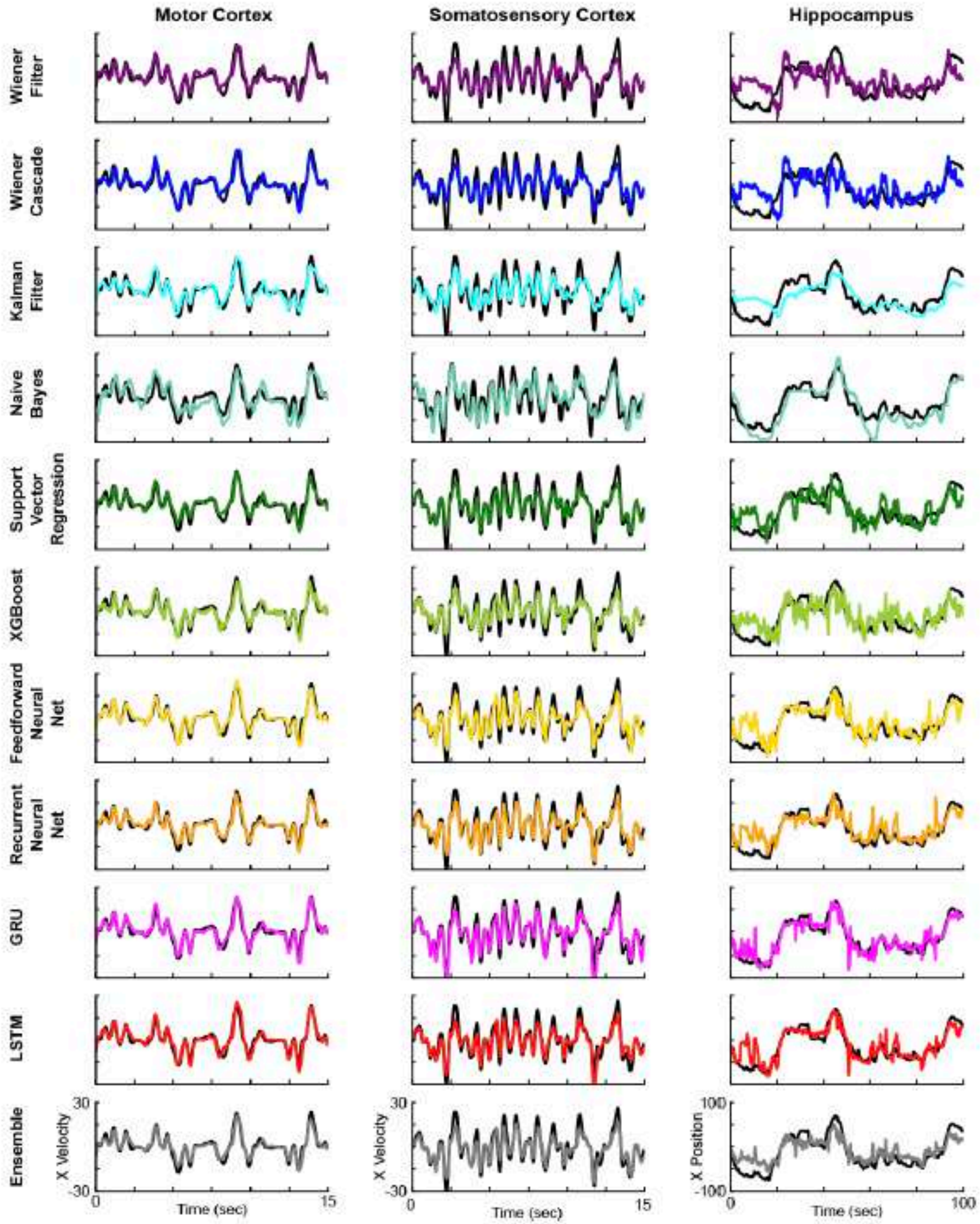


Non-recurrent decoders



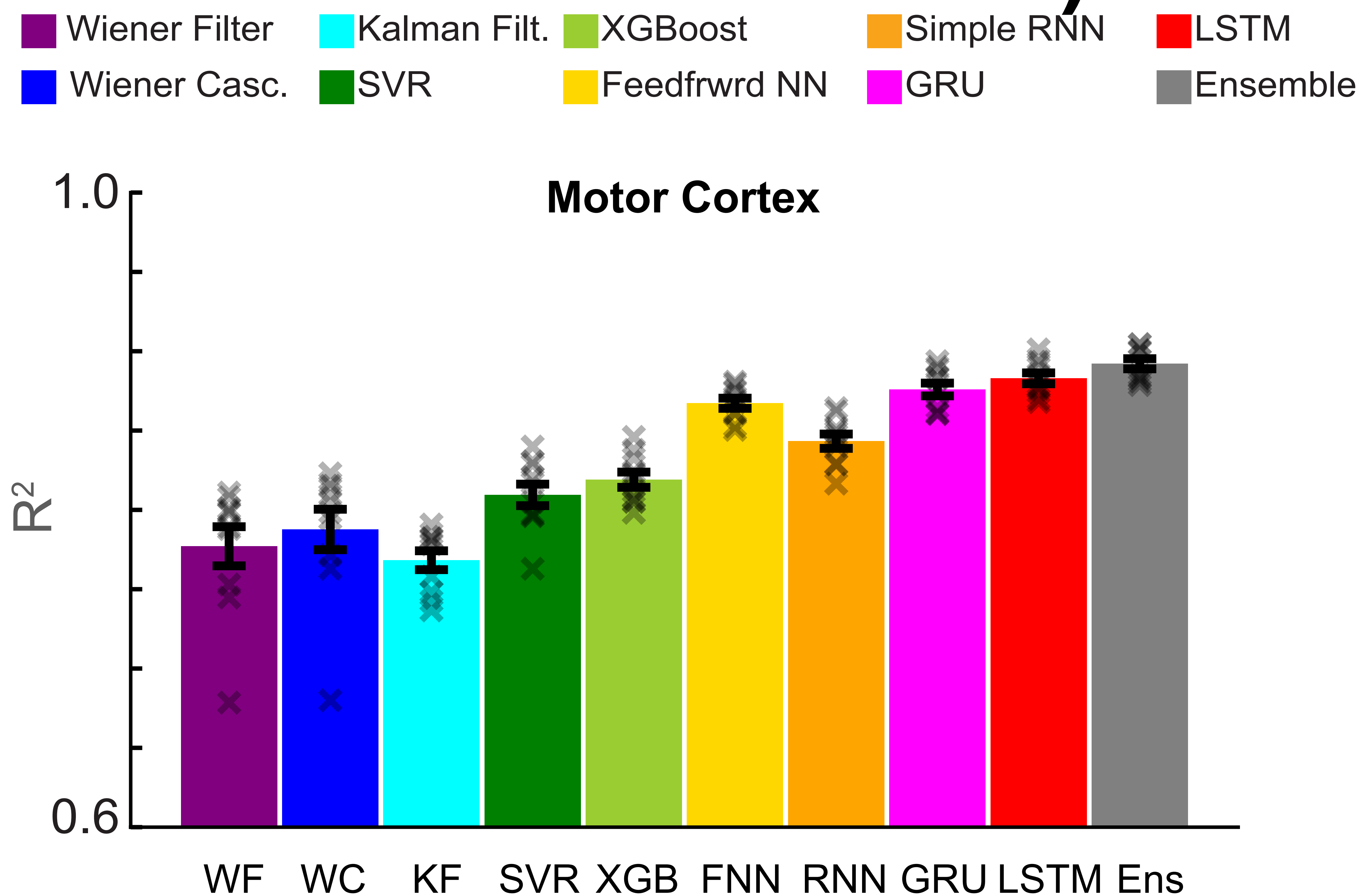
Recurrent decoders



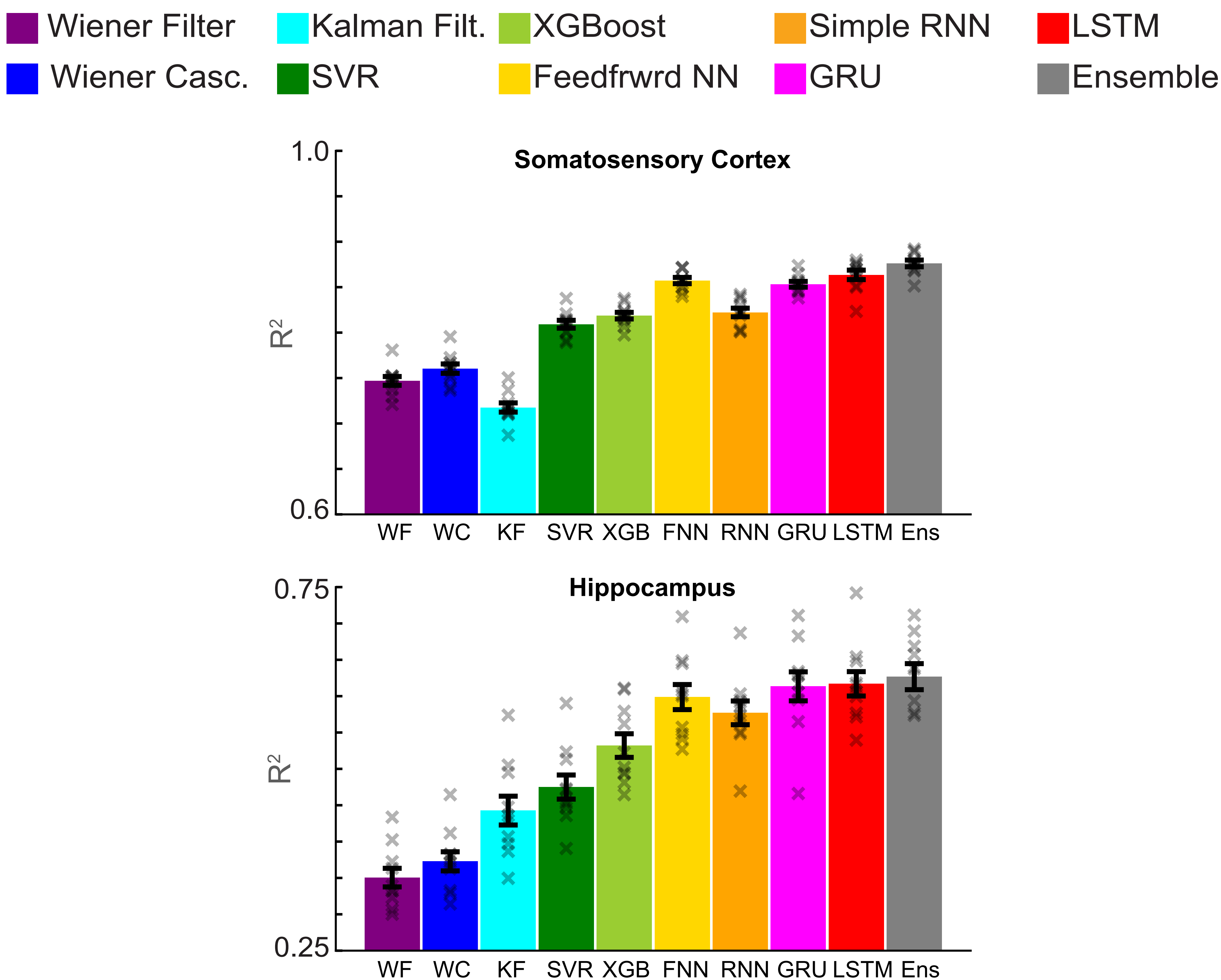


Motor Cortex
Somatosensory Cortex
Hippocampus
Wiener Filter
Wiener Cascade
Kalman Filter
Naive Bayes
Support Vector Regression
XGBoost
Feedforward Neural Net
Recurrent Neural Net
GRU
LSTM
Ensemble
X Velocity
30
-30
0
15
Time (sec)
X Position
100
-100

Decoding (Neurons-> movement)



Finding generalizes

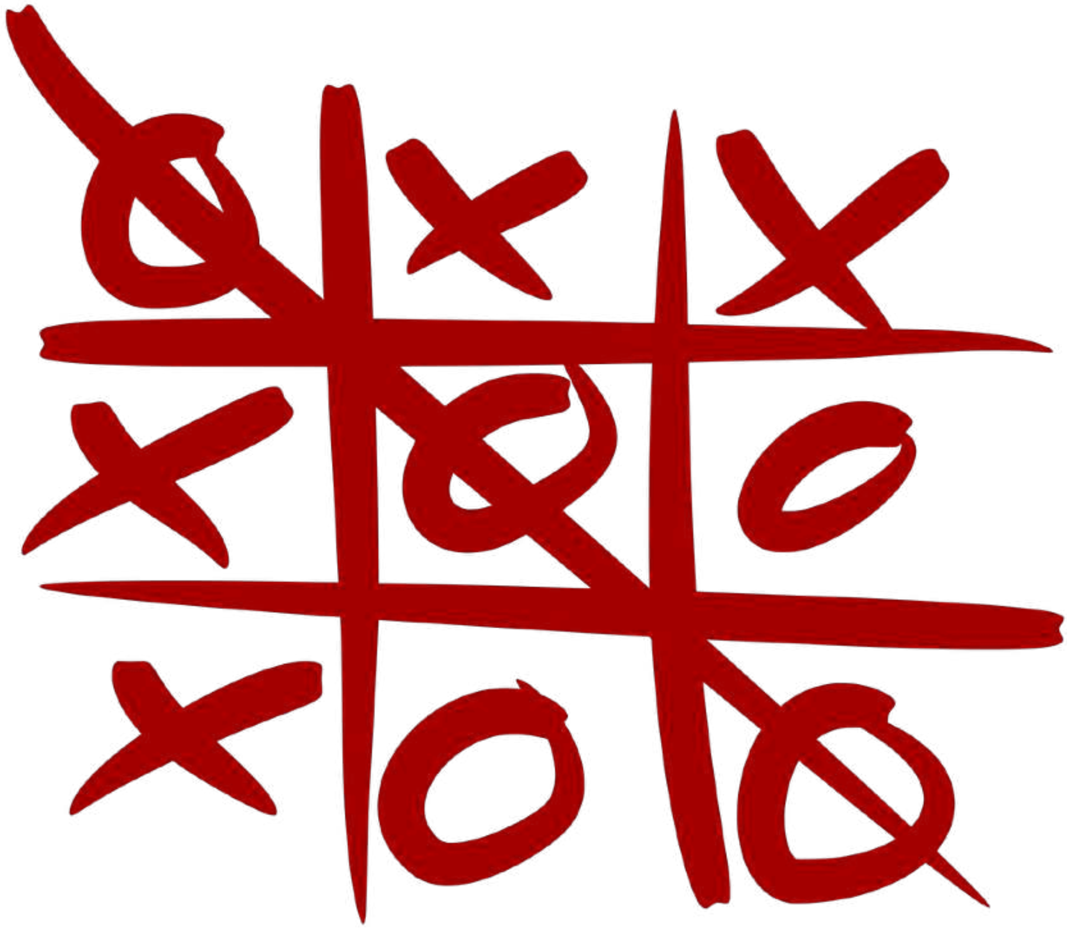


CODE

```
#Use one-hot coding for y
if y_train.ndim==1:
    y_train=np_utils.to_categorical(y_train.astype(int))
elif y_train.shape[1]==1:
    y_train=np_utils.to_categorical(y_train.astype(int))

model=Sequential() #Declare model
#Add recurrent layer

#### MAKE RELU ACTIVATION BELOW LIKE IN REGRESSION????? ####
model.add(SimpleRNN(self.units,input_shape=(X_train.shape[1],X_train.shape[2]),dropout_W=self.dropout,dropout_U=self.dr
if self.dropout!=0: model.add(Dropout(self.dropout)) #Dropout some units (recurrent layer output units)

#Add dense connections to output layer
model.add(Dense(y_train.shape[1]))
model.add(Activation('softplus'))

#Fit model (and set fitting parameters)
model.compile(loss='categorical_crossentropy',optimizer='rmsprop',metrics=['accuracy']) #Set loss function and optimize
model.fit(X_train,y_train,nb_epoch=self.num_epochs,verbose=self.verbose) #Fit the model
self.model=model
```

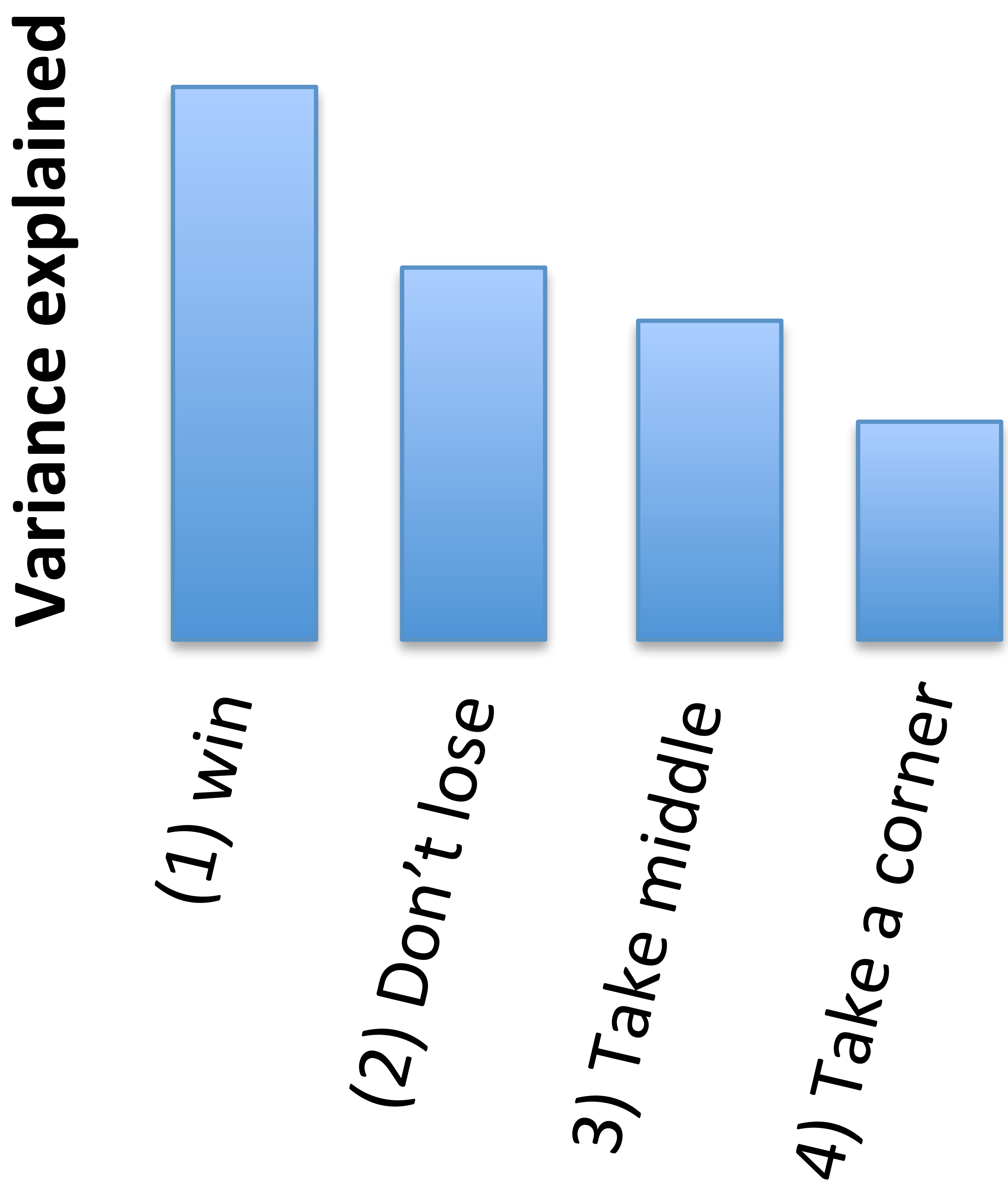
A post-rant

Tic Tac Toe



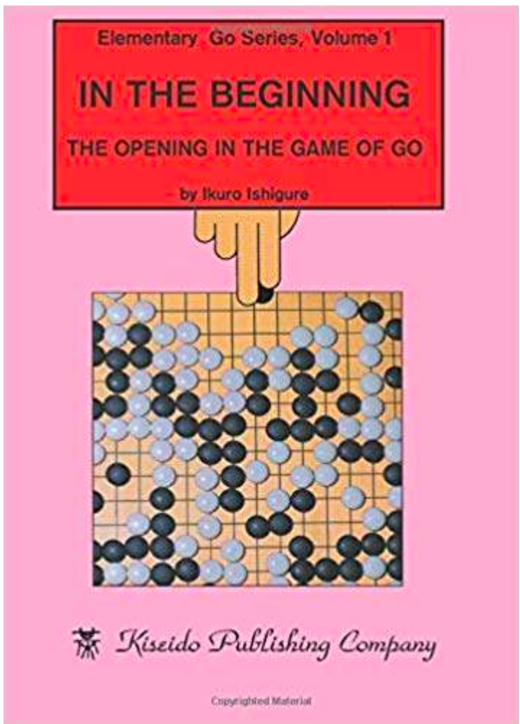
255,168 distinct games!

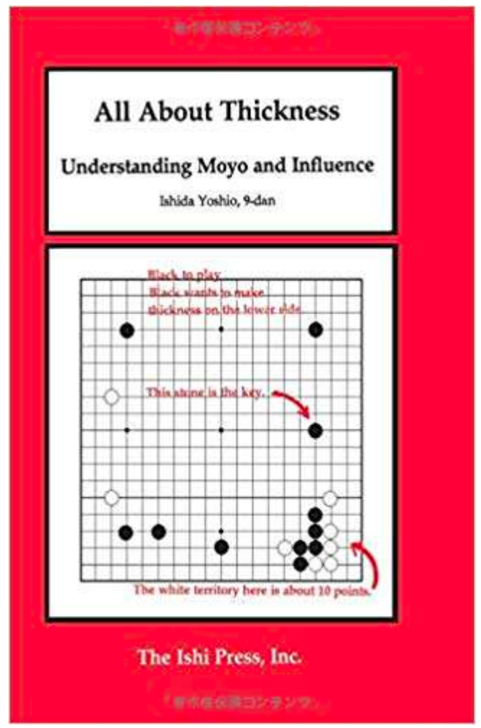
Compressable

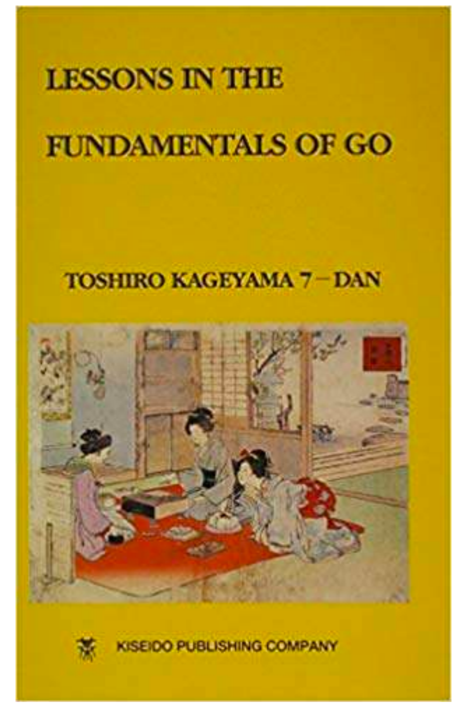


Go

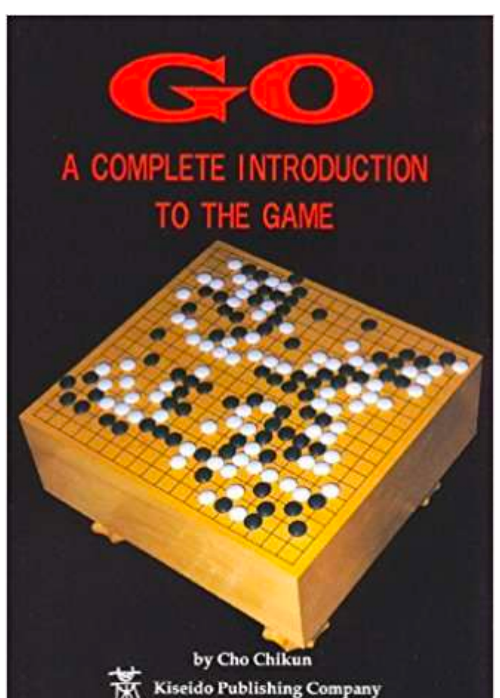
Probably no way to compactly describe it

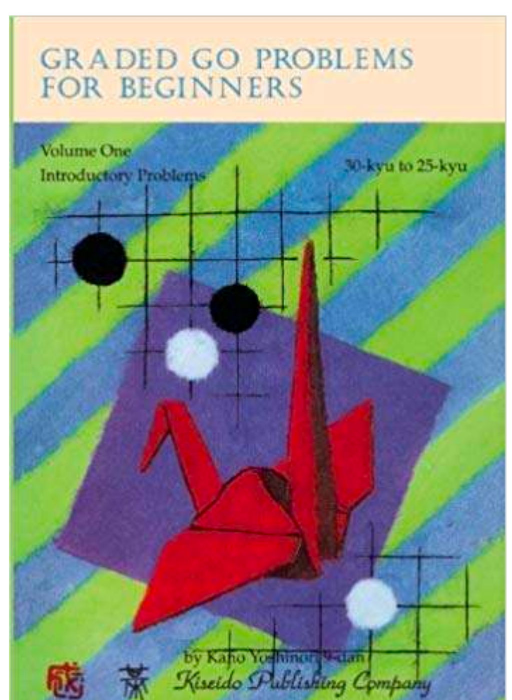


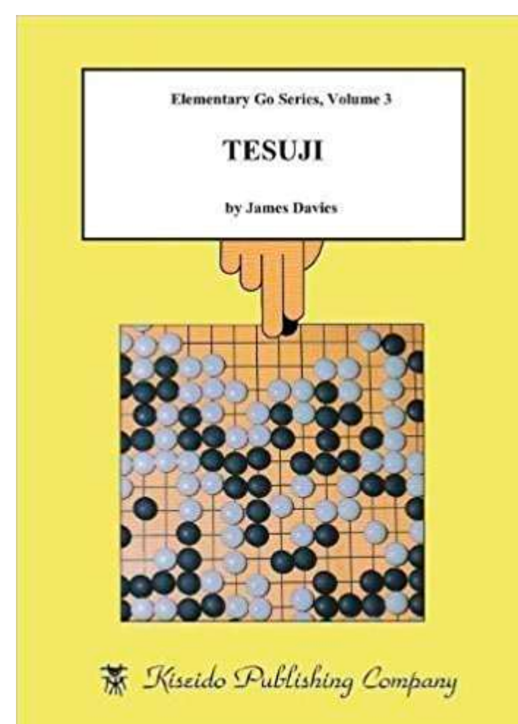


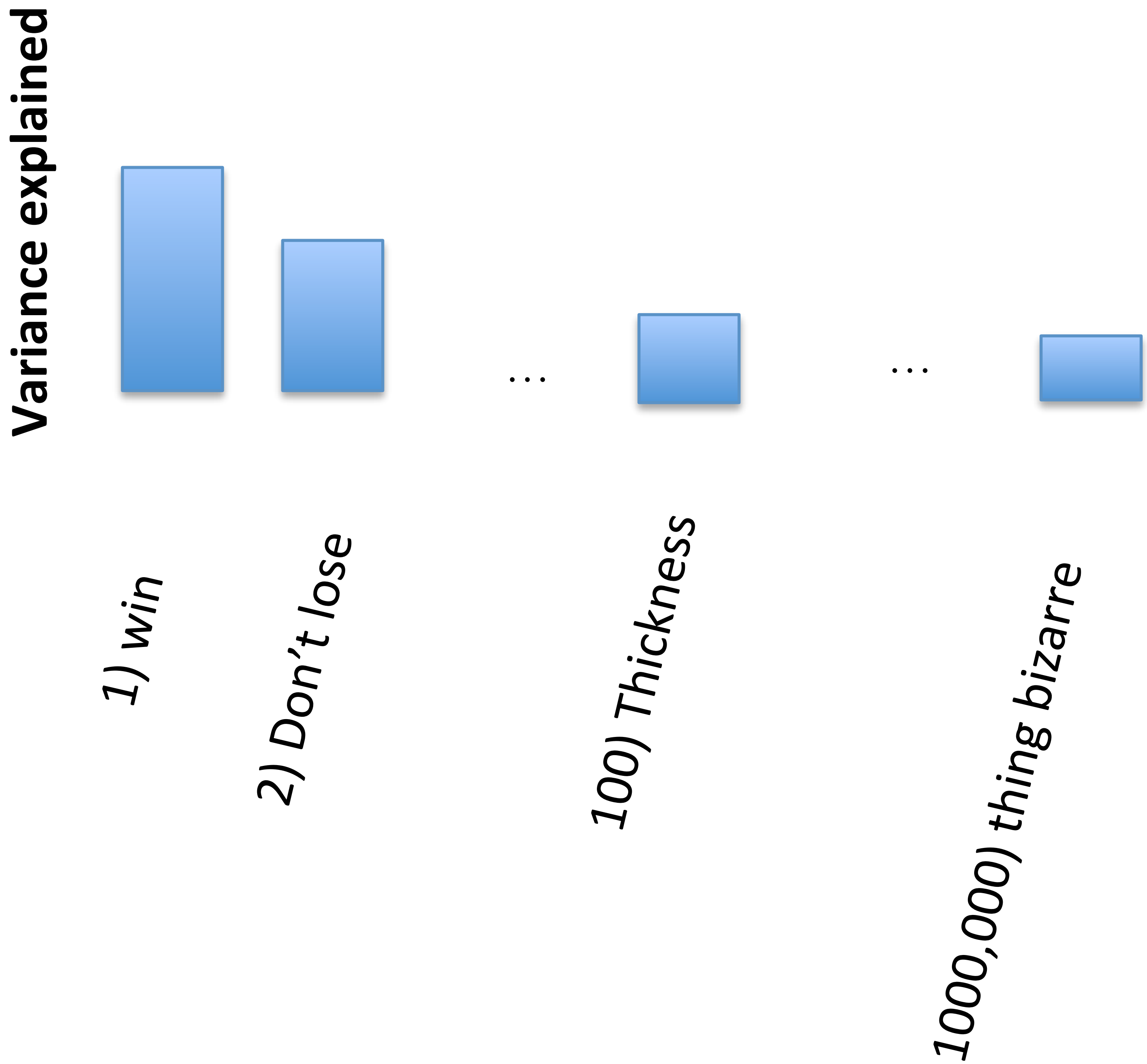


...





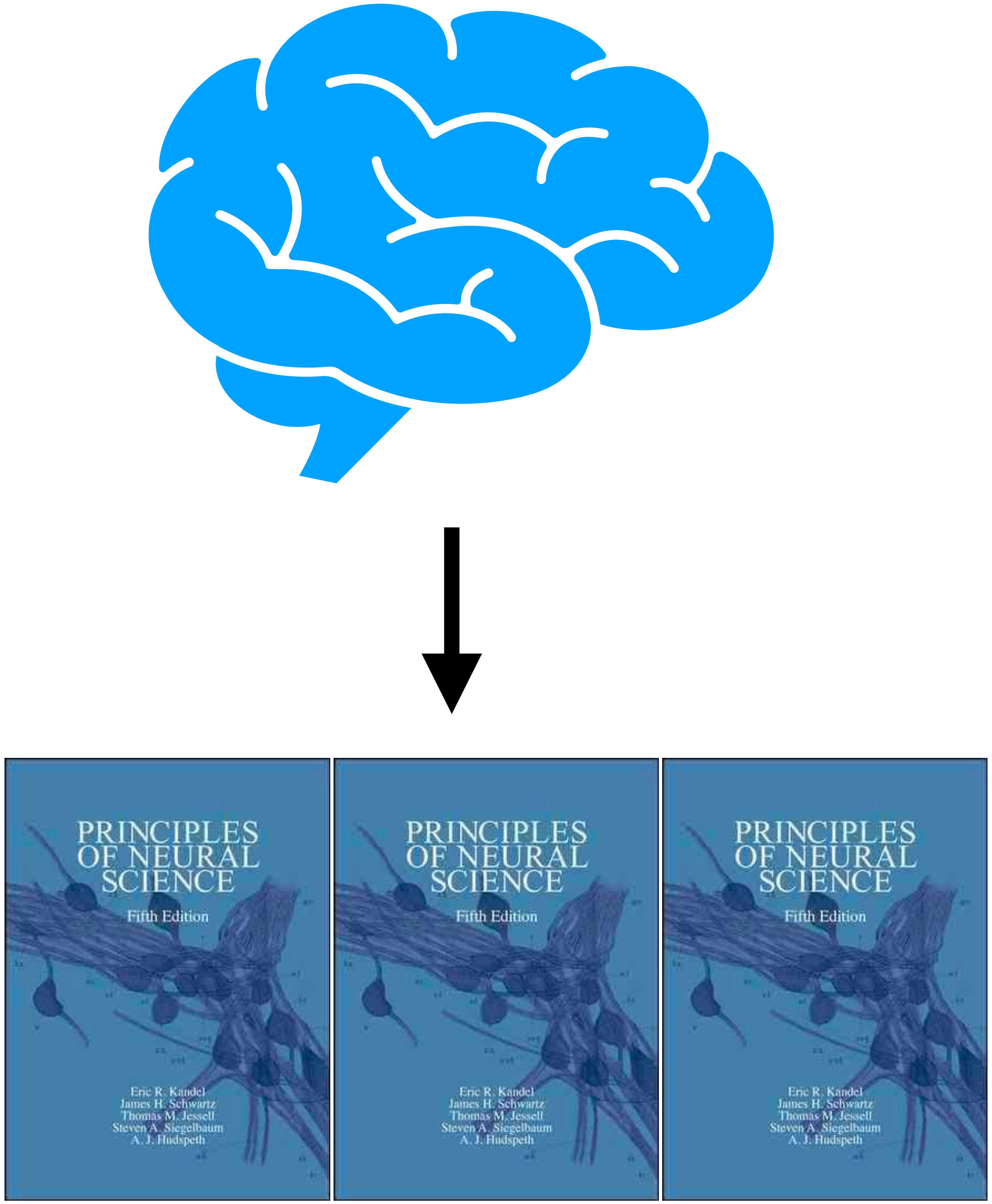




They are all real. Replicable from Go grand master to Go grandmaster.

The brain?

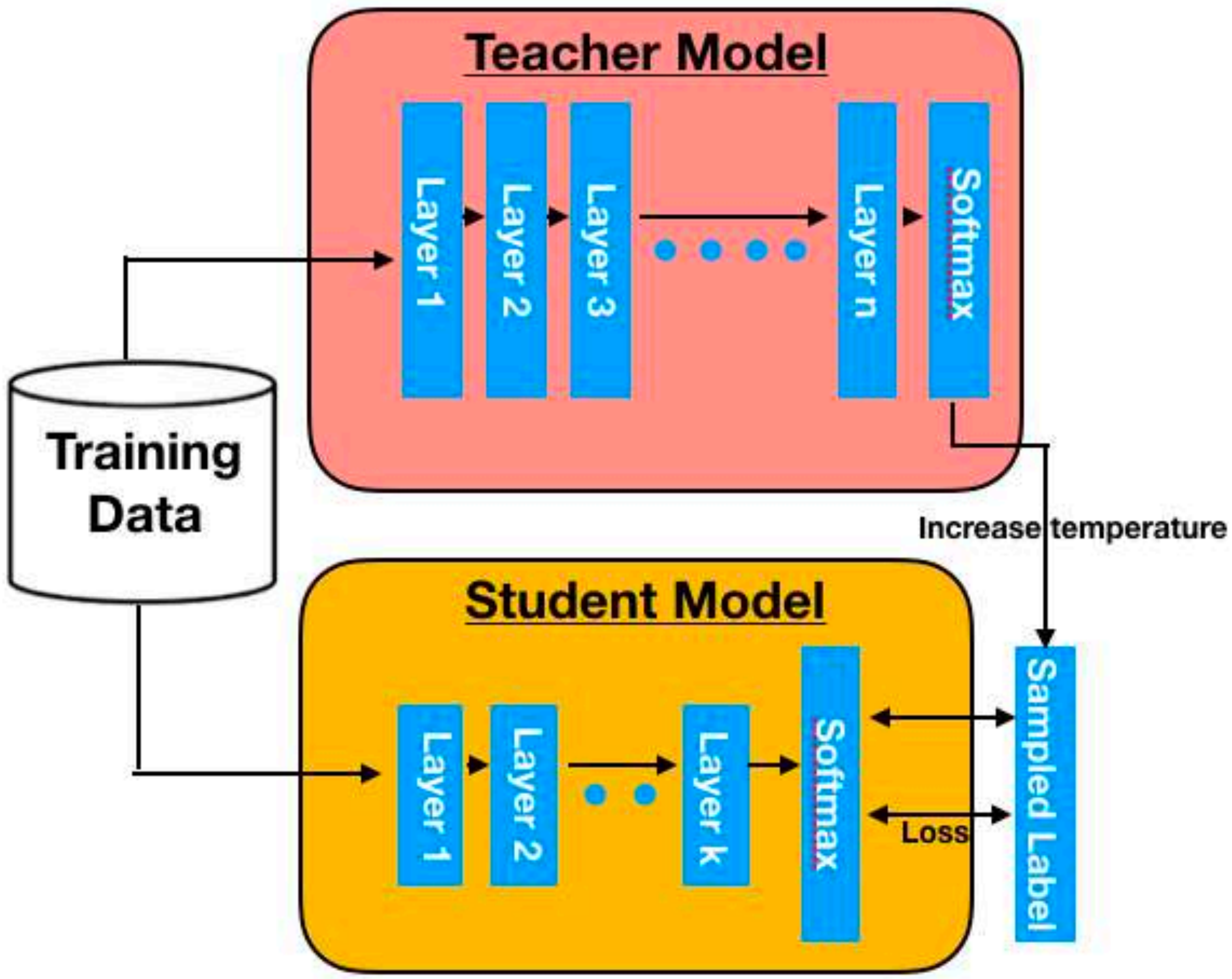
- Can play world class Go
    - as a (semi)hobby
- Recognize images
    - without convolution
- Dance



CS approaches to estimate compressibility

- Distillation
- Complexity calculations
- Back of the envelope calculations

Distillation



from mc.ai

Factor 10-100 on MNIST, imaginet

e.g. Ba and Caruana, Zhu et al 2018

Can we compress NNs?

- MNIST -> soft decision trees
    - BAD
- imagenet

Complexity calculations

- Many distinct ideas.

- e.g. Find which images in a training set do not help
    - Count how many do

Back of the envelope

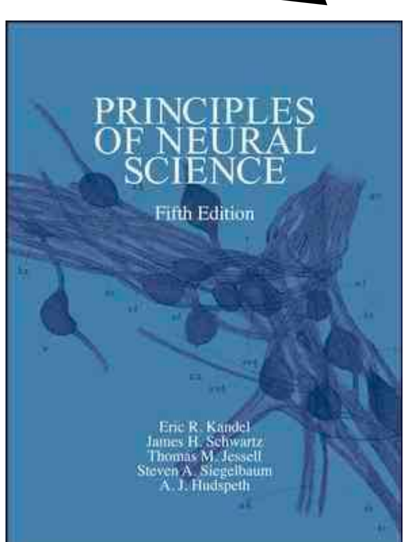
- 10 bits/s
- pi*10^8 seconds/a
- 30 years

- 10^11 bits
- 10^6 bits/book -> 10^5 books

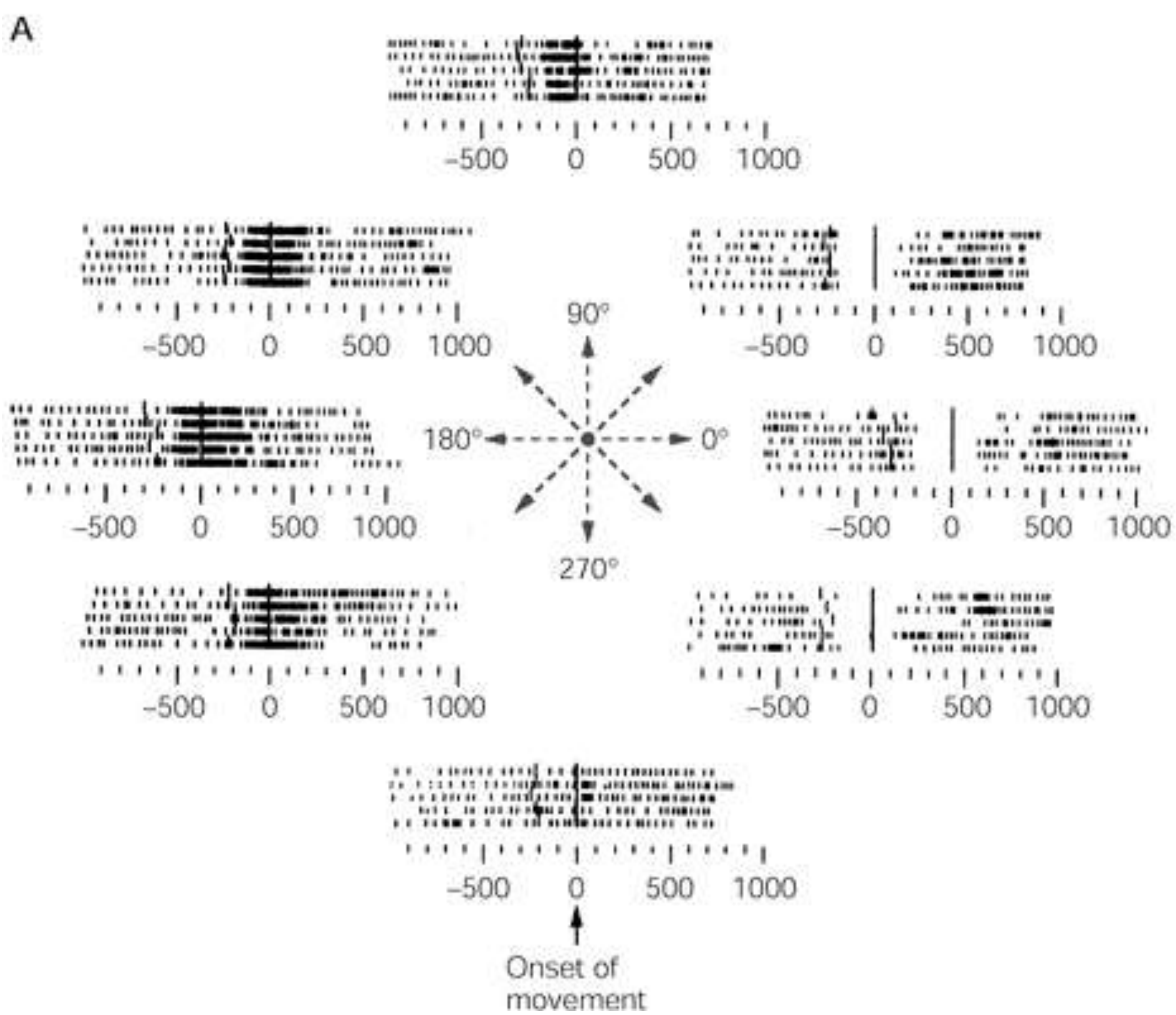
H(DNA)<<H(World)

- DNA: 2*3*10^9 nucleotides
    - mostly non-nervous system
    - of nervous system possibly much non-computational
    - very non-compressed
- Nurture >> Nature

Ok. So what if the brain is not compressible?

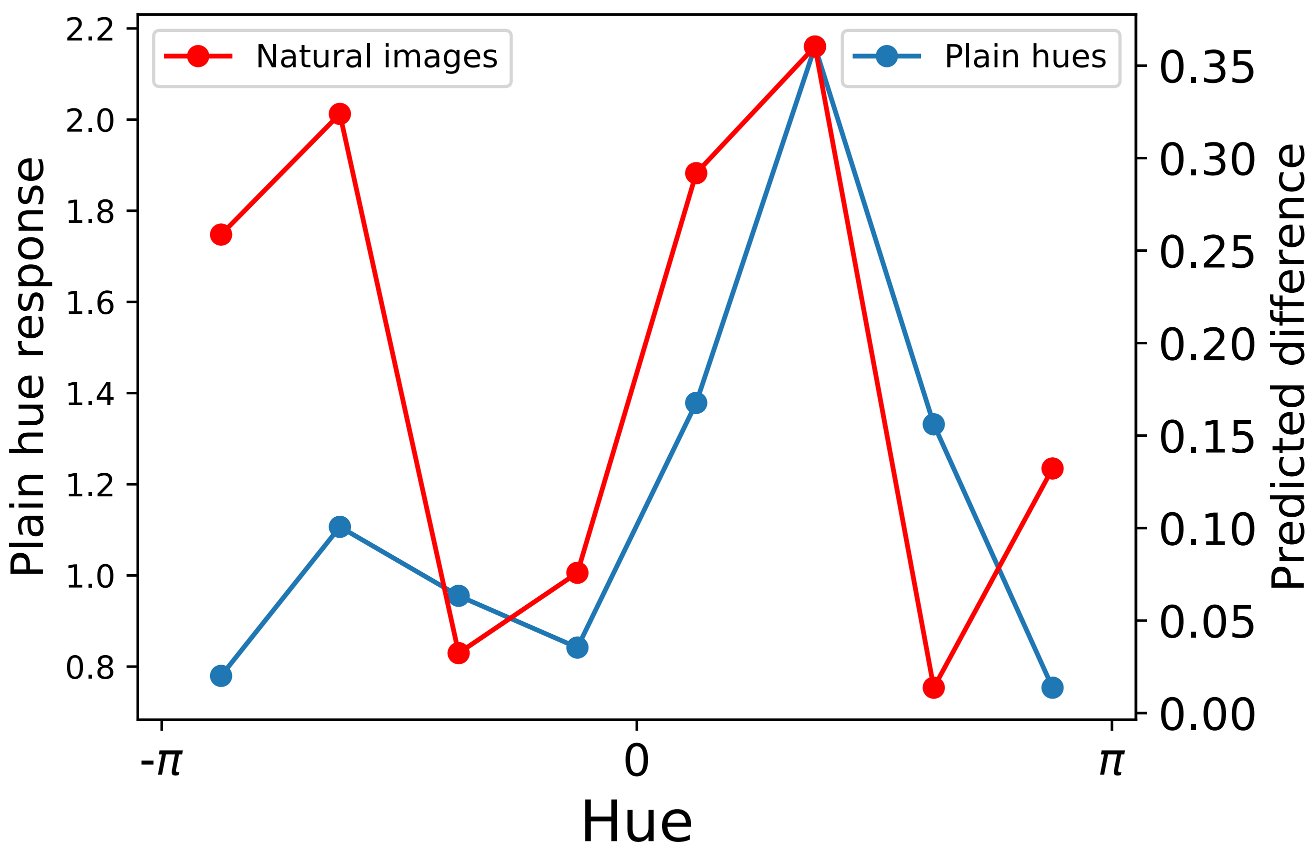


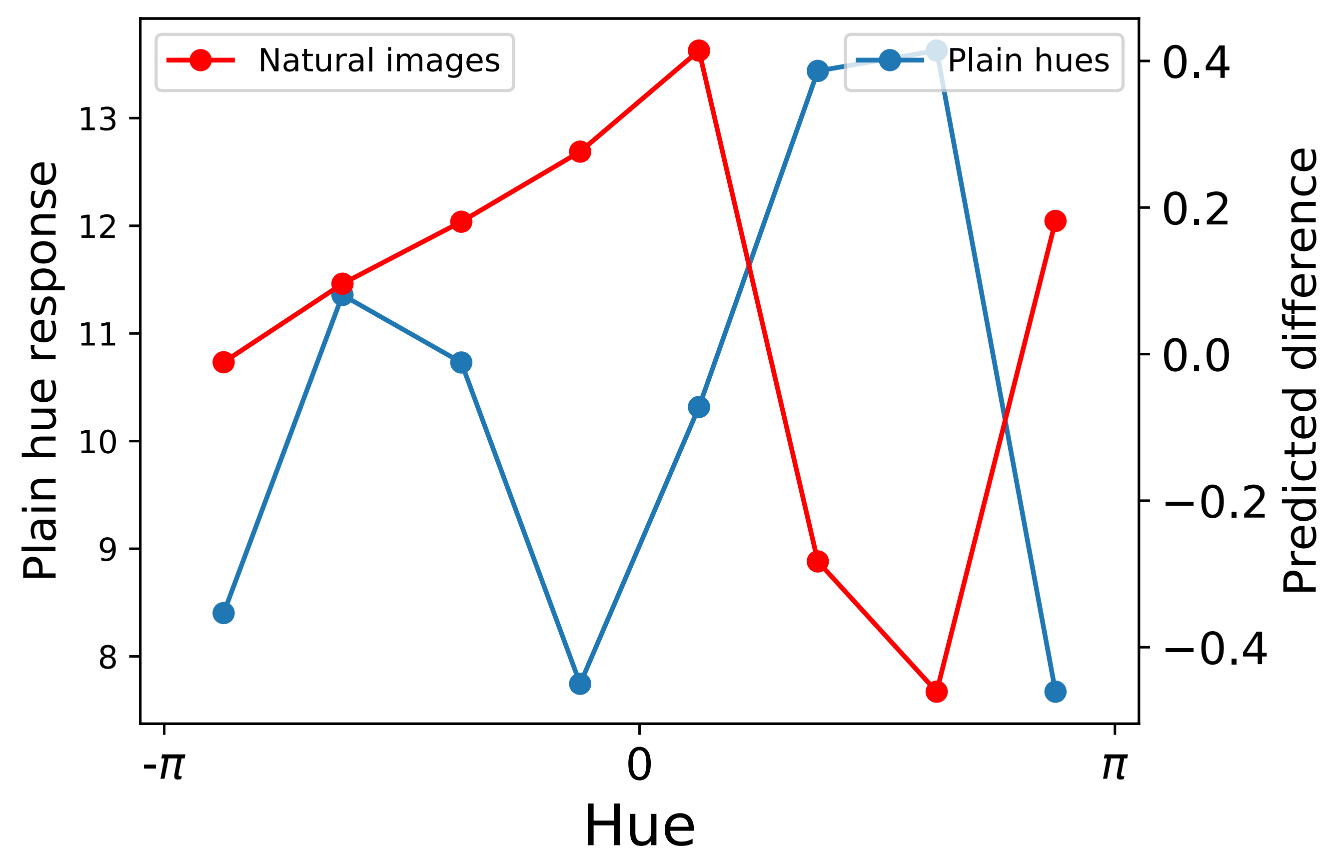
Tuning curves

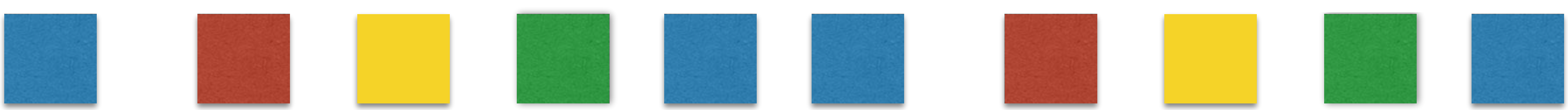


Georgeopoulos

No generalization







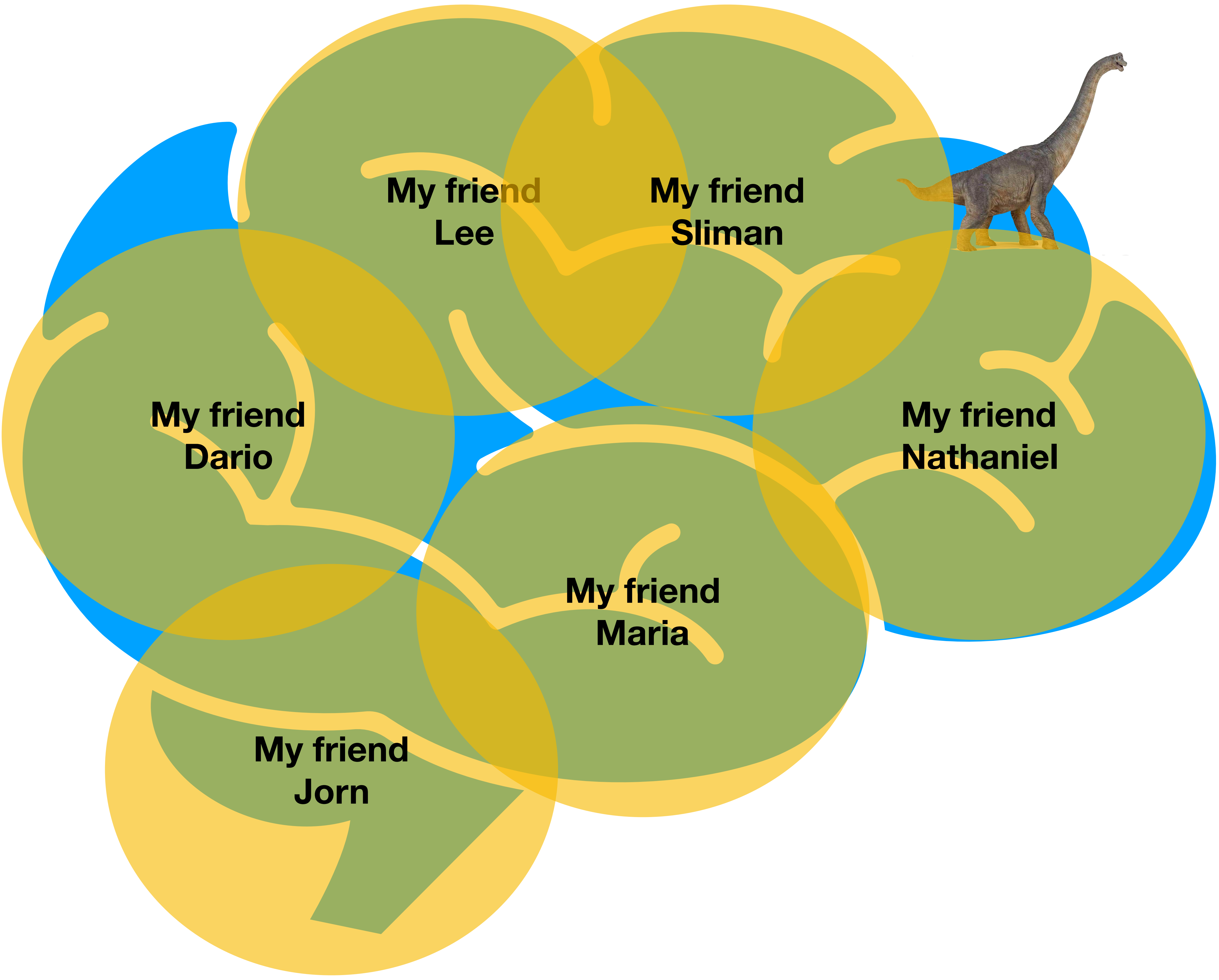
Color

Color

With Matt Smith

The dinosaur


My friend Dario


My friend Lee
My friend Sliman
My friend Dario
My friend Nathaniel
My friend Maria
My friend Jorn

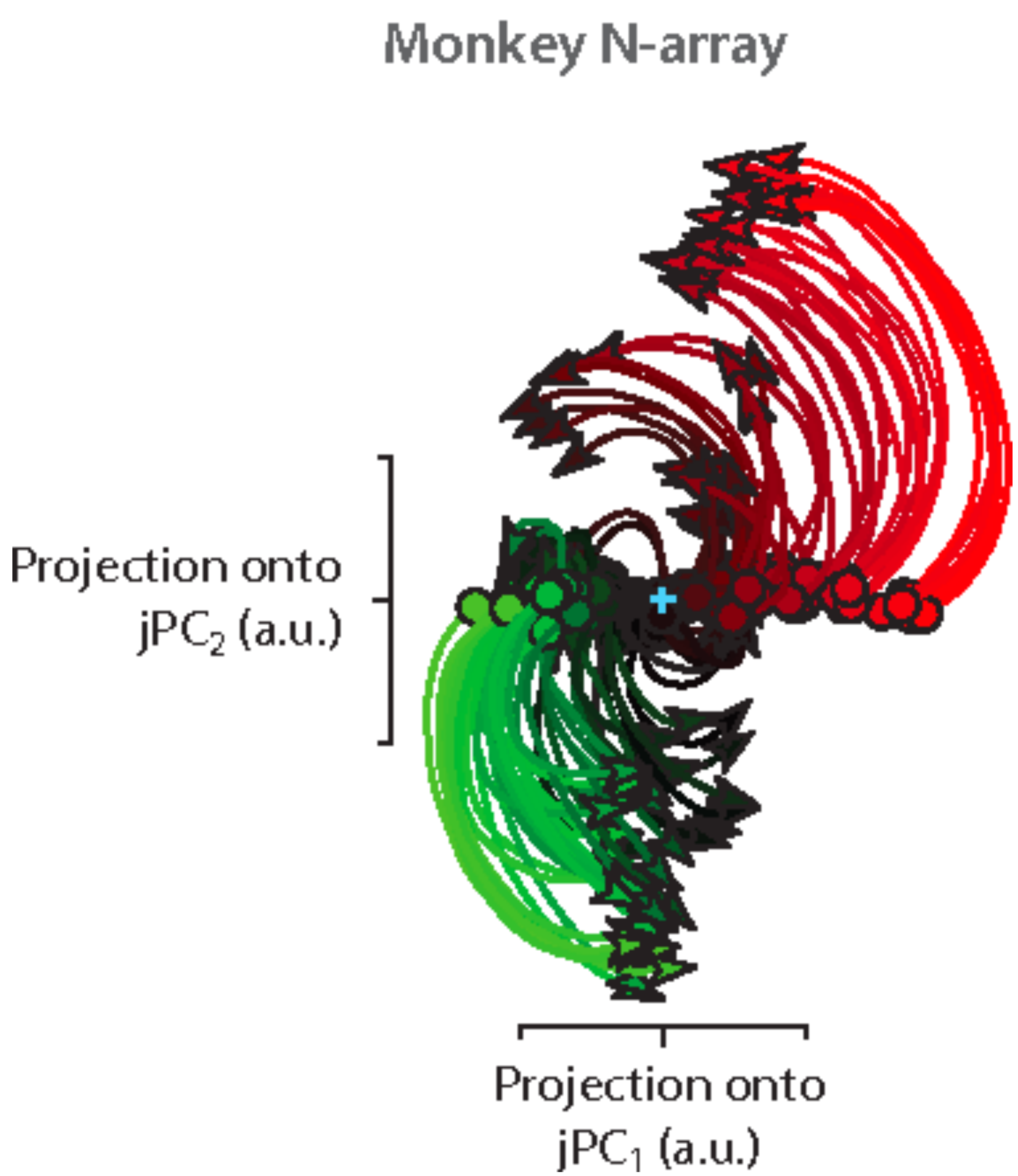
And in the end

Causality missing, the interesting things missing

Connectomics

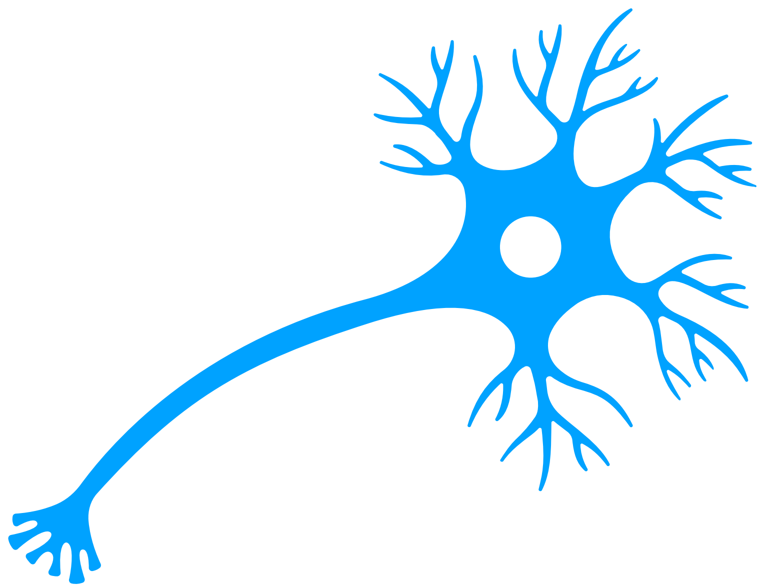
$$p(network) = p^M(1-p)^{\binom{n}{2}-M}$$

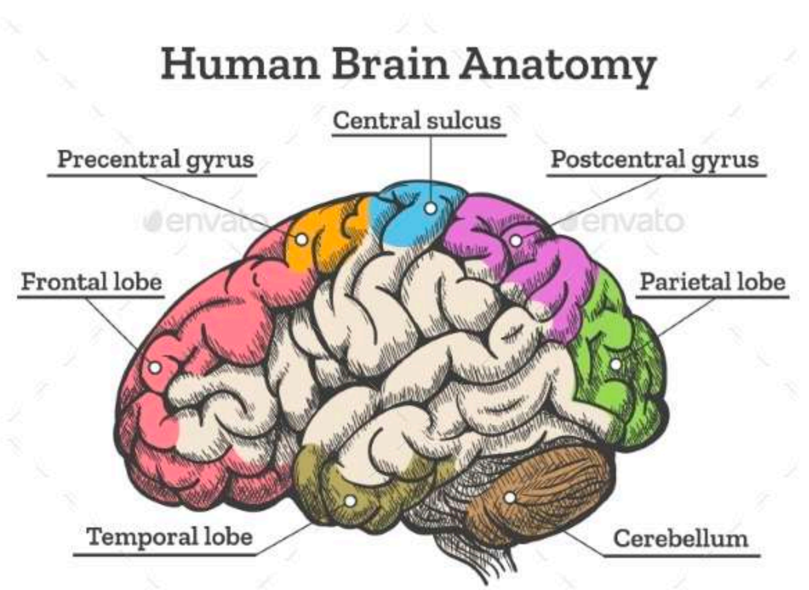
Dynamical systems



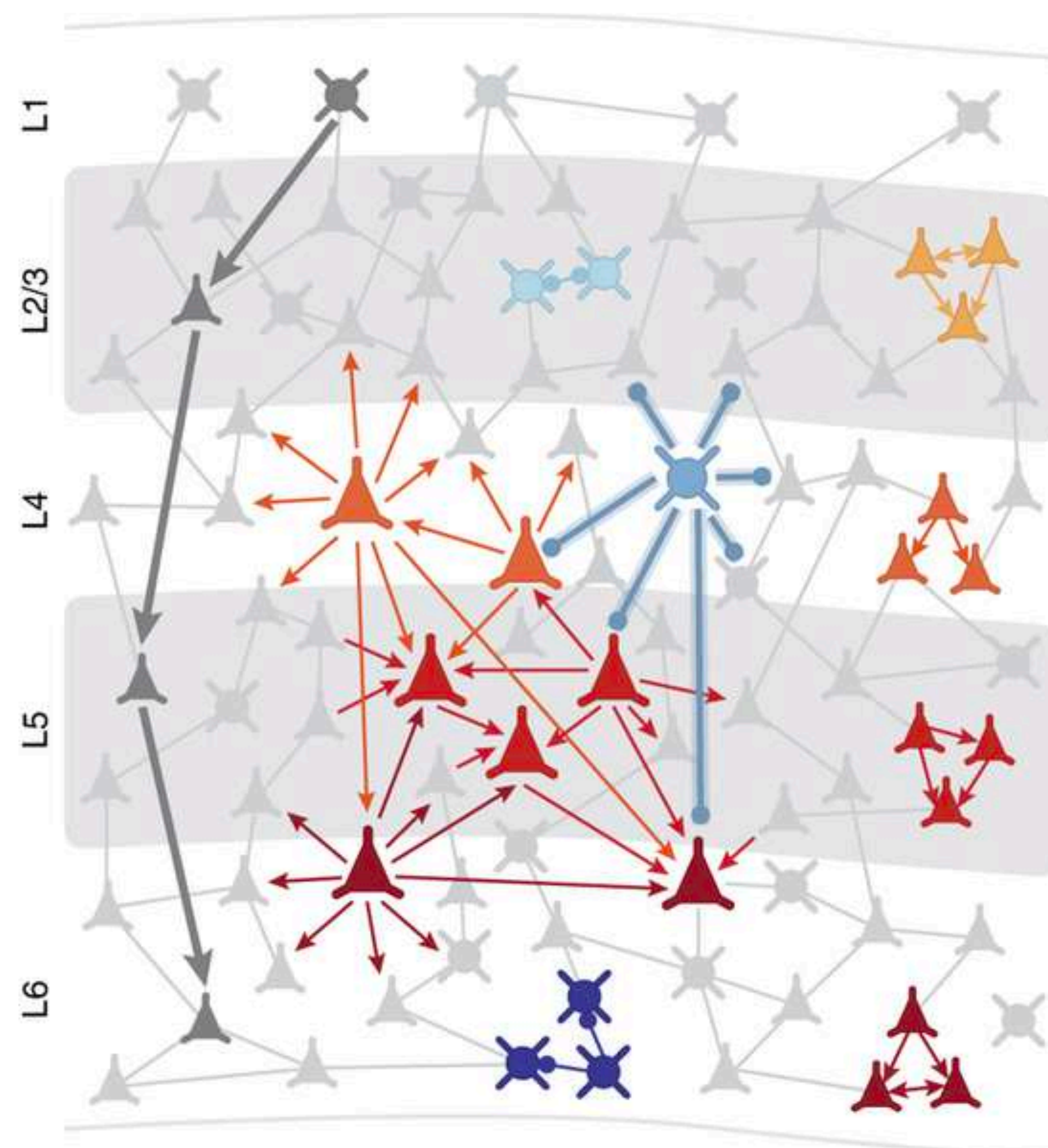
Brain not low-d, low-d description not understanding, causality missing

Neurons/ anatomy

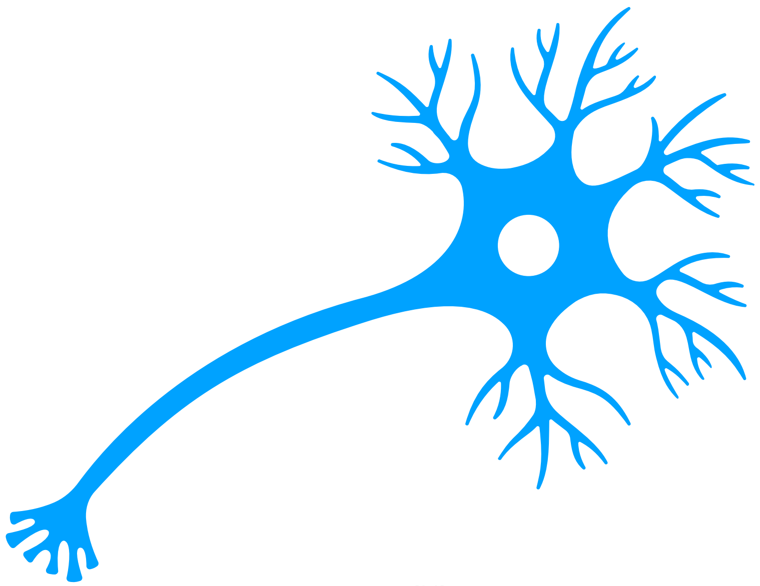


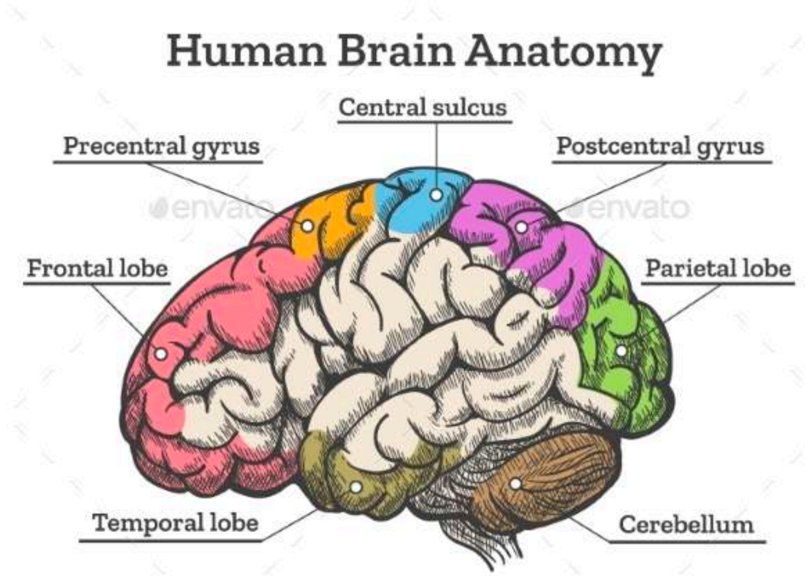


Markram style cell atlas



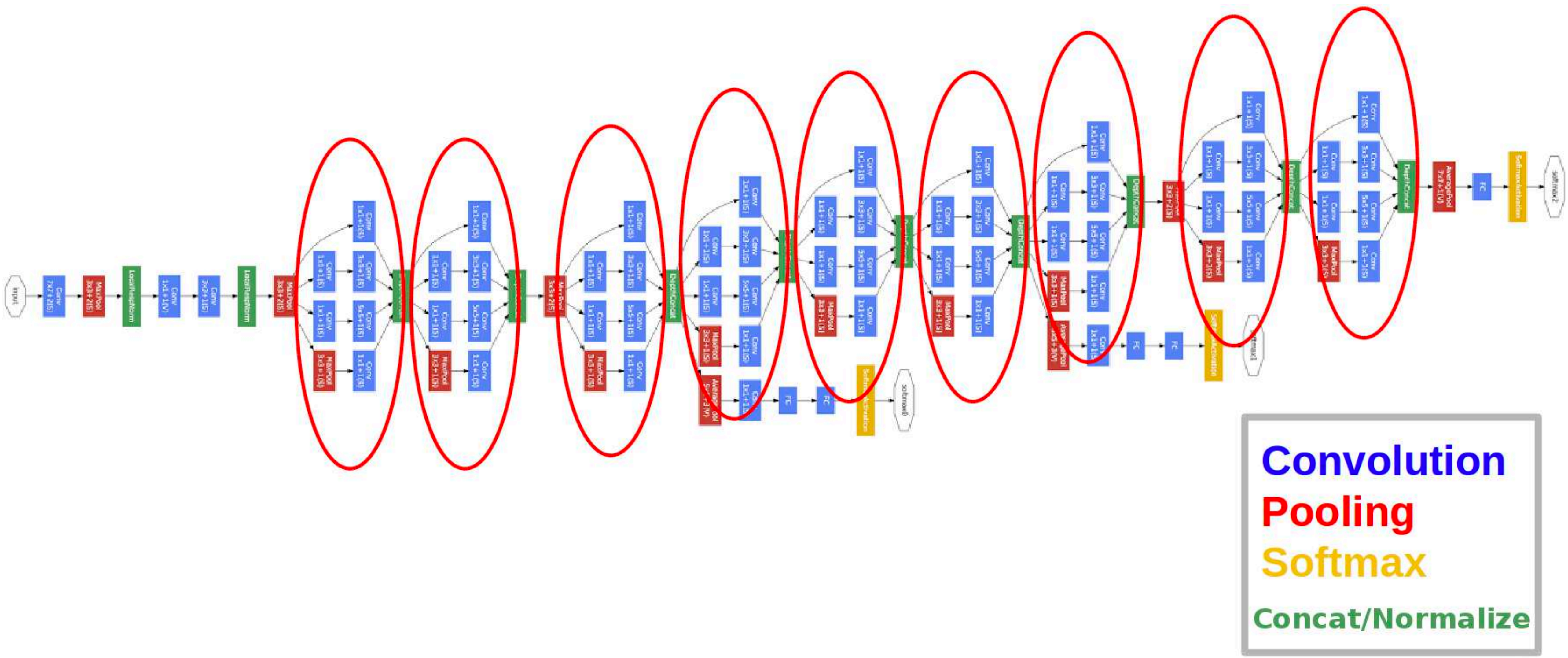
Learning centric





Learning Dynamics

Anatomy (Googlenet)



Objective function (softmax)

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\top \mathbf{w}_k}}$$

Optimizer (SGD)

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w)$$

$$w := w - \eta \nabla Q(w) = w - \eta \sum_{i=1}^{n} \nabla Q_i(w)/n$$