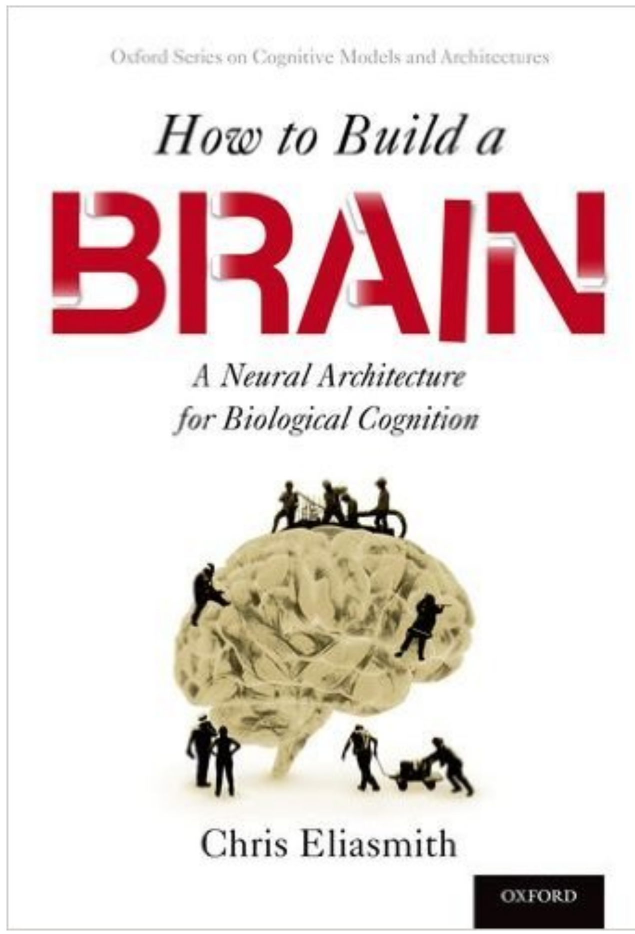
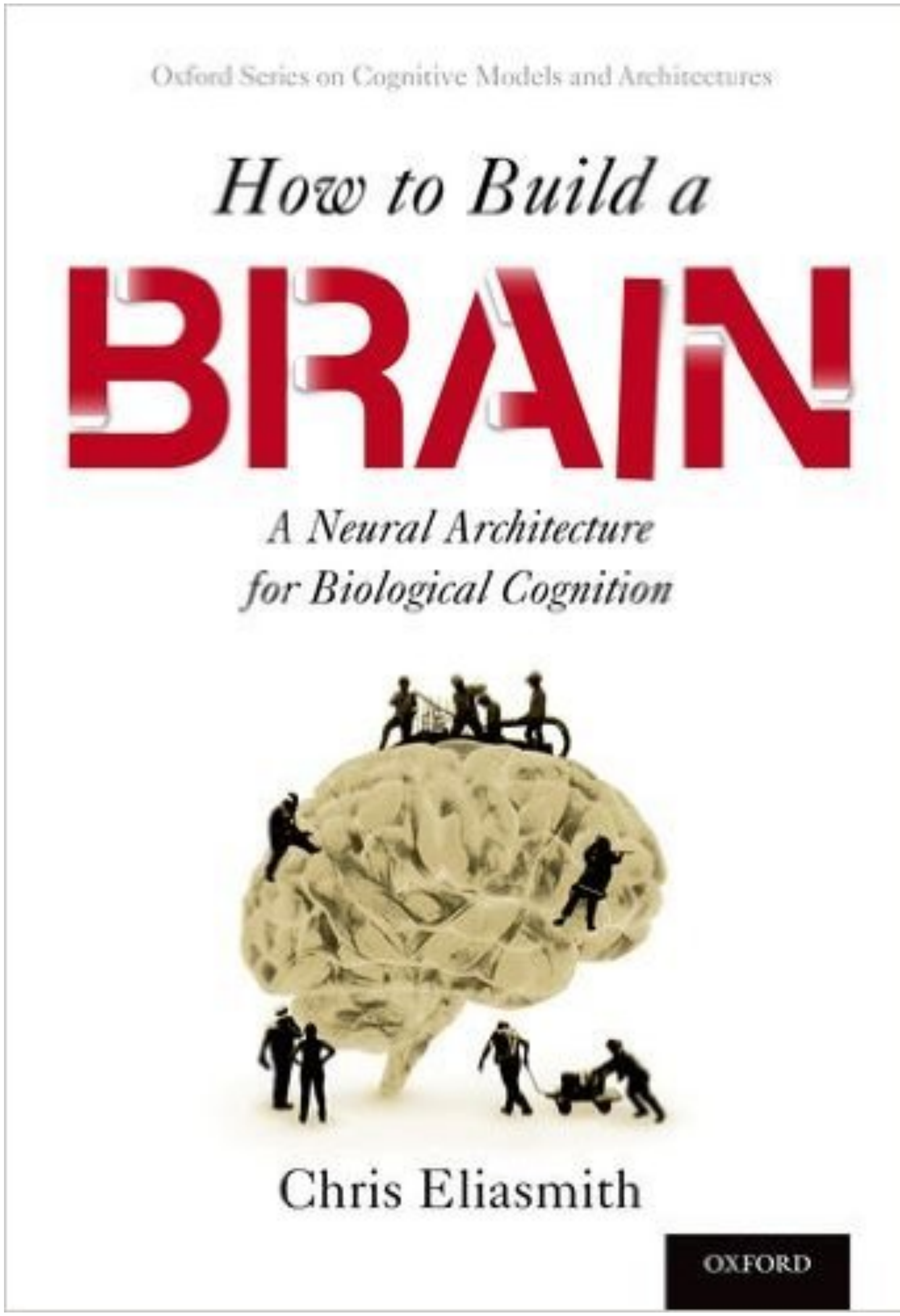
# Large-Scale Model Discussion

Odelia Schwartz

2017






A Large-Scale Model of the Functioning Brain
Chris Eliasmith et al.
Science 338, 1202 (2012)

Nengo software (Python and graphical Interface)

Large-scale neural simulations are becoming increasingly common [see (*1*) for a review]. These include the ambitious Blue Brain Project (*2*), which has simulated about 1 million neurons in cortical columns and includes considerable biological detail, accurately reflecting spatial structure, connectivity statistics, and other neural properties. More recent work has simulated many more neurons, such as the 1 billion neurons simulated in the Cognitive Computation Project (*3*), which has been hailed as a cat-scale simulation. A human-scale simulation of 100 billion neurons has also been reported (*4*).

Although impressive scaling has been achieved, no previous large-scale spiking neuron models have demonstrated how such simulations connect to a variety of specific observable behaviors.

Unfortunately, simulating a complex brain alone does not address one of the central challenges for neuroscience: explaining how complex brain activity generates complex behavior.

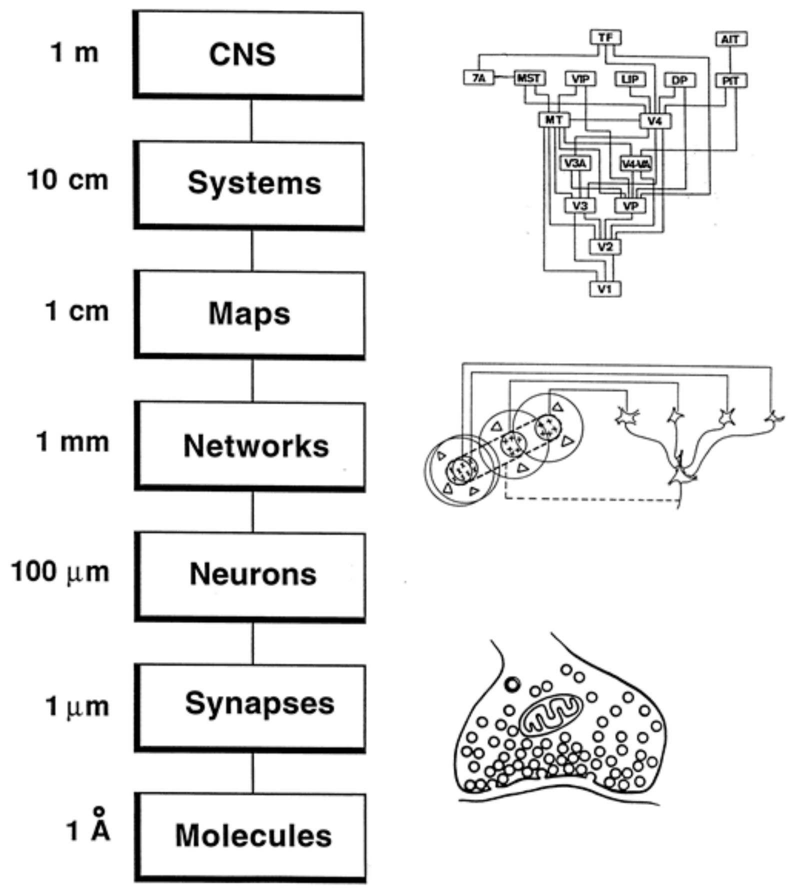



Diagram: Terrence Sejnowski

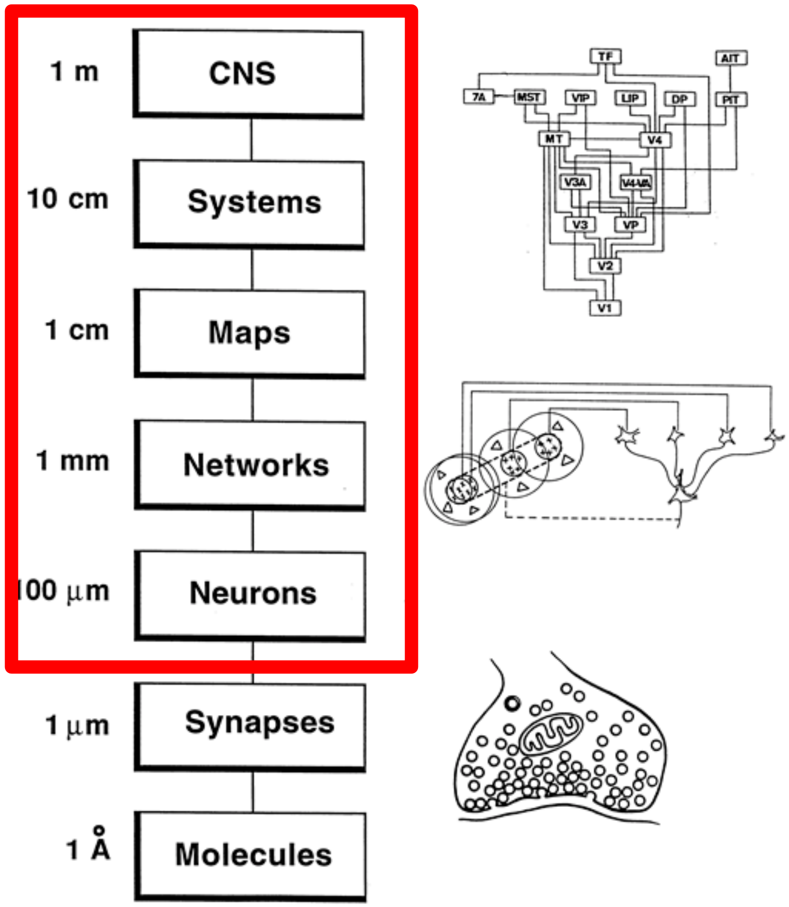



Diagram: Terrence Sejnowski

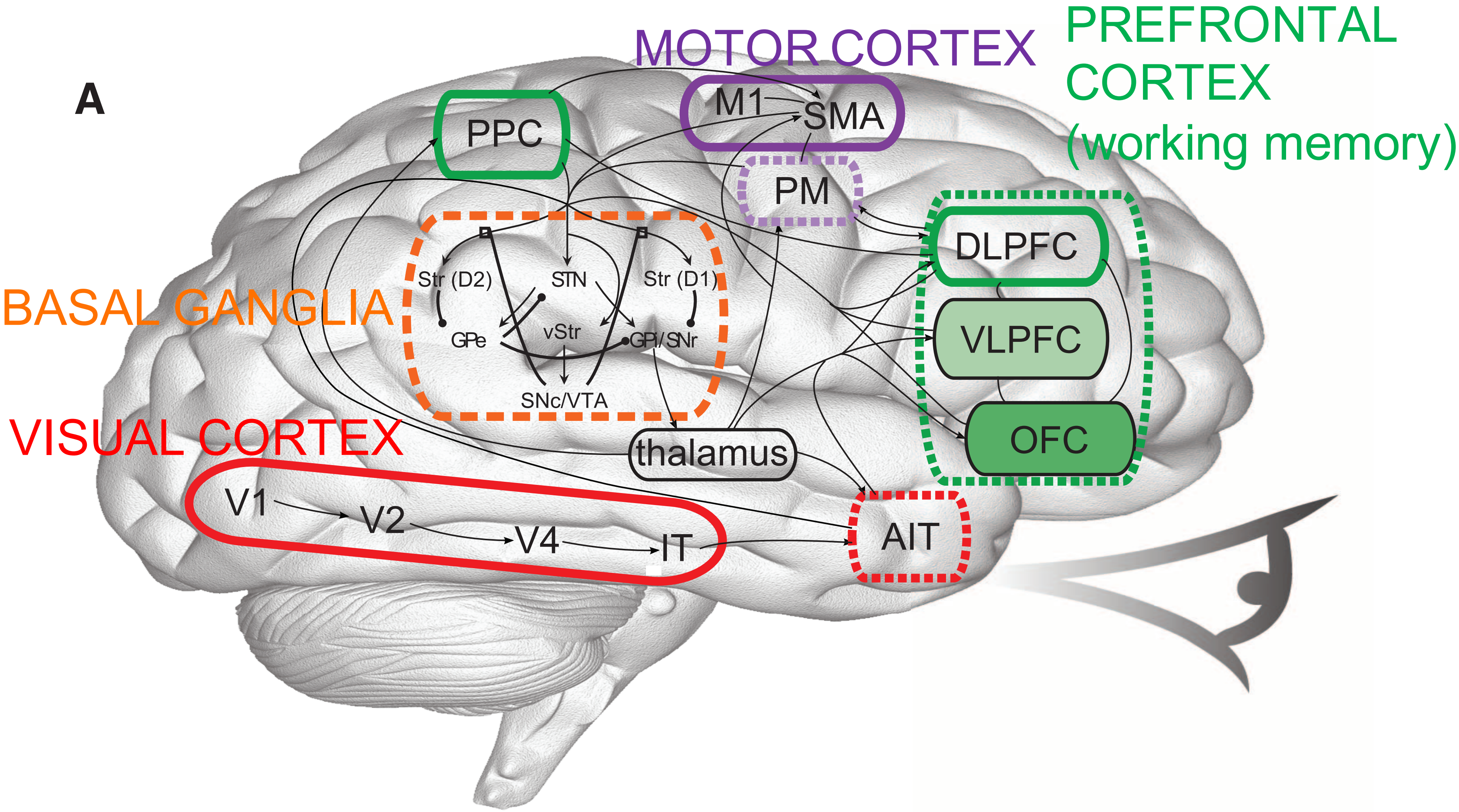
- Bridge gap between complex behaviors and complex neural activity
- Performs whole set of simulated tasks associated with human cognition
- Large scale: 2.5 million neurons
- Uses spiking models of neurons
- Summarizes a lot of papers/work in lab

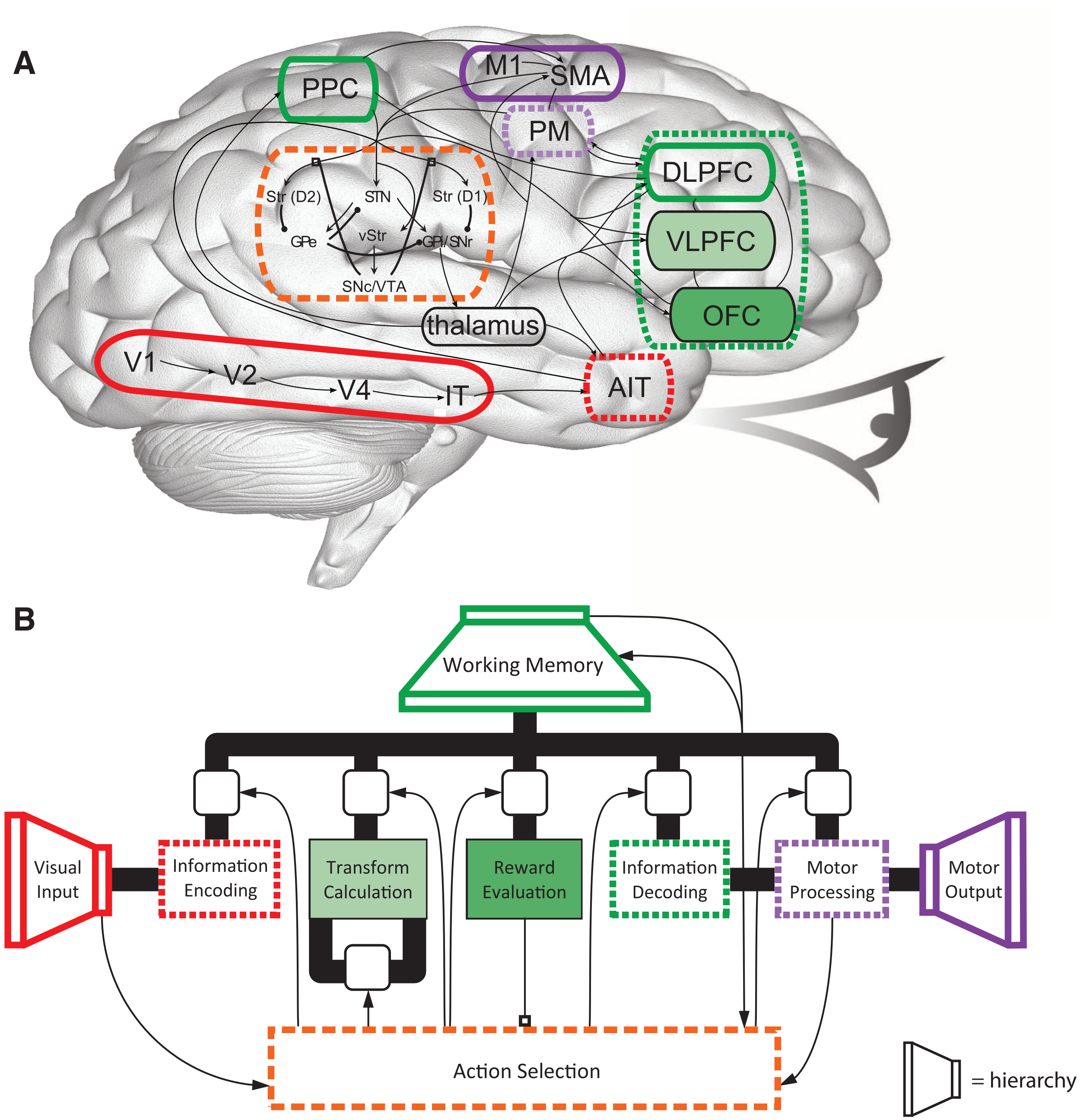
- Inputs to model: 28 by 28 images of typed characters

- Outputs: movements of physically modeled arm

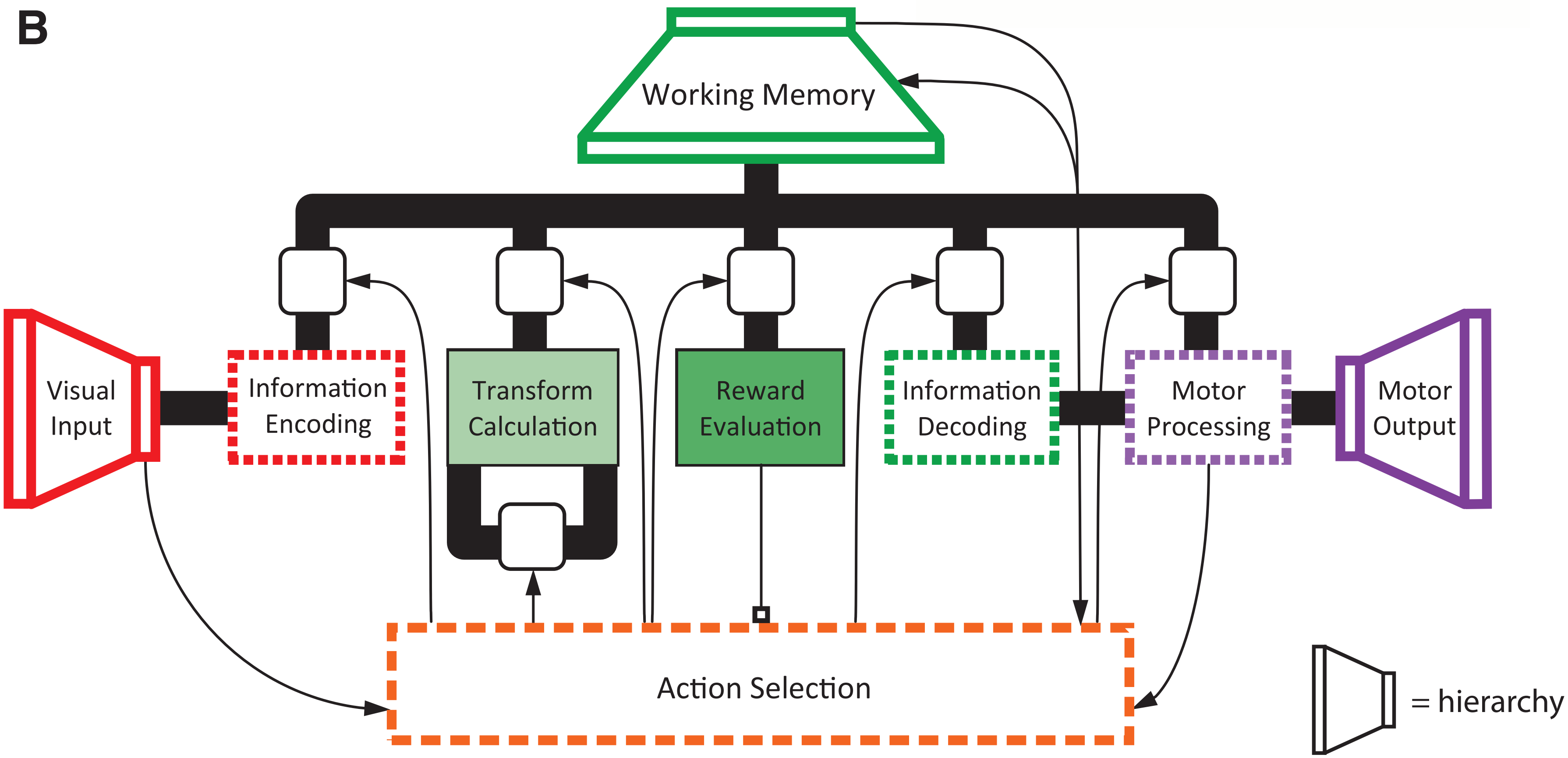
- 8 tasks (some modeled more extensively in their other papers)

- Refer to model as SPAUN (Semantic Pointer Architecture Unified Network)


A
MOTOR CORTEX
PREFRONTAL
CORTEX
(working memory)
PPC
M1
SMA
PM
DLPFC
VLPFC
OFC
BASAL GANGLIA
Str (D2)
STN
Str (D1)
GPe
vStr
GPi/SNr
SNc/VTA
thalamus
VISUAL CORTEX
V1
V2
V4
IT
AIT


A
PPC
M1
SMA
PM
DLPFC
VLPFC
OFC
Str (D2)
STN
Str (D1)
GPe
vStr
GPi/SNr
SNc/VTA
thalamus
V1
V2
V4
IT
AIT
B
Working Memory
Visual Input
Information Encoding
Transform Calculation
Reward Evaluation
Information Decoding
Motor Processing
Motor Output
Action Selection
= hierarchy

B


Working Memory
Visual Input
Information Encoding
Transform Calculation
Reward Evaluation
Information Decoding
Motor Processing
Motor Output
Action Selection
= hierarchy

- Spiking neurons
- Form compressed representation

The specific compression hierarchies in Spaun are (see Fig. 1B): (i) a visual hierarchy, which compresses image input into lower-dimensional firing patterns; (ii) a motor hierarchy that decompresses firing patterns in a low-dimensional space to drive a simulated arm; and (iii) a WM, which constructs compressed firing patterns to store serial position information.

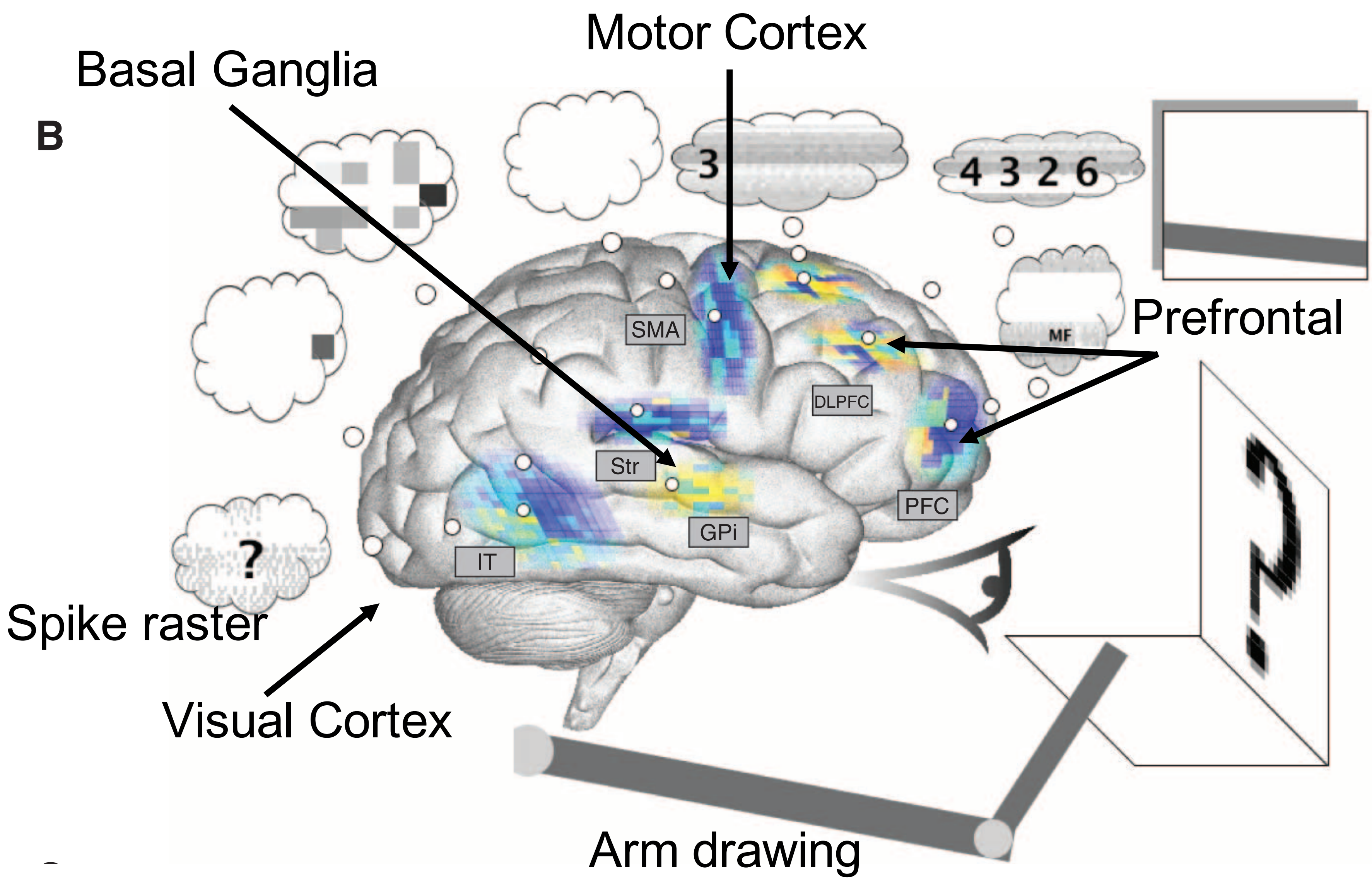
- Spiking neurons
- Form compressed representation
- 2.5 hours to simulate 1 second of data !

https://www.youtube.com/watch?v=g2HHJfovb5E

TEDx talk

https://www.youtube.com/watch?v=dKaqFz_WoIw&feature=youtu.be

Intro video



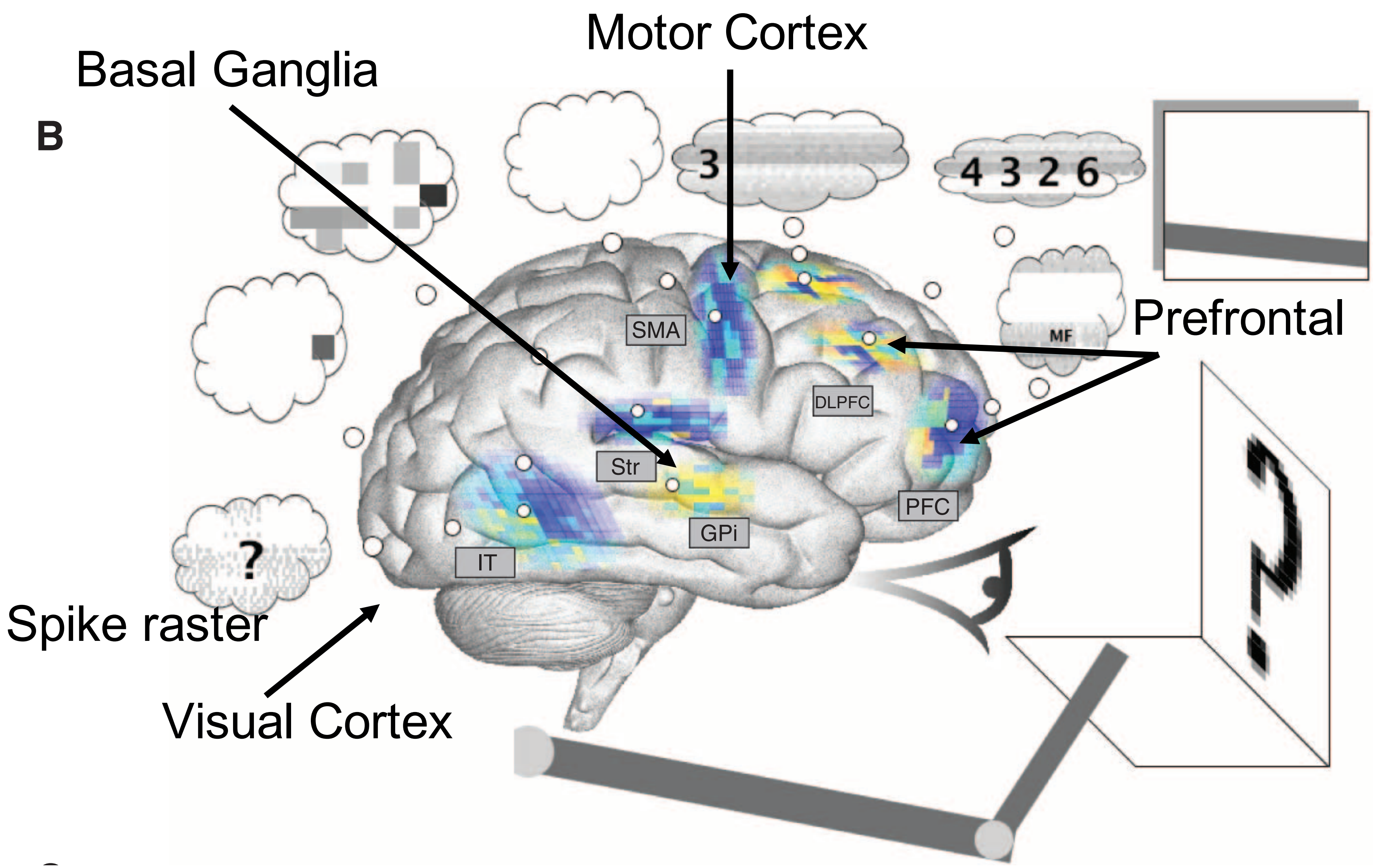

red highly active; blue low

https://youtu.be/vuGDYajWyhU

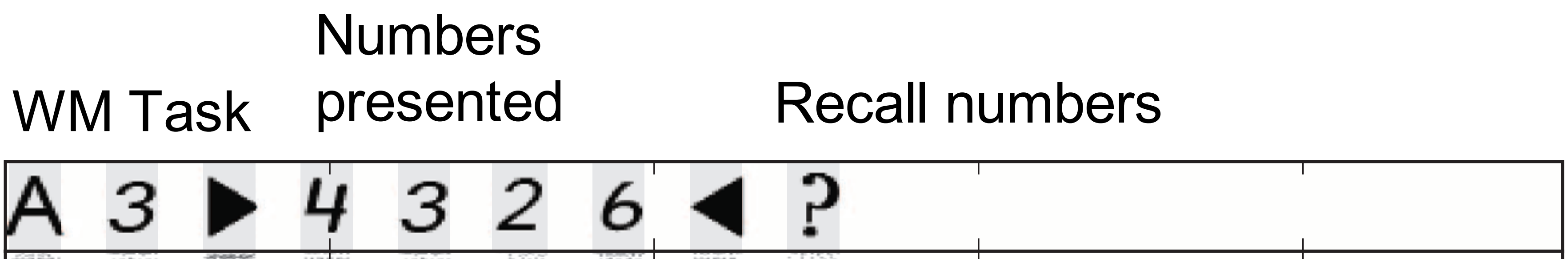
Reinforcement learning task

https://youtu.be/XxIzmkWygjY

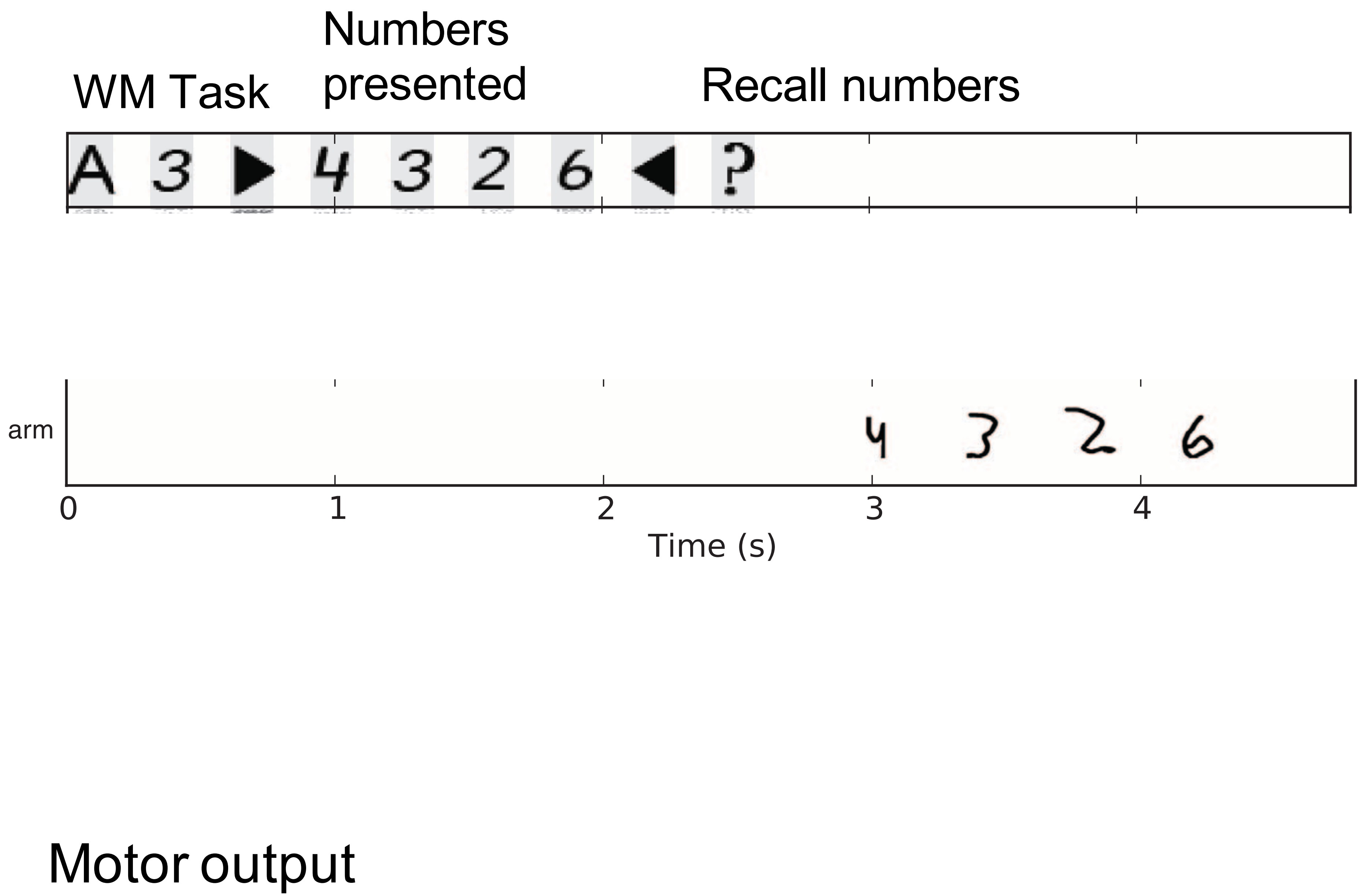
Serial working memory task



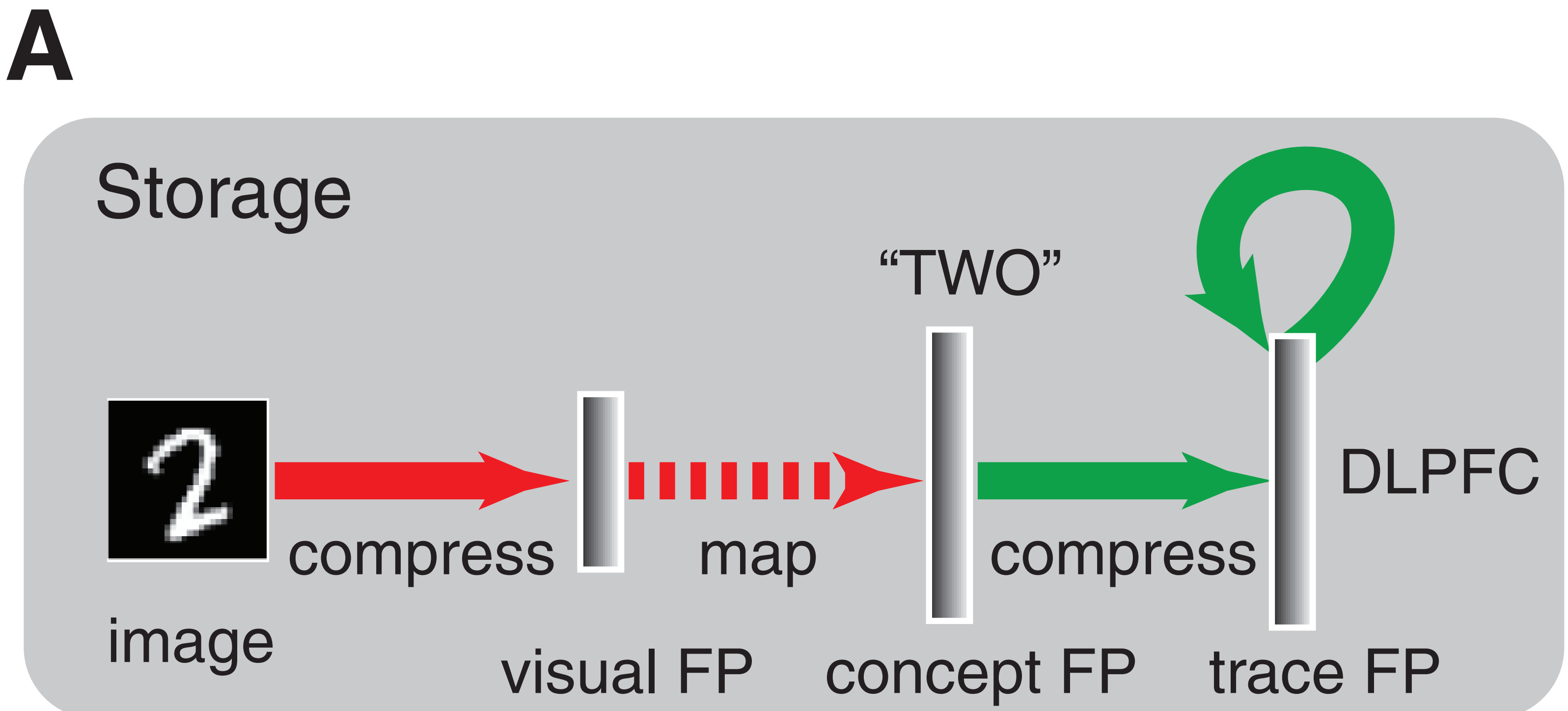

Serial working memory task (red highly active; blue low)




Visual input: A3 indicated working memory task

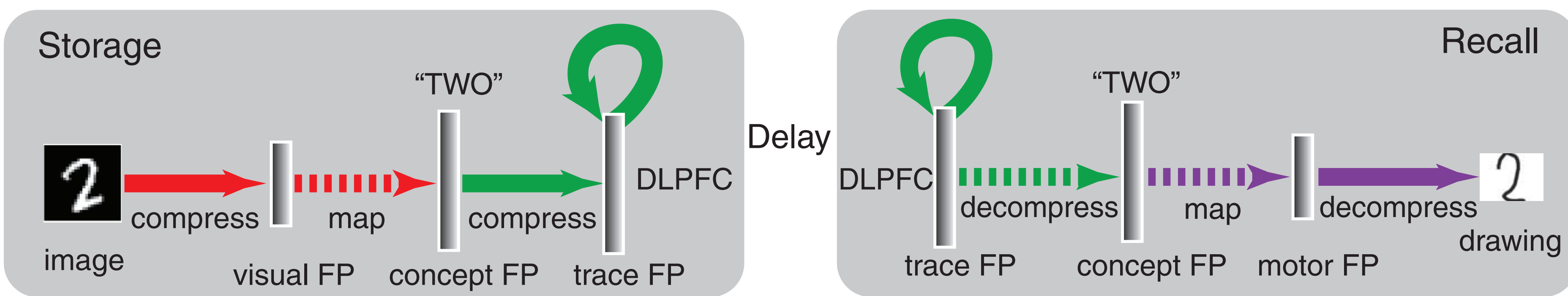


Numbers
presented
WM Task
Recall numbers
A 3 4 3 2 6 ?
arm
4 3 2 6
0
1
2
3
4
Time (s)
Motor output

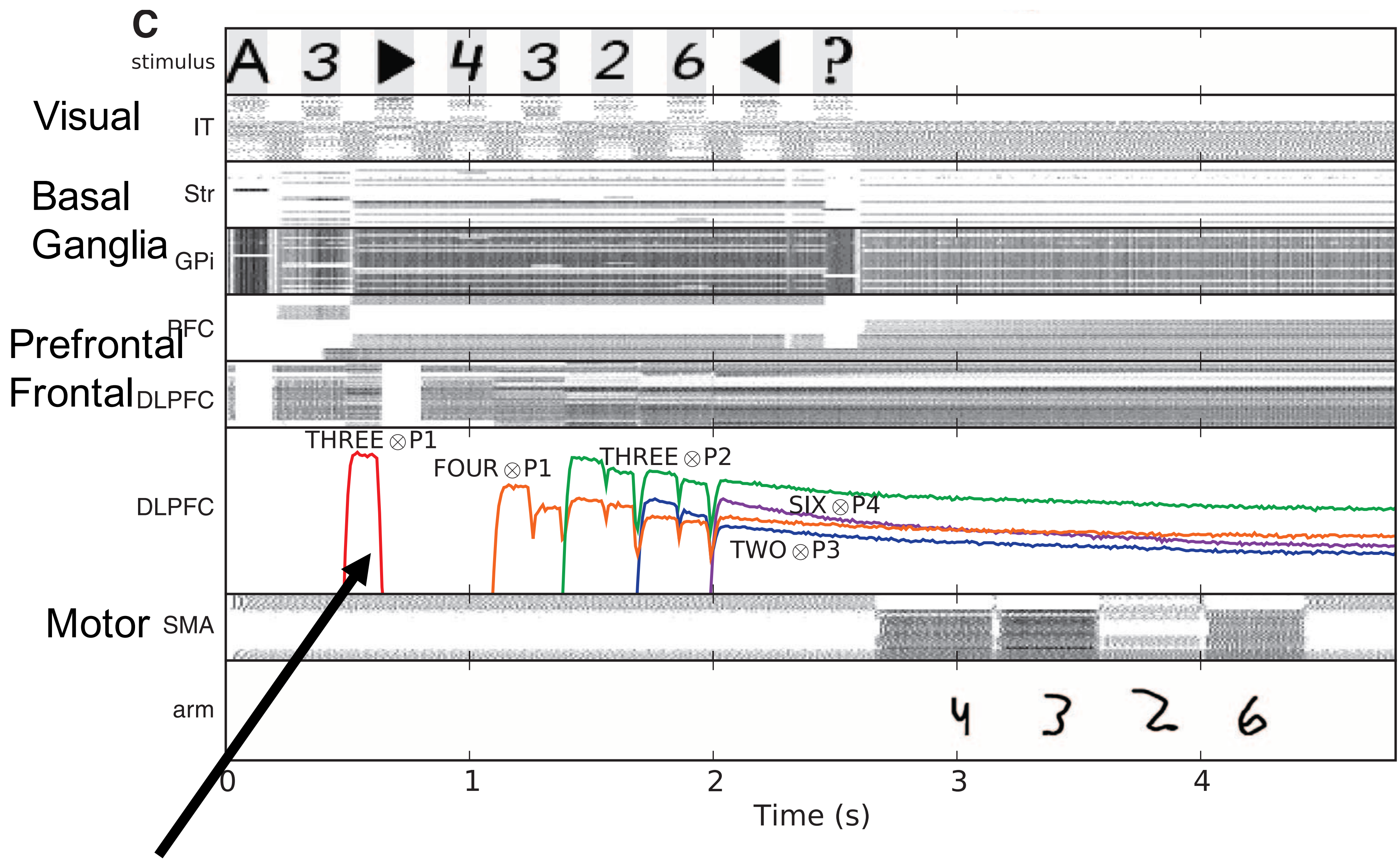



Information flow through Spaun during the WM task.
Storage in memory. FP = Firing pattern

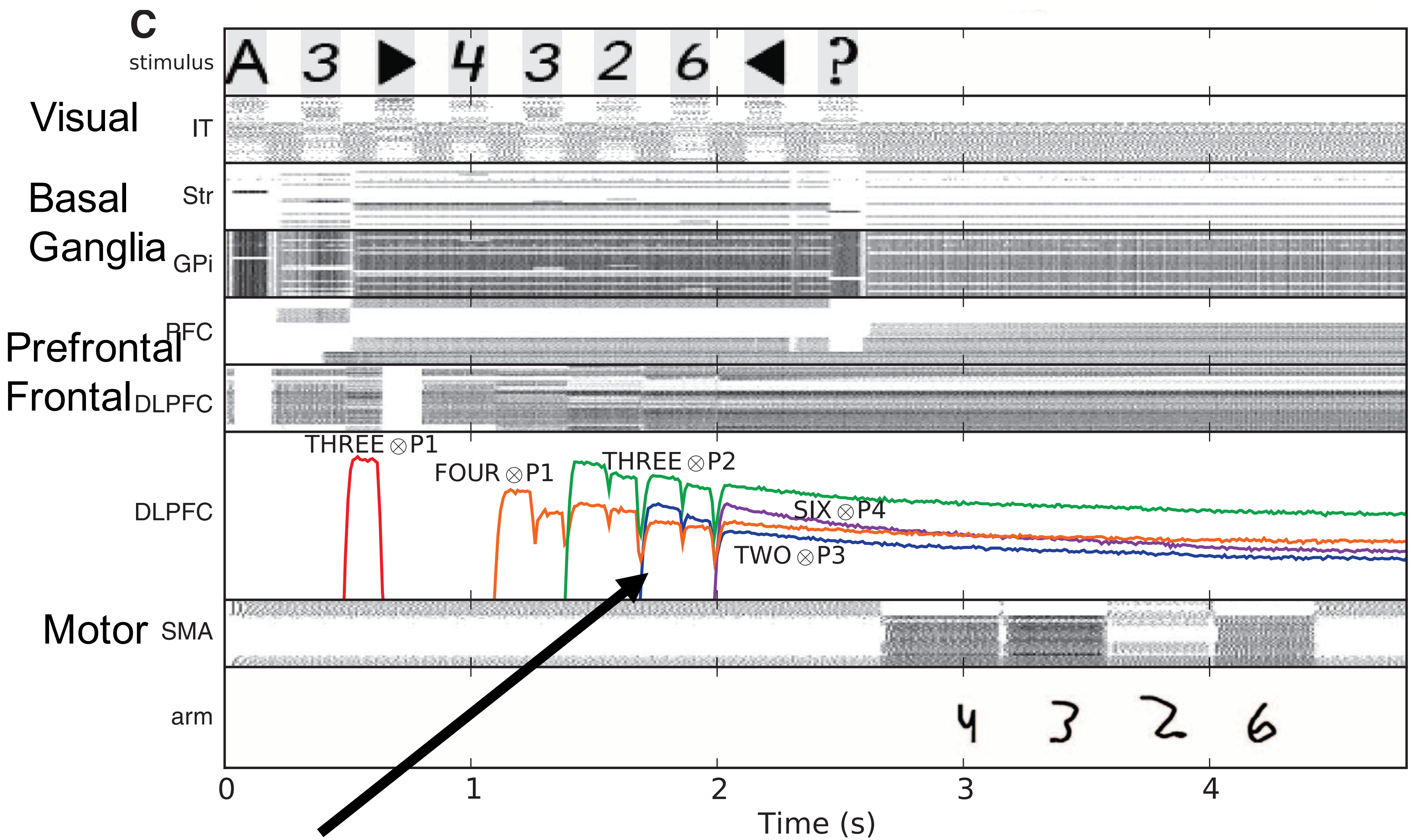
A



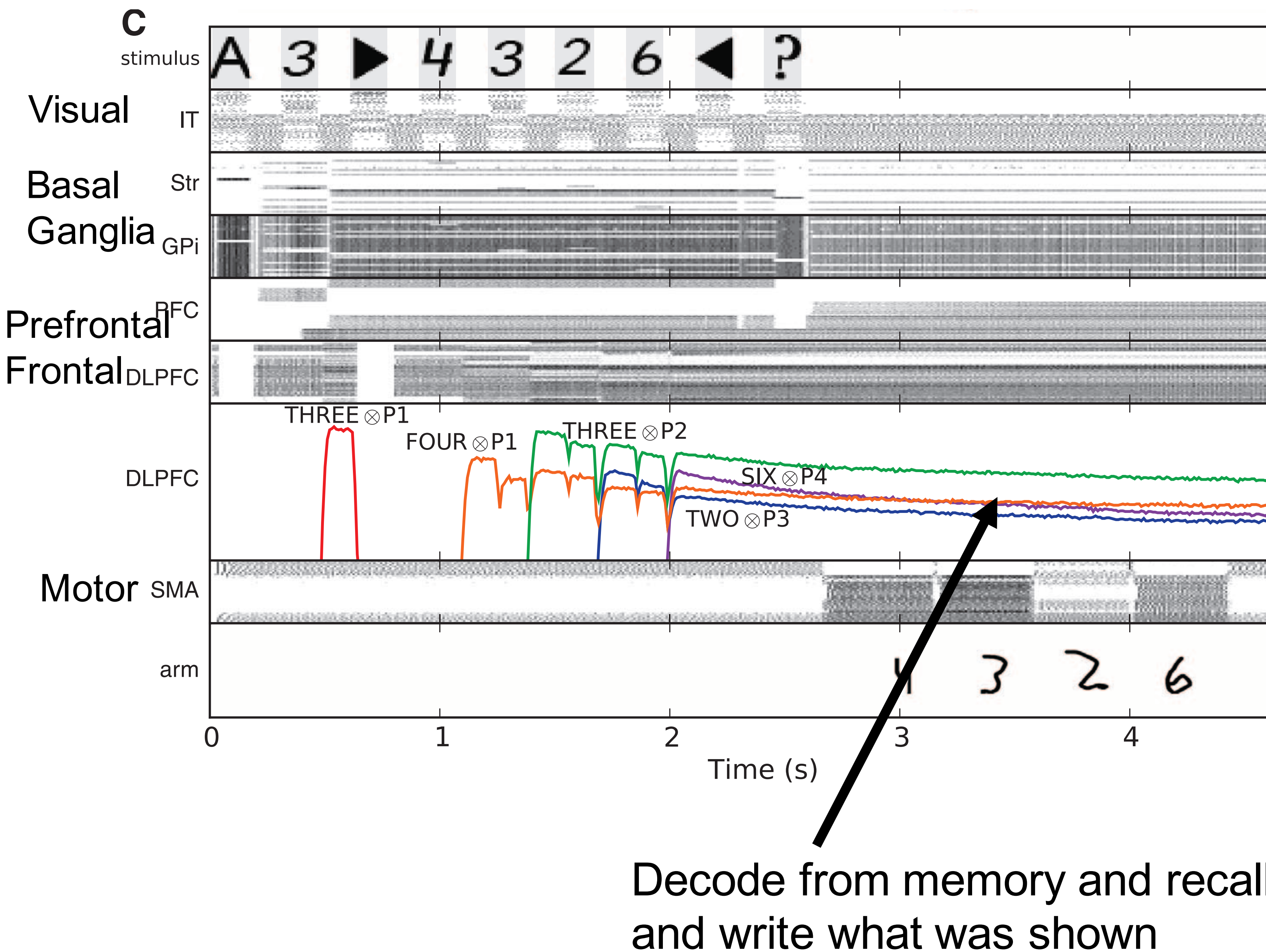

Information flow through Spaun during the WM task.
Storage in memory. FP = Firing pattern
Delay = delay during task (need working memory)

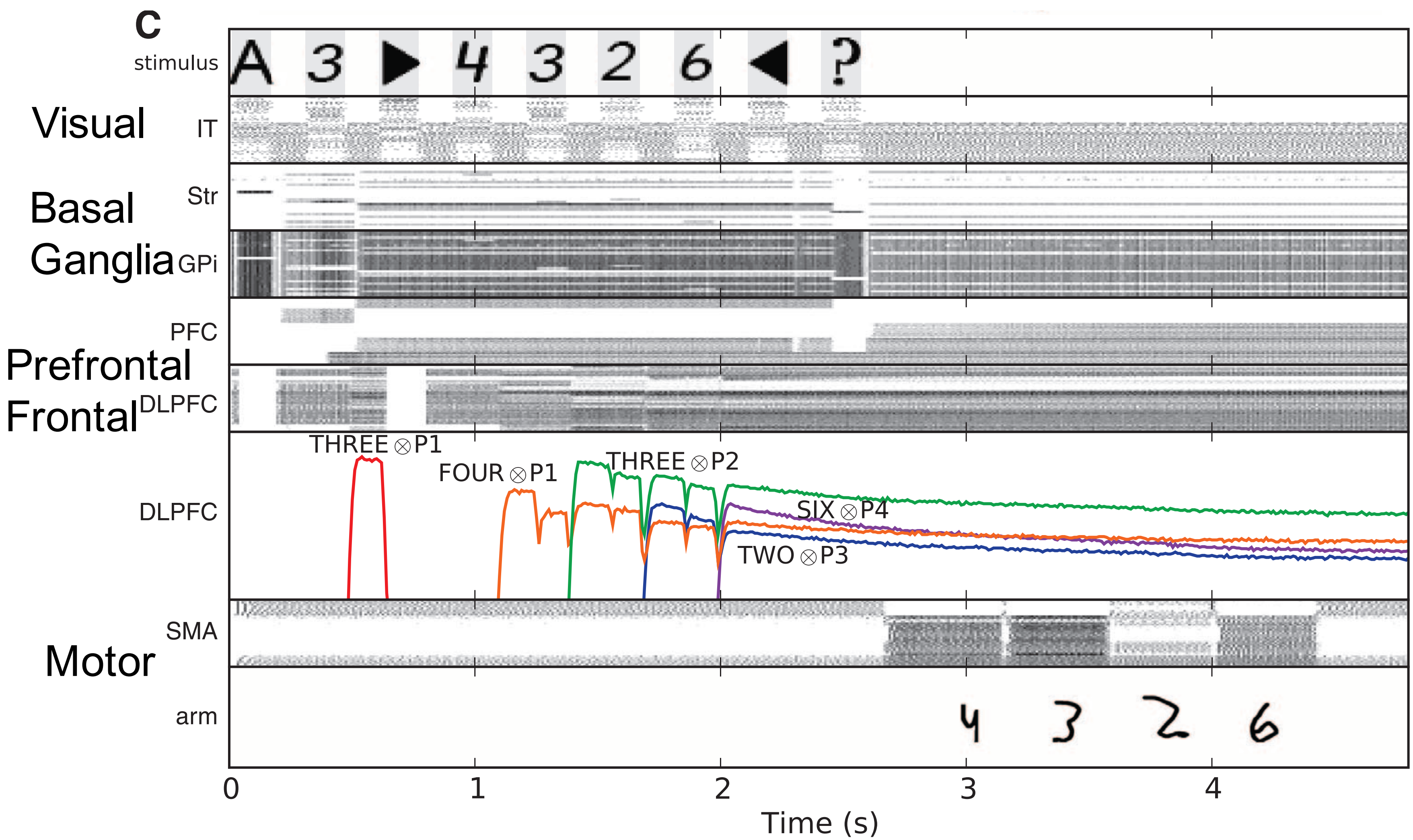



A3 = Switch Spaun into WM task




Bind digits to positions in memory


c
stimulus
A 3 4 3 2 6 ?
Visual
IT
Basal Ganglia
Str
GPi
Prefrontal Frontal
PFC
DLPFC
THREE ⊗ P1
FOUR ⊗ P1
THREE ⊗ P2
SIX ⊗ P4
TWO ⊗ P3
DLPFC
Motor
SMA
arm
4 3 2 6
0
1
2
3
4
Time (s)
Decode from memory and recall and write what was shown




Similarity plots (solid colored lines) show the dot product (i.e., similarity) between the decoded representation from the spike raster plot and concepts in Spaun's vocabulary. These plots provide a conceptual decoding of the spiking activity

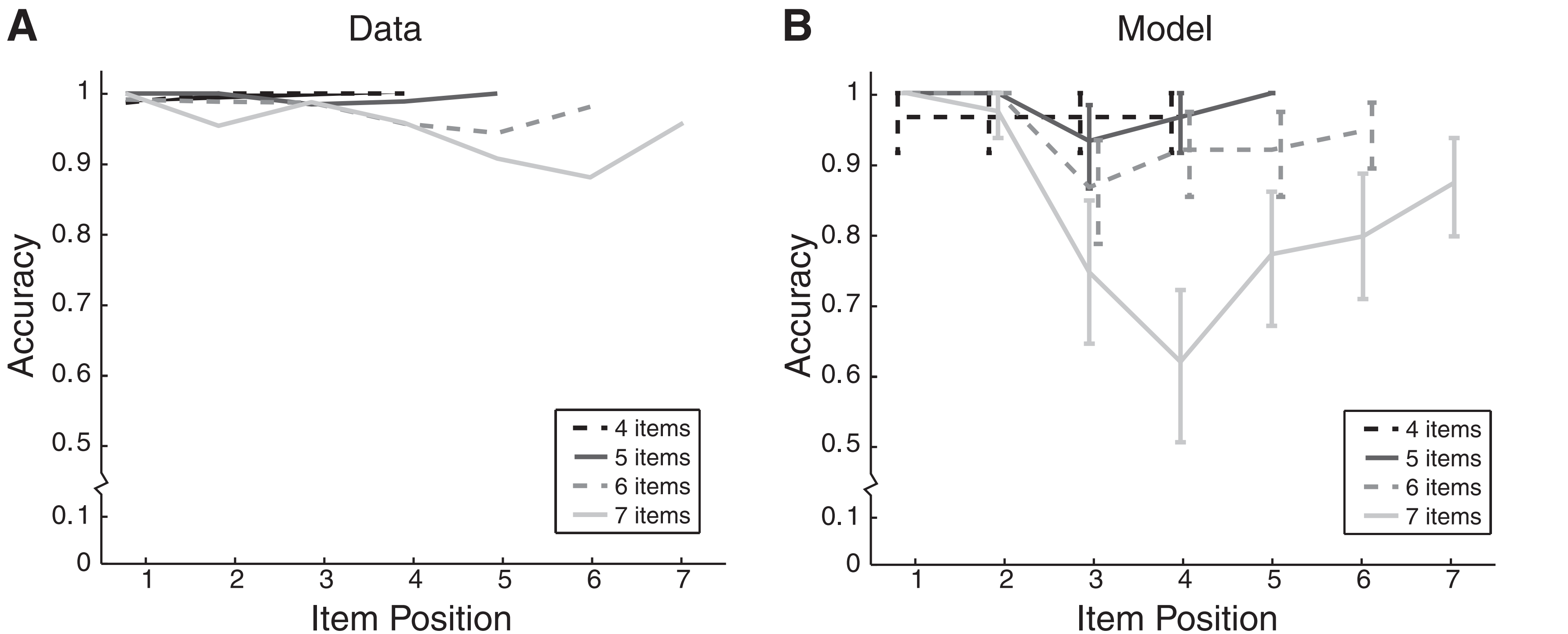



**Fig. 4.** Population-level behavioral data for the WM task. Accuracy is shown as a function of position and list length for the serial WM task. Error bars are 95% confidence intervals over 40 runs per list length. (**A**) Human data taken from (*18*) (only means were reported). (**B**) Model data showing similar primacy and recency effects.

Things learned earlier and later remembered better; and effect of number items; have papers with better compatibility – here chose data best matches the Spaun task

# SPAUN: main approaches used

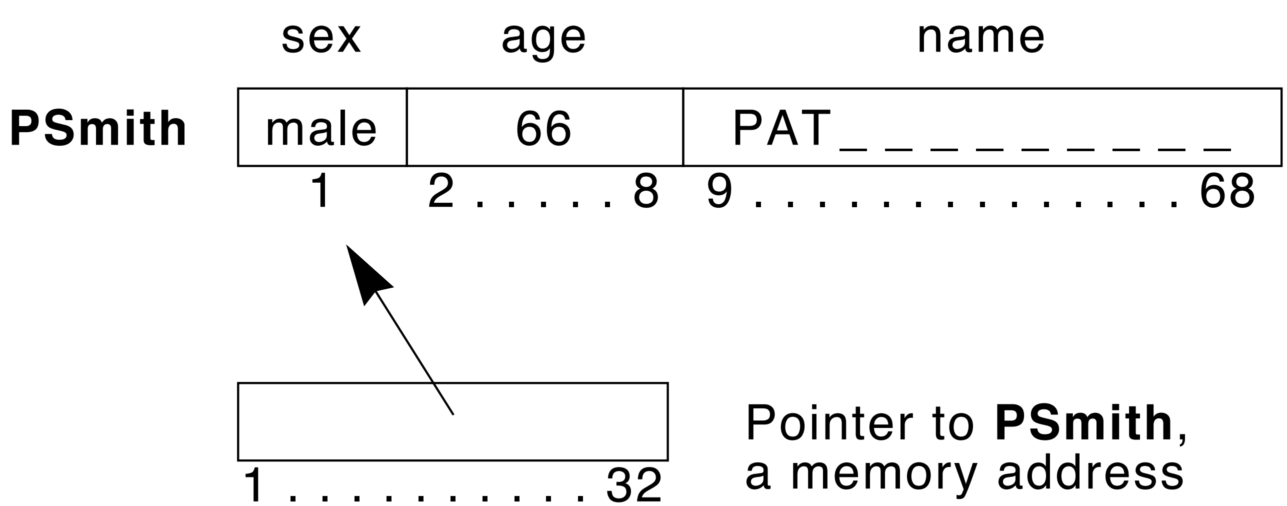
- Semantic pointer architecture and firing patterns
- Neural Engineering Framework
- Implemented as Integrate and Fire neurons

# SPAUN: main approaches used

- Sematic pointer architecture and firing patterns
- Neural Engineering Framework
- Implemented as Integrate and Fire neurons

# Encode: (including compression)

$$\mathbf{PSmith} = [\mathbf{name} \otimes \mathbf{Pat} + \mathbf{sex} \otimes \mathbf{male} + \mathbf{age} \otimes \mathbf{66}]$$

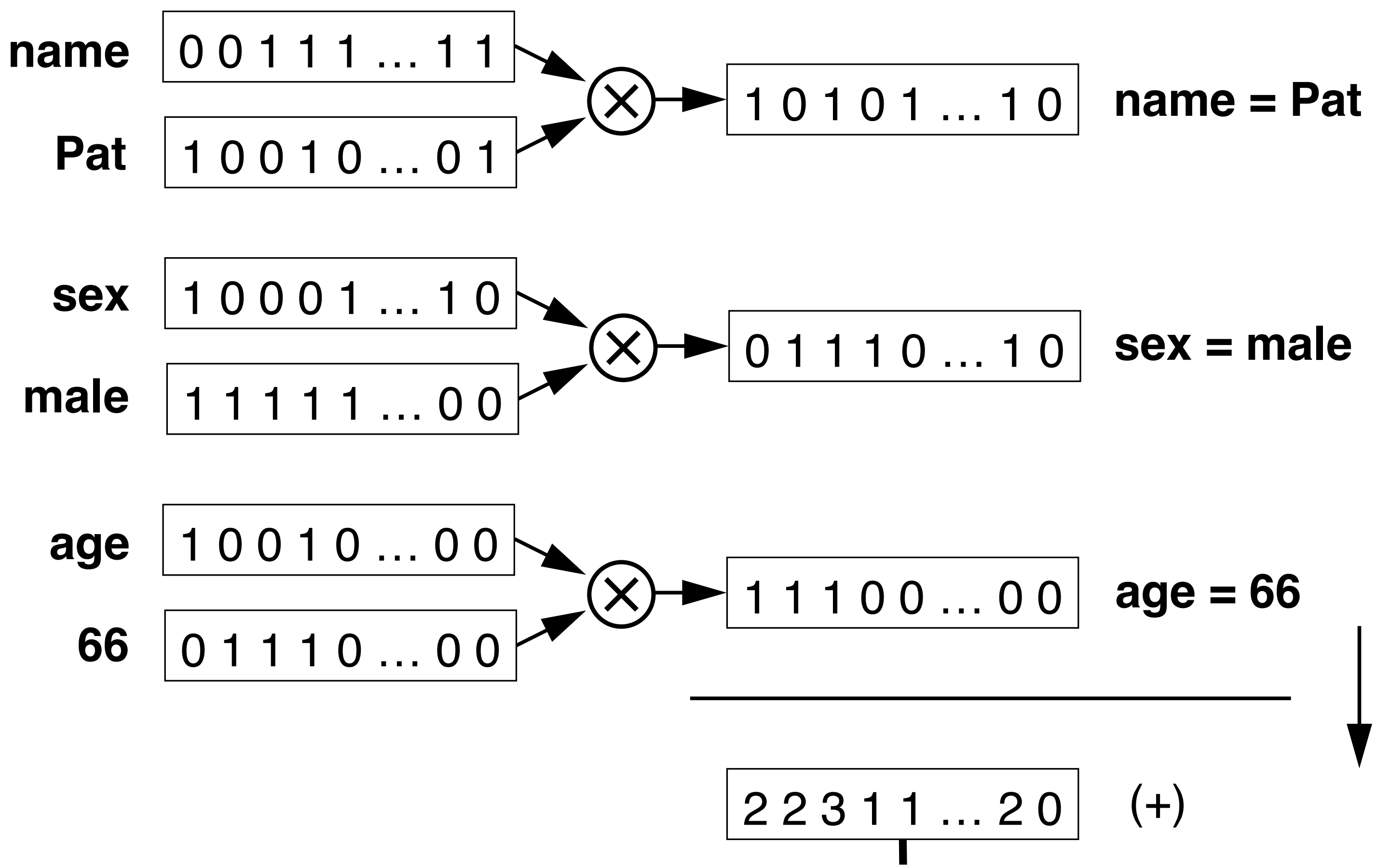



(But carries semantic information that is similar but compressed version of original)

Original framework: Plate 1991, 1993; figures from Kanerva 1997

# Encode: (including compression)

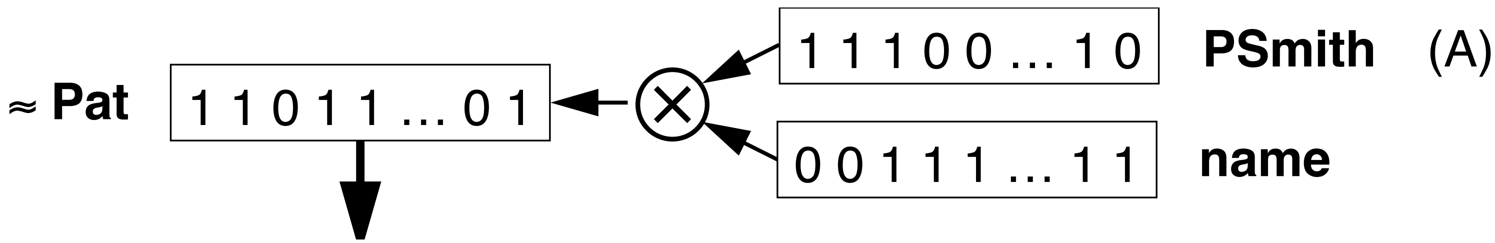
**PSmith = [name⊗Pat + sex⊗male + age⊗66]**




Original framework: Plate 1991, 1993; figures from Kanerva 1997
(here XOR example; can also use circular convolution to encode)

# Decode:

$$\mathbf{PSmith = [name \otimes Pat + sex \otimes male + age \otimes 66]}$$

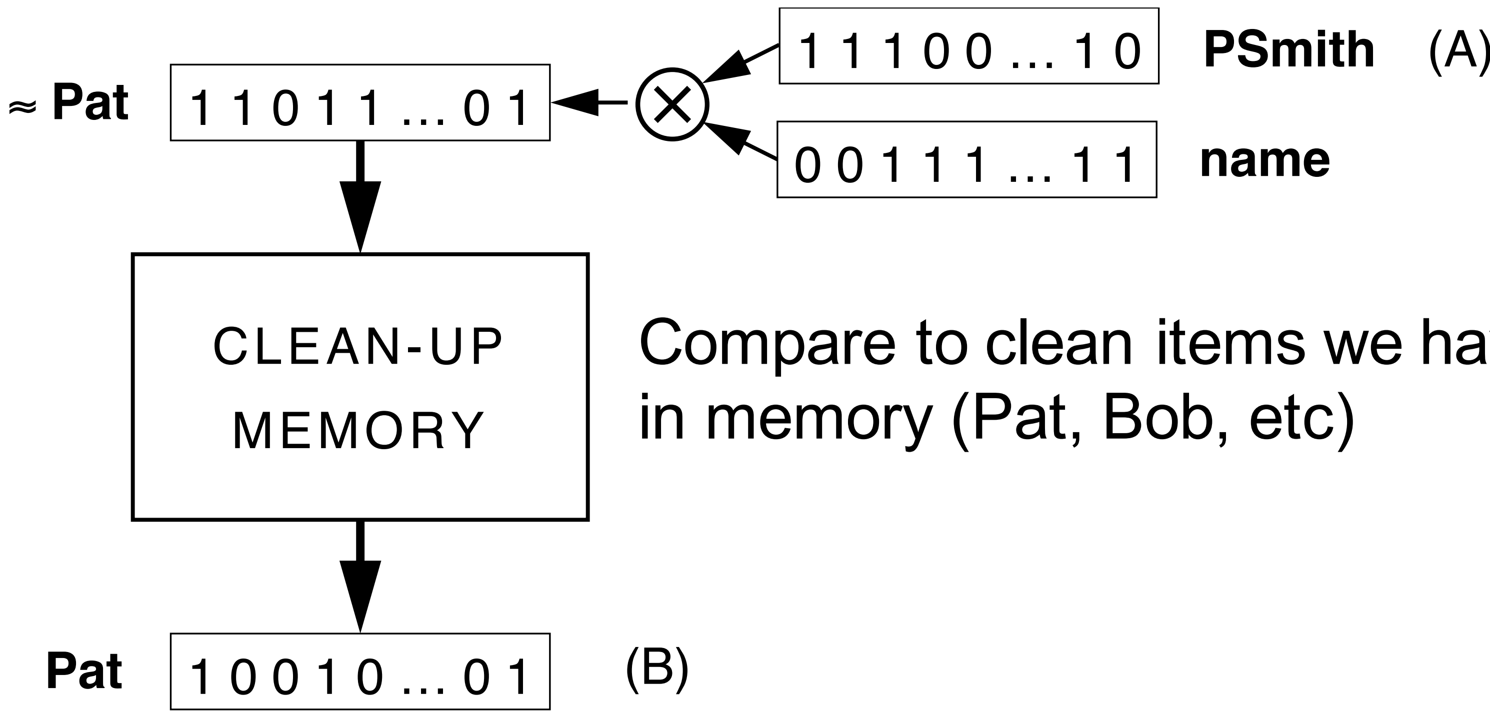



In decoding, we reverse the operation. Due to compression we retrieve noisy version of original

Original framework: Plate 1991, 1993; figures from Kanerva 1997
(here use XOR; can also use circular convolution inverse)

# Decode:

$$\mathbf{PSmith = [name \otimes Pat + sex \otimes male + age \otimes 66]}$$




Compare to clean items we have in memory (Pat, Bob, etc)

Original framework: Plate 1991, 1993; figures from Kanerva 1997

# In Eliasmith paper:

$$MemoryTrace = Position1 \otimes Item1 + Position2 \otimes Item2 + ...$$

The items are numbers (digits) in SPAUN
Positions are for instance positions in list

See supplementary material (can be learned within a spiking neural network)

## SPAUN: main approaches used

- Sematic pointer architecture and firing patterns
- Neural Engineering Framework
- Implemented as Integrate and Fire neurons

- Neural Engineering Framework

A group of neurons can represent a vector space over time, and connections between neurons can compute functions on those vectors. Provides methods to determine what these connections should be to compute a given function.

Ex: Visual model includes receptive fields that are essentially learned (like V1 filters). Spiking activity can be specified on the neural population

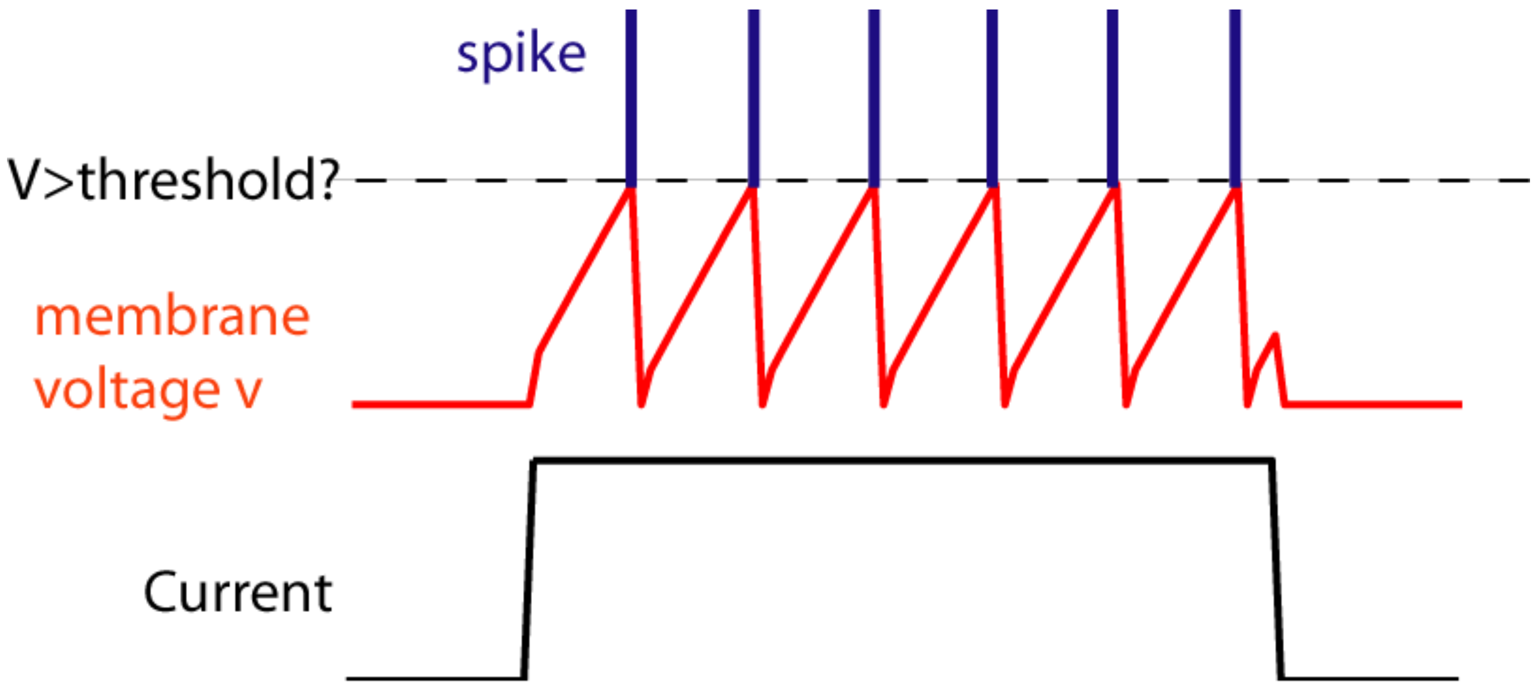
## SPAUN: main approaches used

- Sematic pointer architecture and firing patterns
- Neural Engineering Framework
- Implemented as Integrate and Fire neurons

# Leaky Integrate and Fire Model

- Describes some properties of voltage change over time and spiking activity
- Parameters correspond to known properties of neurons (and electrical circuits)
- Simple (doesn’t model biophysical detail; compare to Hodgkin Huxley)
- Simple (DE can be solved, example, in simple version using separable DE!)
- Simple (widely used in brain modeling, scales up to networks of neurons)

# Leaky Integrate and Fire

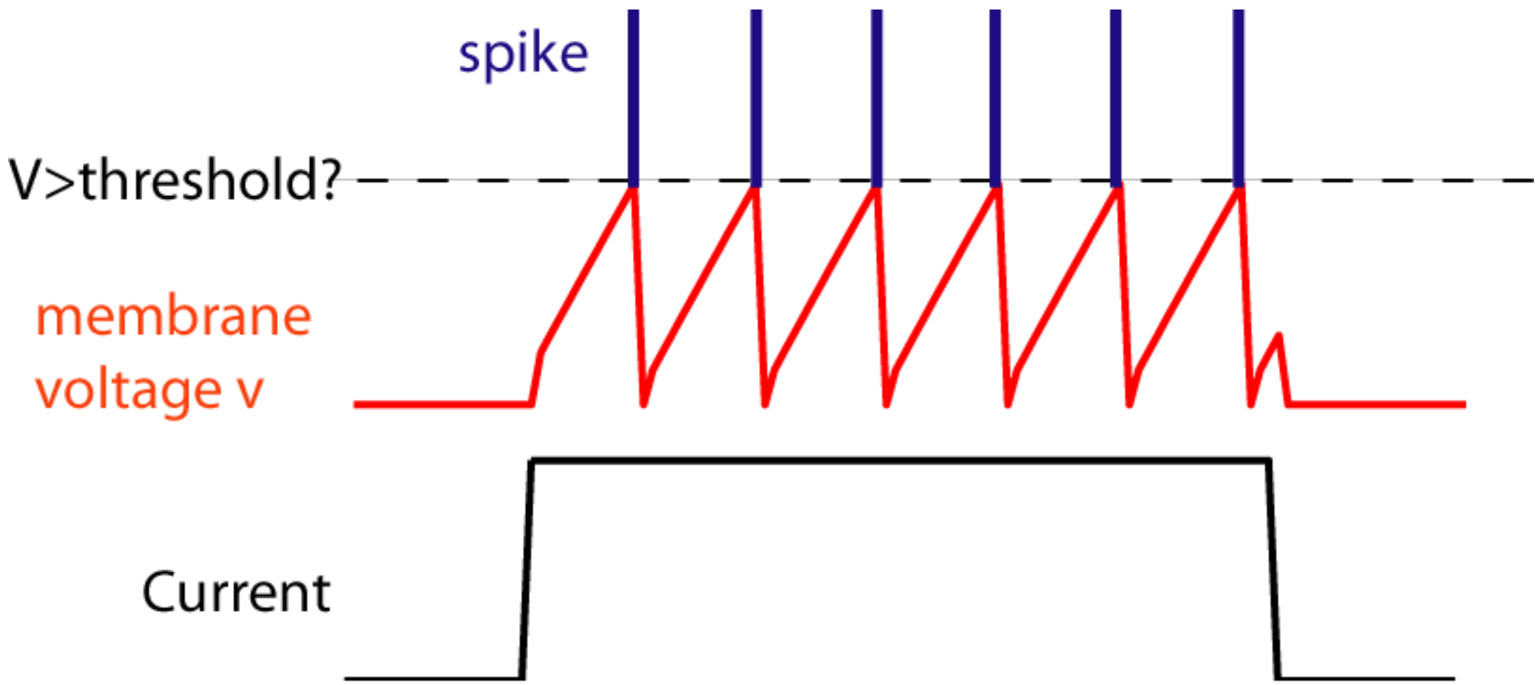



Leak

Current

$$C\frac{dv}{dt} = \frac{-v}{R} + I(t)$$

# Leaky Integrate and Fire




Leak

Assume constant current

$$\frac{dv}{dt} = \underbrace{\frac{-v}{\tau}}_{\text{Leak}} + \frac{I}{C}$$

$$\tau = RC$$

# Leaky Integrate and Fire DE

- DE $$\frac{dv}{dt} = \frac{-v}{\tau} + \frac{I}{C}$$

- Change with time: v(t), t

- Assume constants: I, R, C, $\tau = RC$

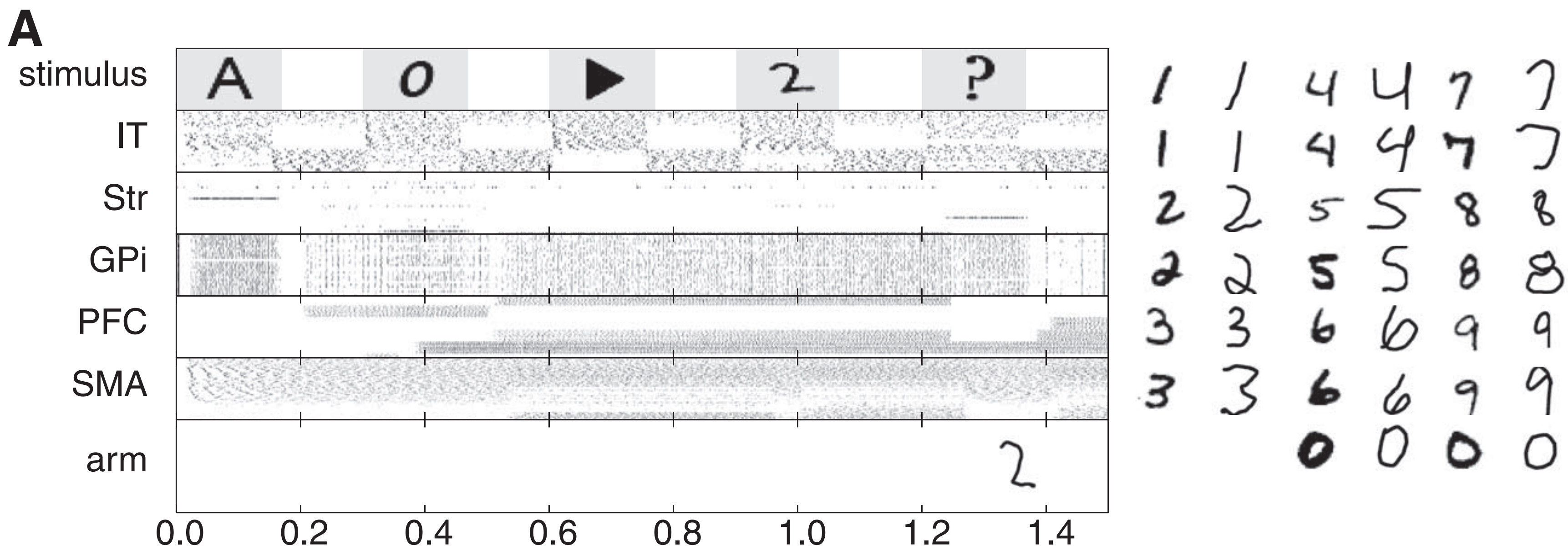
- Solving:

$$v(t) = v(t=0)e^{-t/\tau} + RI(1 - e^{-t/\tau})$$

Back to SPAUN and tasks…

https://youtu.be/WNnMhF7rnYo

Copy drawing task




Copy Drawing Task. Captures in drawing the particular digit style (eg, of the 2) shown https://youtu.be/qcZe-2eWaeM

Raven’s progressive matrix task

Learn: 1 11 111
4 44 444

Show input: 5 55

Output? 555

(also other patterns: 1 2 3; 5 6 7..
learns the “rules”: 3 4 ?)

Raven’s Progressive Memory Task

- Bridge gap between complex behaviors and complex neural activity

- Performs whole set of simulated tasks associated with human cognition

- Large scale: 2.5 million neurons

- Principles of encoding decoding (and compression)

- Uses spiking models of neurons

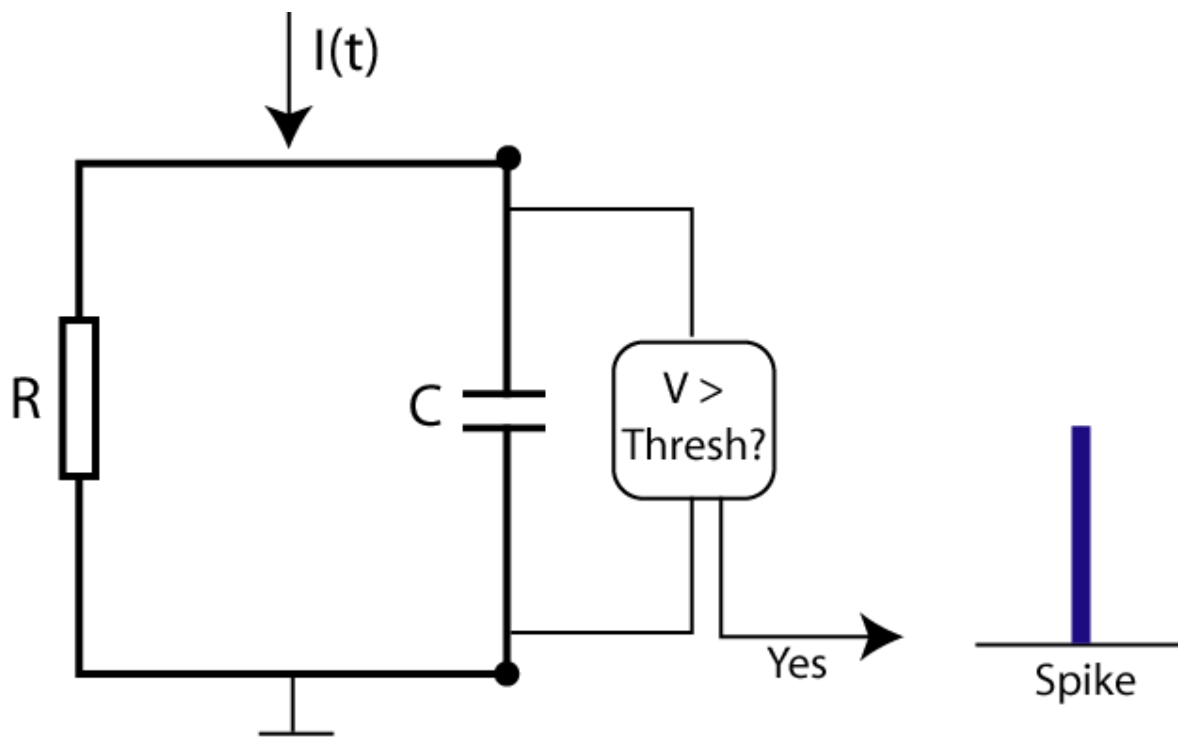
Limitations? …

Limitations? …

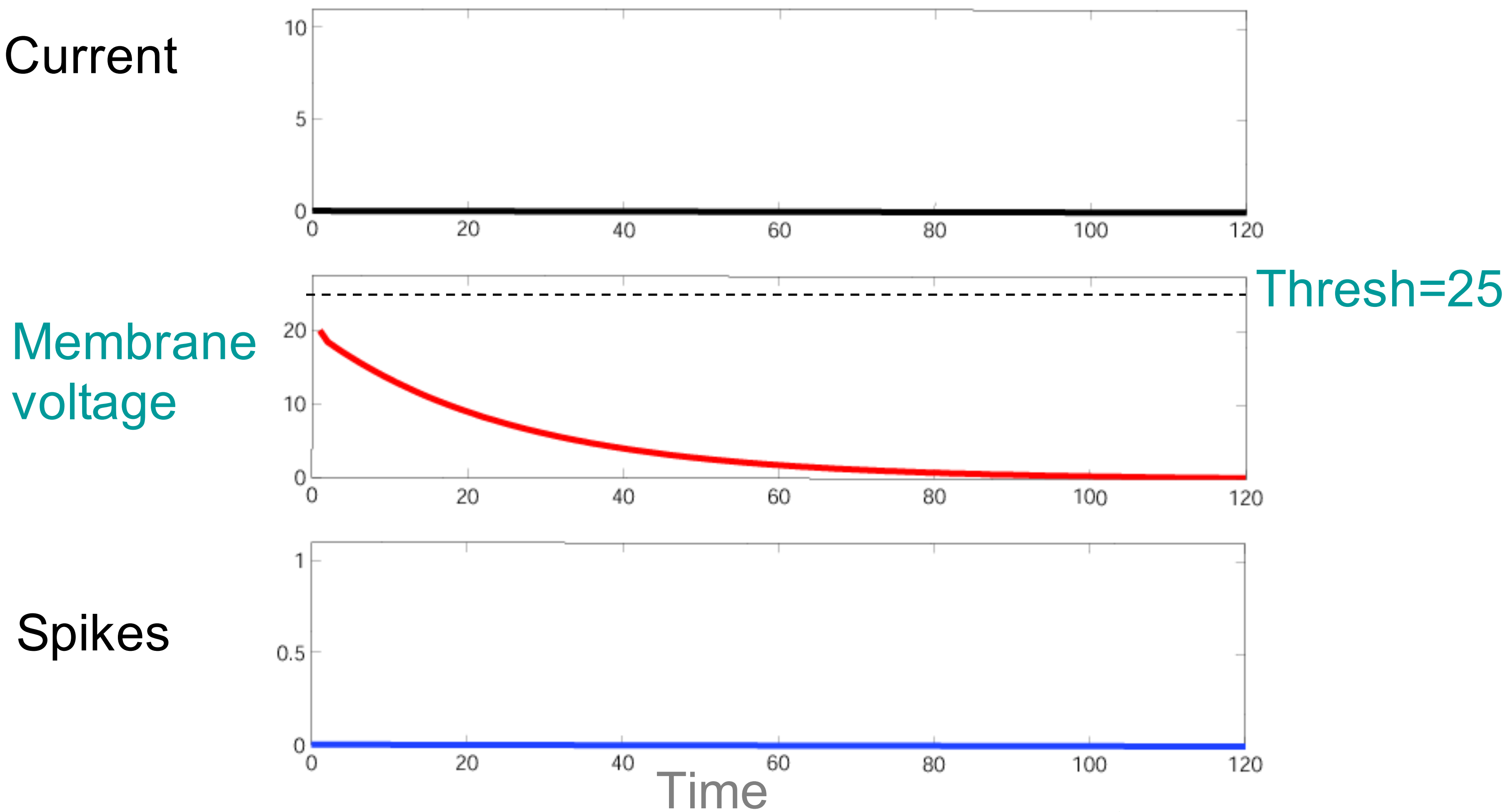
- “Little to say about how that complex, dynamical system develops from birth”
- “Not as adaptive as a real brain … ”
- “attention, eye position fixed”
- “limited to space of digits from 0 to 9”
- “missing areas of the brain…”

“Even in its current form, Spaun offers a distinctly functional view and set of hypotheses regarding the neural mechanisms and organization that may underlie basic cognitive functions.”

# Leaky Integrate and Fire Circuit



NO CURRENT I




$$v(t) = v(t=0)e^{-t/\tau} + \cancel{RI(1-e^{-t/\tau})}$$

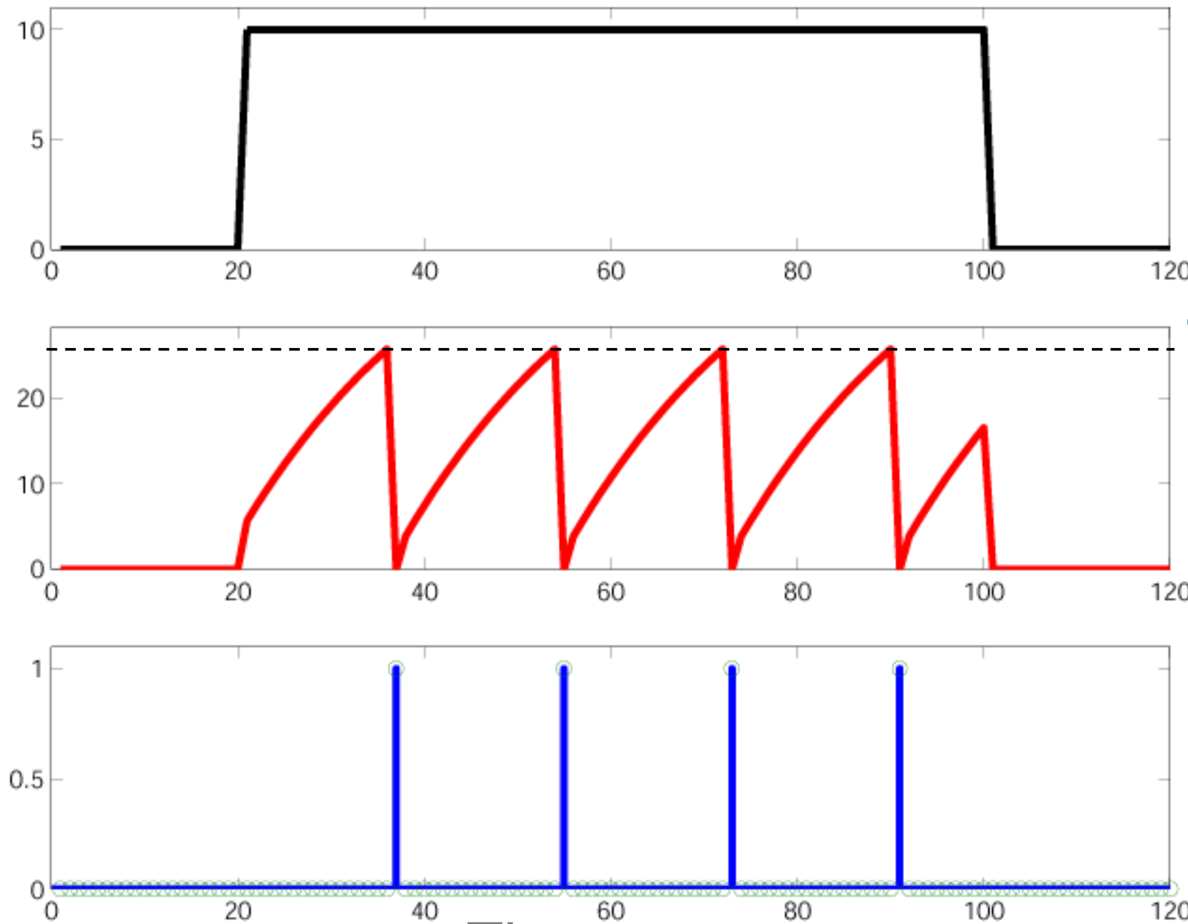
## WITH CURRENT I and V(t=0)=0

Current

Membrane voltage

Spikes

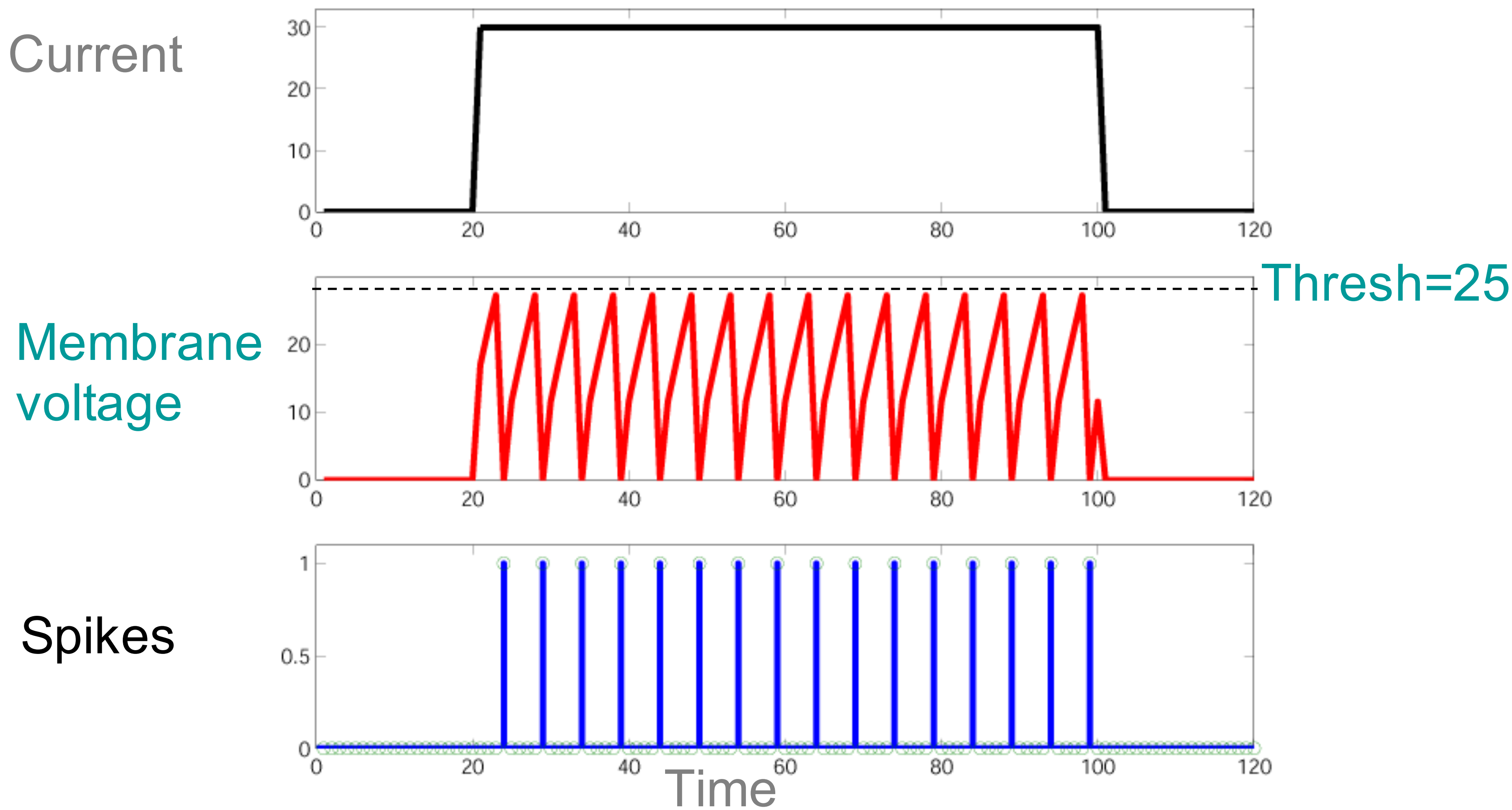
Thresh=25




Time

$$v(t) = v(\cancel{t = 0})e^{-t/\tau} + RI(1 - e^{-t/\tau})$$

# INCREASE CURRENT I




$$v(t) = v(\cancel{t = 0})e^{-t/\tau} + RI(1 - e^{-t/\tau})$$

# Membrane voltage and spiking