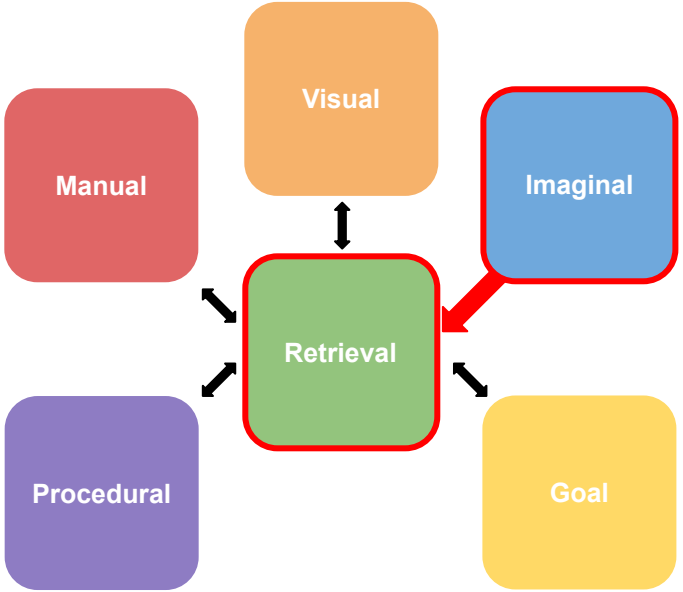
Context is Everything: possibilities for a larger role of spreading activation in ACT-R

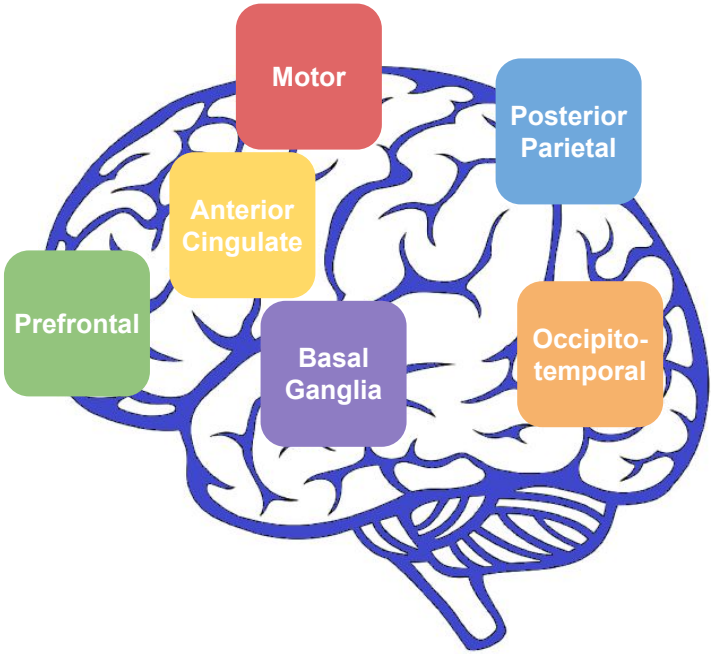
Patrick Rice, Ph.D.
Computer-Human Interaction Lab
Rice University

Who here uses spreading activation in their ACT-R modeling?

- If you've ever called:
  - `(sgp :esc t :mas [>0])`
- You have!


Visual
Manual
Imaginal
Retrieval
Procedural
Goal

S_i

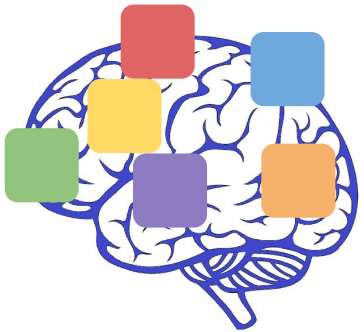

Motor
Posterior Parietal
Anterior Cingulate
Prefrontal
Basal Ganglia
Occipito-temporal

But what does spreading activation do for us?

- Convenient answer: makes it more likely to retrieve some chunk
- ACT-R theory answer: provides a particular form of context to declarative memory retrieval operations

What does “context” mean in a cognitive setting?

- Population encoded representations (Nadeau 2020)
- A state or set of states closely related to the current cognitive state
  - Recent: connection weights integrate experience over time
  - Similar/analogous: auto-associator networks
- The information contained in these states influences ongoing processing
- ACT-R’s concept of spreading activation is one manner by which particular information informs ongoing processing

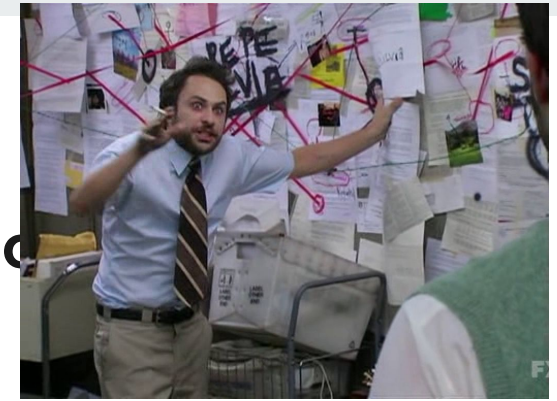


Another way of looking at things...

- Technical(ly correct) answer:
  - Setting `:esc t` enables subsymbolic computation; retrieval of chunks is controlled by activation

$$A_i = B_i + S_i + P_i + \varepsilon_i$$

But what does spreading activation do



- Technical answer:
  - Setting `:esc t` enables subsymbolic computation; retrieval of chunks is controlled by activation
  - Setting `:mas` to a nonzero positive value

$$A_i = B_i + S_i + P_i + \varepsilon_i$$

$$S_i = \sum_k \sum_j W_{kj} S_{ji}$$

But what does spreading activation do for us?

- Technical answer:
  - Setting `:esc t` enables subsymbolic computation; retrieval of chunks is controlled by activation
  - Setting `:mas` to a nonzero positive value

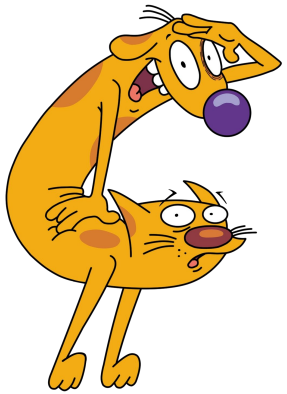
$$S_i = \sum_k \sum_j W_{kj} S_{ji}$$

`:mas`

$$S_{ji} = S - \ln(fan_{ji})$$

Thought expt: Mindless memory model

```
(sgp     :esc t
         :mas [1.0,2.0,3.0]
         :ans 0.2
)
```



:mas

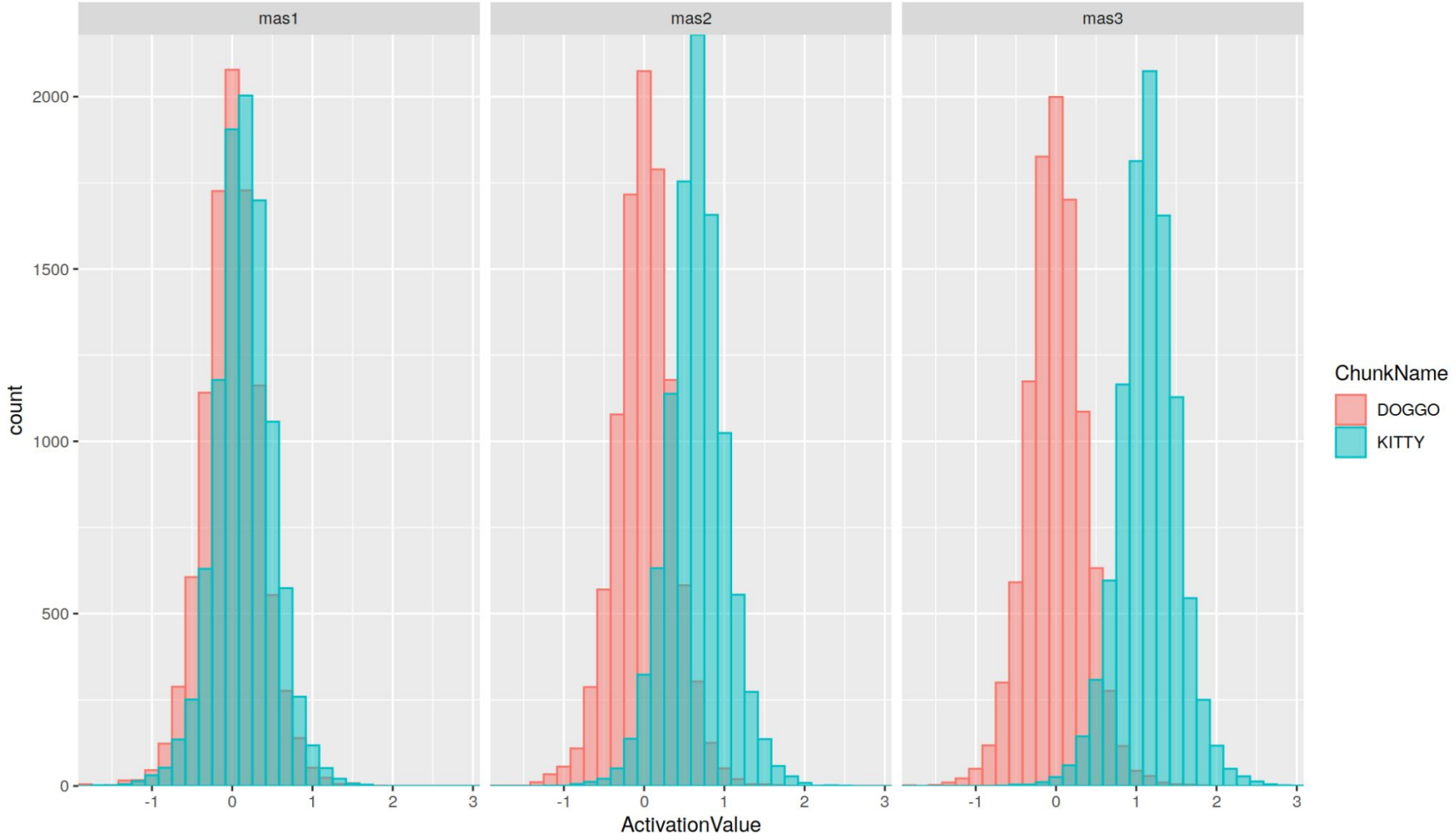
$$S_{ji} = S - \ln(fan_{ji})$$

Mindless memory model

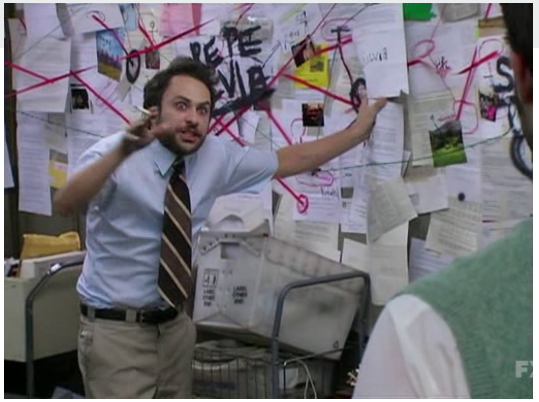
```
(P parse-prompt
        =goal>
                ISA parse-prompt
                state parse
        =visual>
==>
        =visual>
        @imaginal>
                =visual
        =goal>
                state retrieve
)
```

Mindless memory model

```
(P retrieve-fact
        =goal>
                ISA parse-prompt
                state retrieve
        ?retrieval>
                state free
==>
        +retrieval>
)
```


mas1
mas2
mas3
count
2000
1500
1000
500
0
-1
0
1
2
3
ActivationValue
ChunkName
DOGGO
KITTY

Mindless memory model



- Why did increasing `:mas` increase the discriminability of `kitty` over `doggo`?
- The visual prompt, stored in the imaginal buffer, specified `diet carnivore`
- The model only knows that `kitty` chunks have a `diet` slot value of `carnivore`

$$S_i = \sum_k \sum_j W_{kj} S_{ji}$$

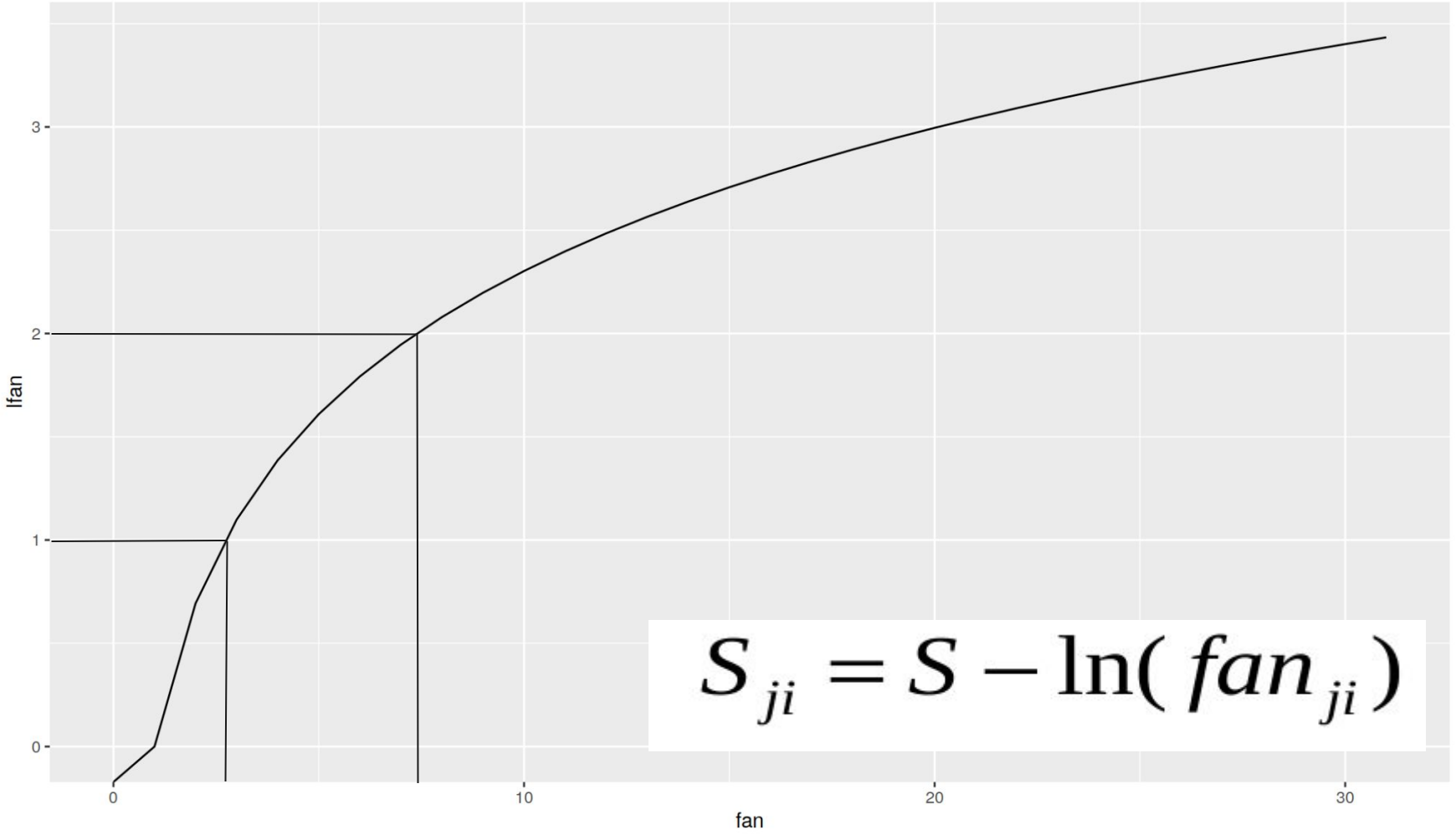
`:mas`

$$S_{ji} = S - \ln(fan_{ji})$$

Fan_{ji}tastic

$$S_{ji} = S - \ln(fan_{ji})$$

$$fan_{ji} = \frac{1 + slots_j}{slotsof_{ji}}$$



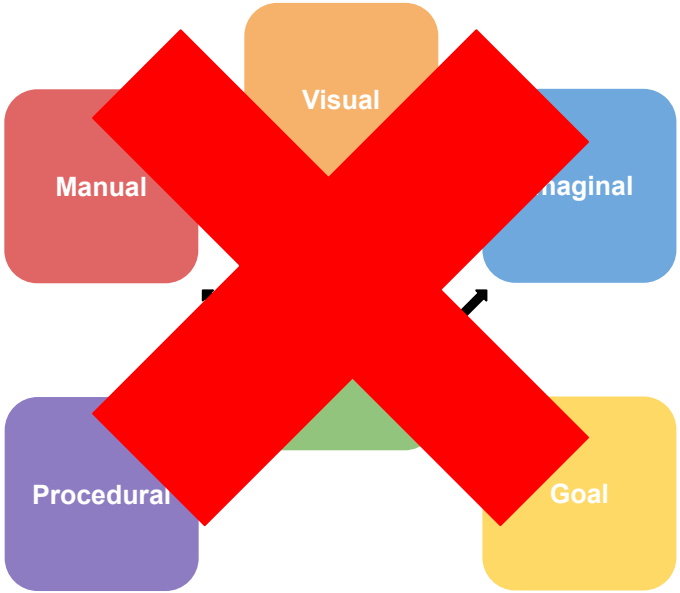
$$S_{ji} = S - \ln(fan_{ji})$$

But what about W_{kj}?

$$S_i = \sum_k \sum_j W_{kj} S_{ji}$$

- By default, is defined as the value of the buffer k's activation spread parameter divided by the number of potential sources j represented by the chunk in buffer k

But what about W_{kj}?



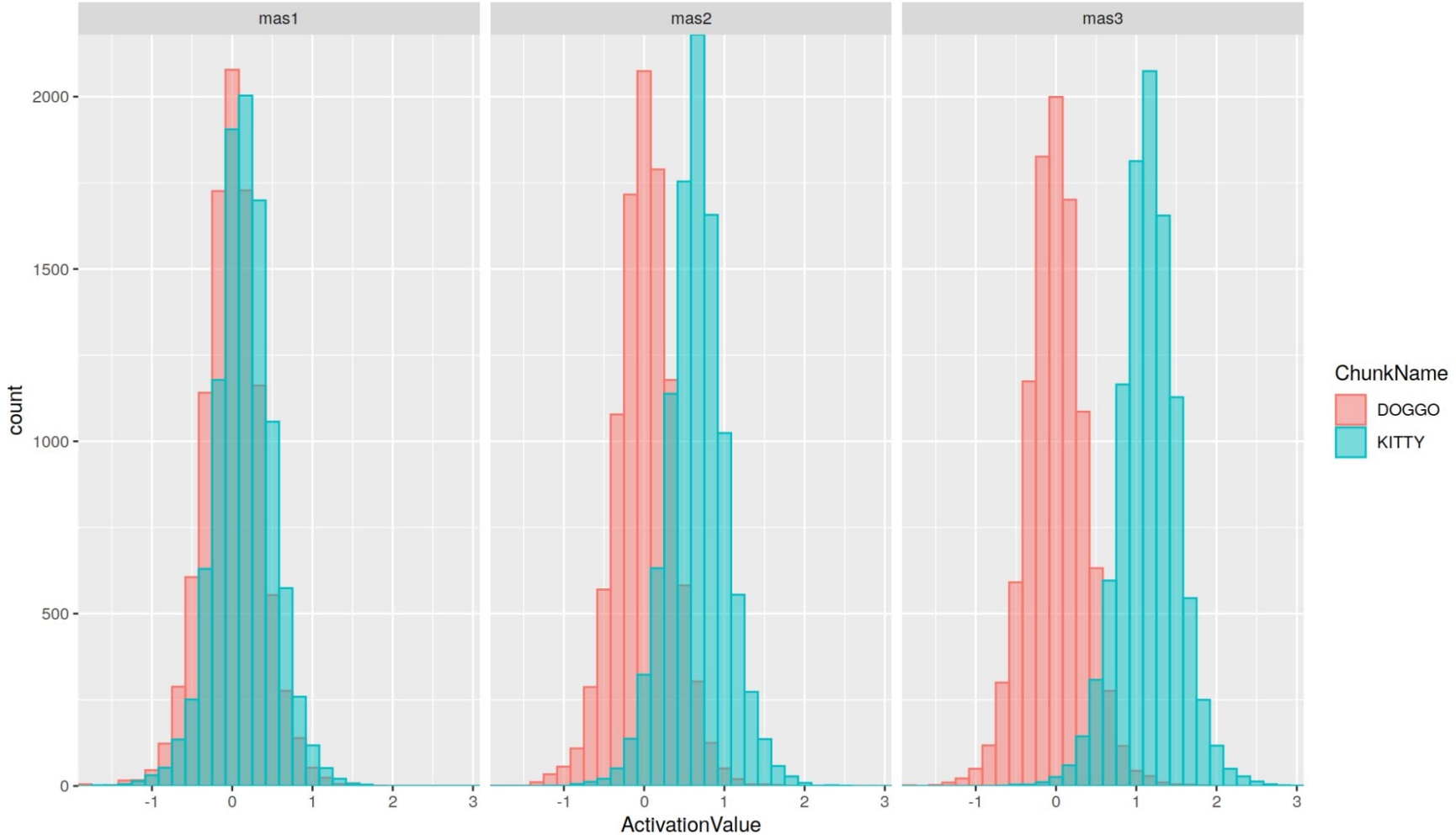
$$S_i = \sum_k \sum_j W_{kj} S_{ji}$$

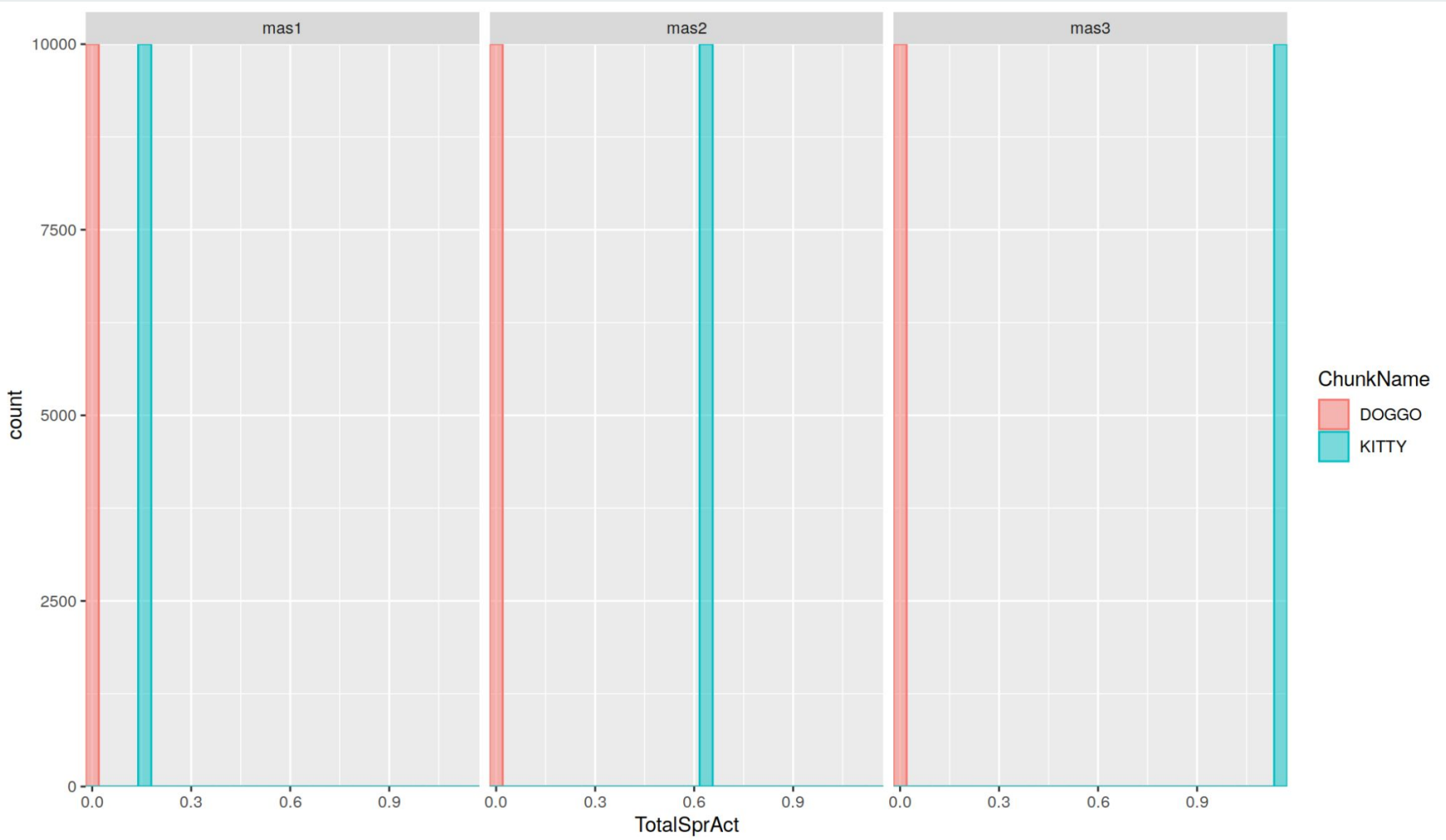
- By default, only the imaginal buffer's activation spread parameter is greater than zero
- Activation spread parameters exist for every other buffer, but by default, all are zero - negating that buffer's contribution to spreading activation

Do you know what YOUR buffers are spreading?

- Production buffer
  - :production-activation
- Goal buffer
  - :ga
- Retrieval buffer
  - :retrieval-activation
- Visual/visual-location buffers
  - :visual-activation/:visual-location-activation
- Aural/aural-location buffers
  - :aural-activation/:aural-location-activation
- Manual buffer
  - :manual-activation
- Vocal buffer
  - :vocal-activation
- Temporal buffer
  - :temporal-activation

Source of variability in spreading activation


mas1
mas2
mas3
count
2000
1500
1000
500
0
-1
0
1
2
3
ActivationValue
ChunkName
DOGGO
KITTY


mas1
mas2
mas3
10000
7500
5000
2500
0
count
0.0
0.3
0.6
0.9
TotalSprAct
ChunkName
DOGGO
KITTY

Source of variability in spreading activation

- Is it intuitive that there's no variability in spreading activation?
- In attractor networks, the amount of activation that spreads is dependent on the strength of association between the representational units (Lerner, Bentin, & Shriki 2012)
    - Similar to ACT-R - strength of association is representative of the relatedness between two concepts
    - Noise in spreading activation is driven by noise in unit activation
    - How well does ACT-R's S_i capture this?

Questions

- Could the implementation of ACT-R's spreading activation be expanded?
  - By default, only one buffer contributes to the mechanism
  - Only one operation is modulated by the mechanism
  - Do we actually want "spreading activation"?
- How can :mas be interpreted in the PER framework?
  - Should :mas be informed by some global fan value?
  - Could source activation be driven by the quality of the module's representation?
- Is the lack of spreading activation variability reasonable?
  - Should distinct, functionally specialized mechanisms have their own noise parameters?
  - Potentially enabled by representation-driven source activation

Thank you!

Mindless memory model

```
(chunk-type mammal type domesticated diet body social communication)

(add-dm (retrieve-facts isa parse-prompt state find type nil)

        (kitty isa mammal type feline domesticated late diet carnivore body
        small social less communication scent)

        (doggo isa mammal type canine domesticated early diet omnivore body
        large social more communication bork)

)
```

Mindless memory model

```
(add-visicon-features '(screen-x 10 screen-y 10 diet carnivore))
```

- Have the model make 10,000 retrievals under these conditions
- Examine effect on chunk activation across these retrievals