From Verbal Theories to Computational Models

Klaus Oberauer

University of Zurich

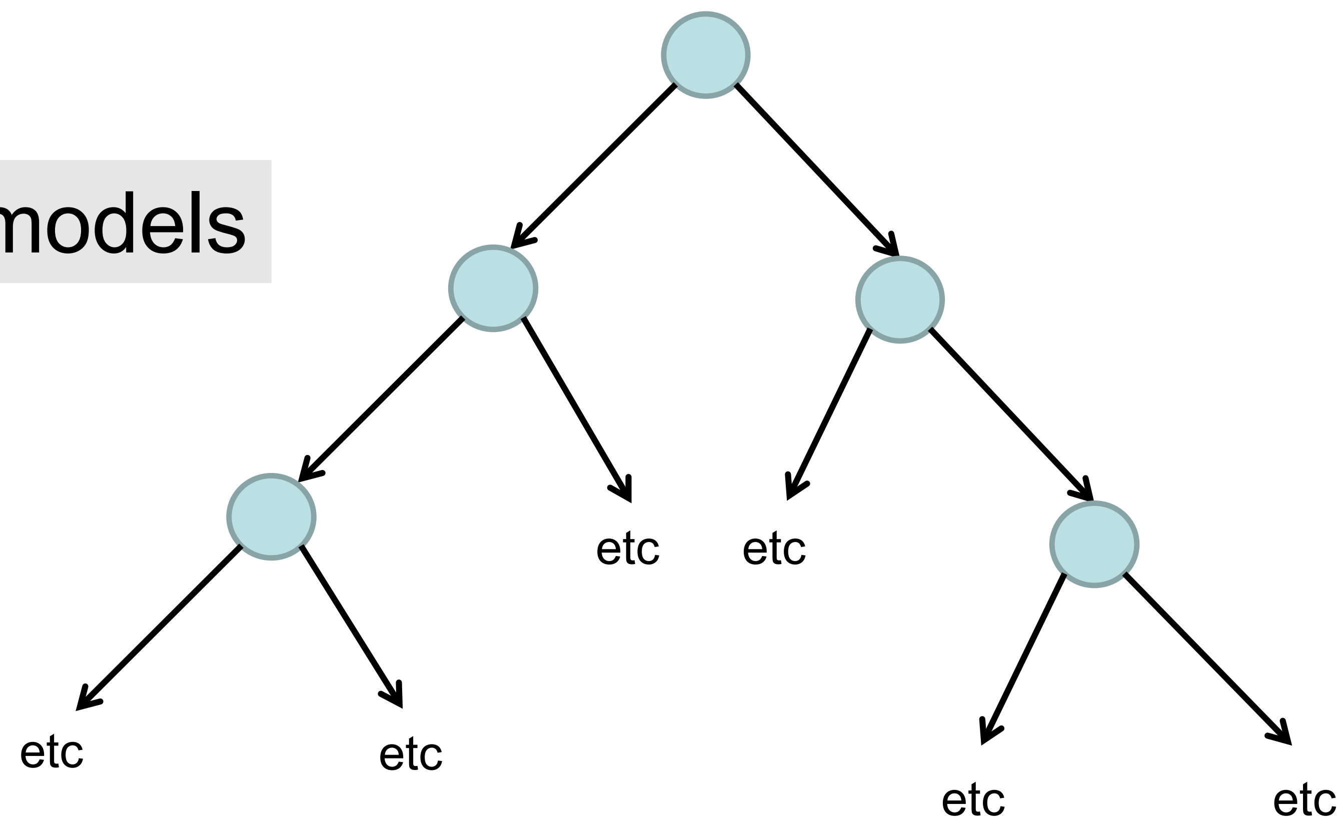
Verbal Theories and Computational Models

- Theories and Models = sets of data patterns they are compatible with
  - Possible worlds semantics

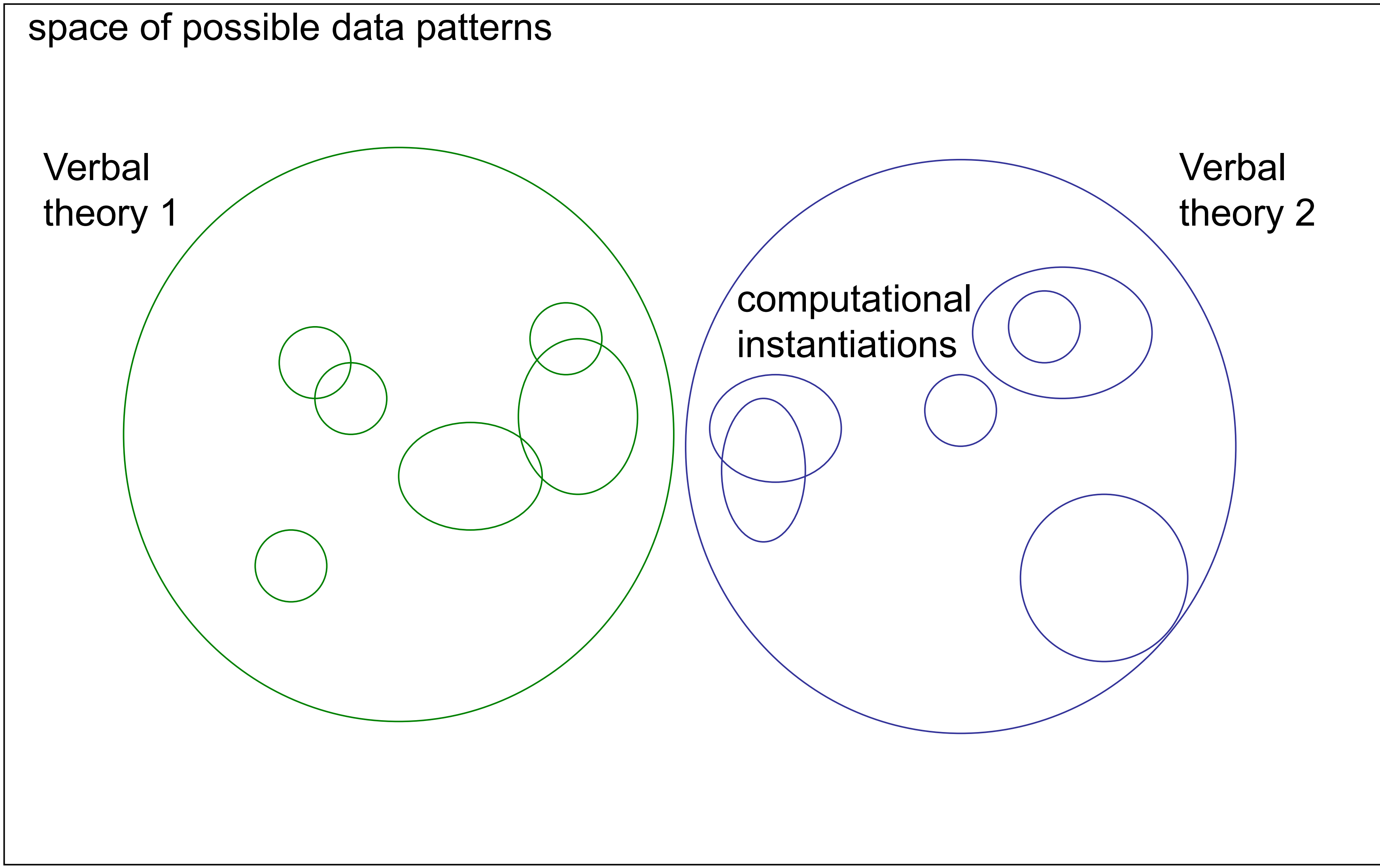
From Verbal Theories to Computational Models

- Need to specify many details
- *n* decision points → many computational models

2^n models



Verbal Theories and Models

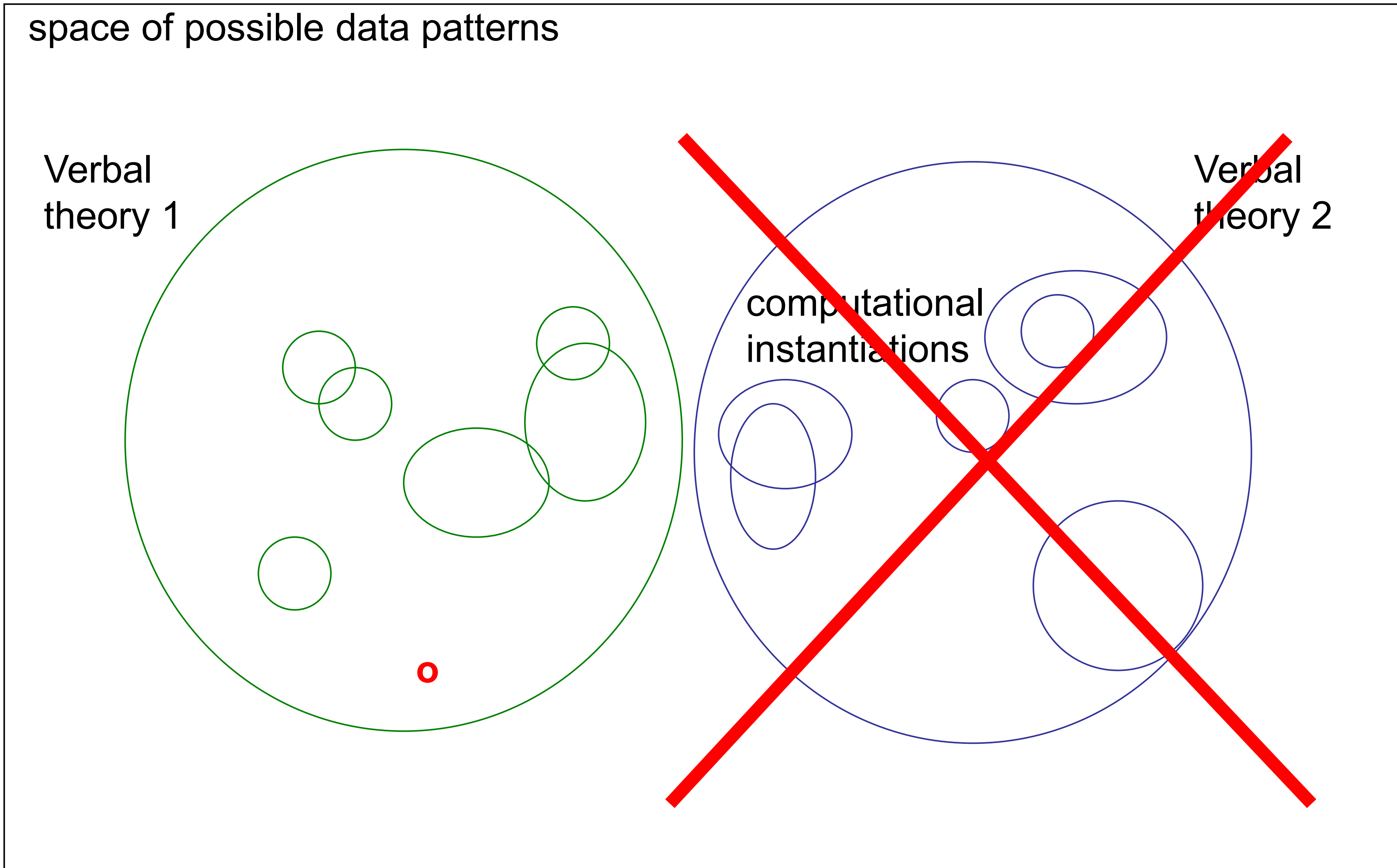


Verbal Theories and Models

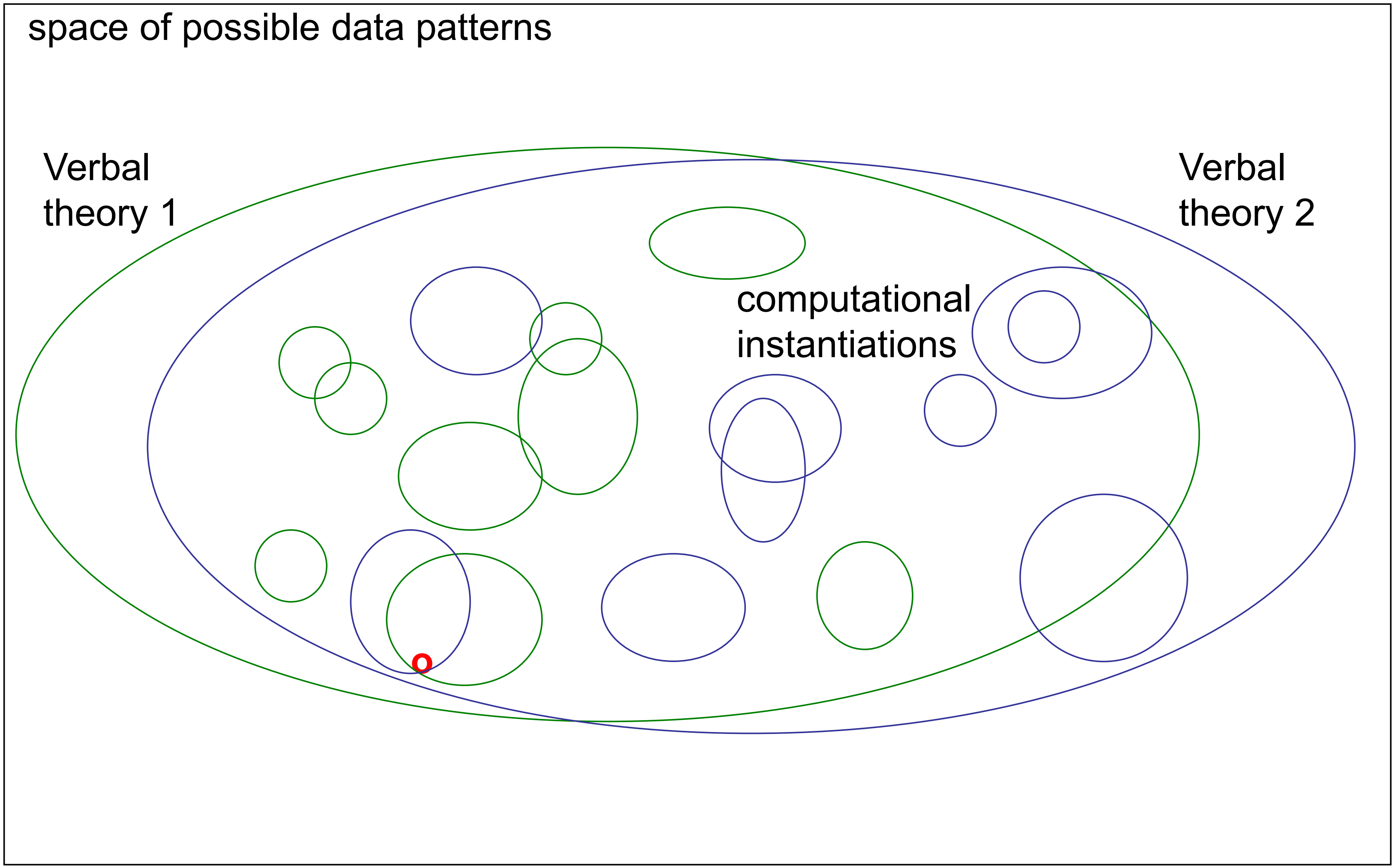
- A verbal theory is worth ~2^n models
- Negative: We don‘t say anything about the details
- Positive: We can say something general without knowing about the details
  - well, can we?

- 2 cases:
  - Verbal theory allows general prediction
  - Verbal theory predicts more or less anything, depending on implementation

The optimistic scenario



The pessimistic scenario



Why the pessimistic scenario is plausible

- Theories have to assume multiple mechanisms working together
  - e.g, decay, rehearsal, attention, interference, multiple buffers, multiple control processes...
- They interact in often unforseeable ways
- Can‘t test for individual mechanisms out of context
  - Allen Newell (1973): „You can‘t play 20 questions with nature and win“

The benefit of computational implementation

- We usually don‘t know in advance what a verbal theory predicts
  - intuition often fails!
- Implementation clarifies that
- Possible outcomes:
  - there is no consistent implementation (rare)
  - there are many consistent implementations that predict different things → testability issue
  - there is a limited set of consistent implementations, they all predict (in some regard) the same

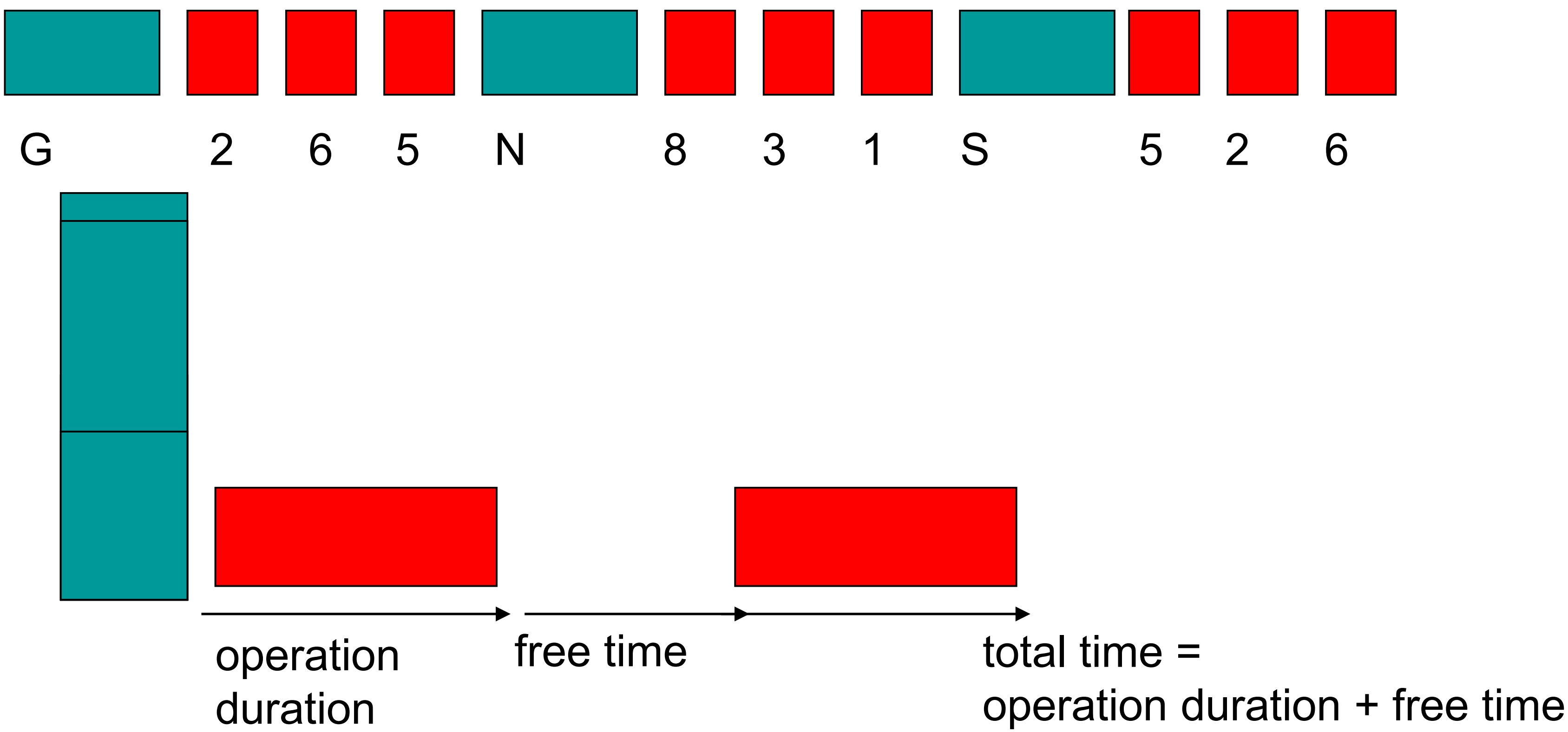
An example: Implementing the TBRS
(Oberauer & Lewandowsky, 2011, PB&R)

- „Time-Based Resource-Sharing“ model
  - Barrouillet, Camos, and colleagues

- Basic assumptions:
  - Traces in WM decay over time
  - They can be refreshed by general attention
  - general attention acts as a bottleneck, can do only one thing at a time

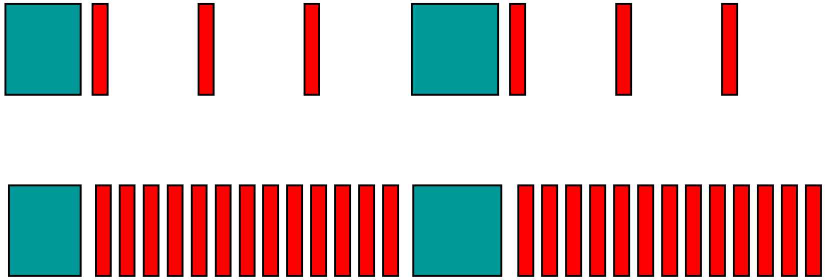
The complex span paradigm

- Alternate between encoding of memory items and steps on processing task



„Cognitive Load“ in the TBRS

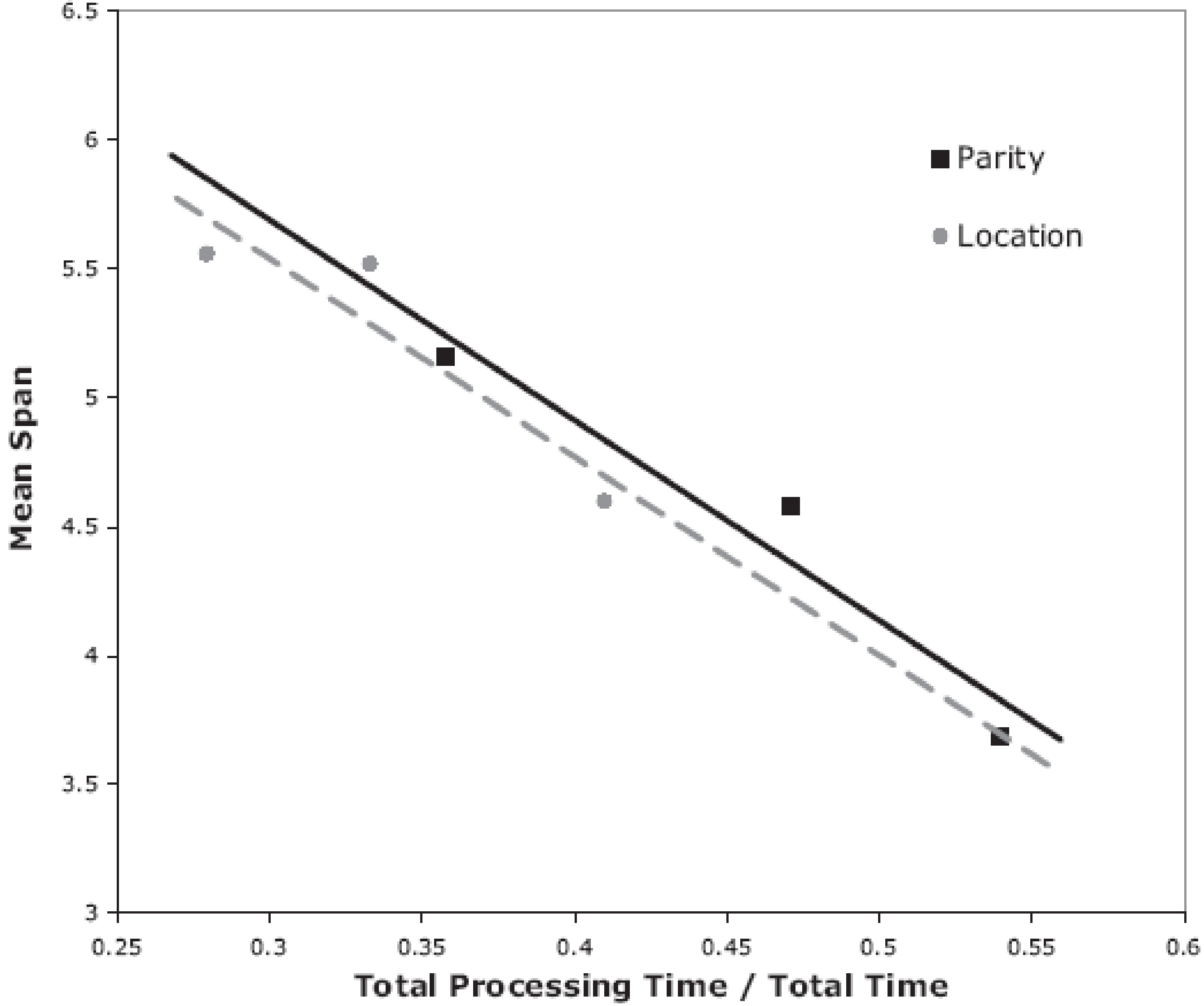
- CL = operation duration / total time
- Span declines (linearly) with CL



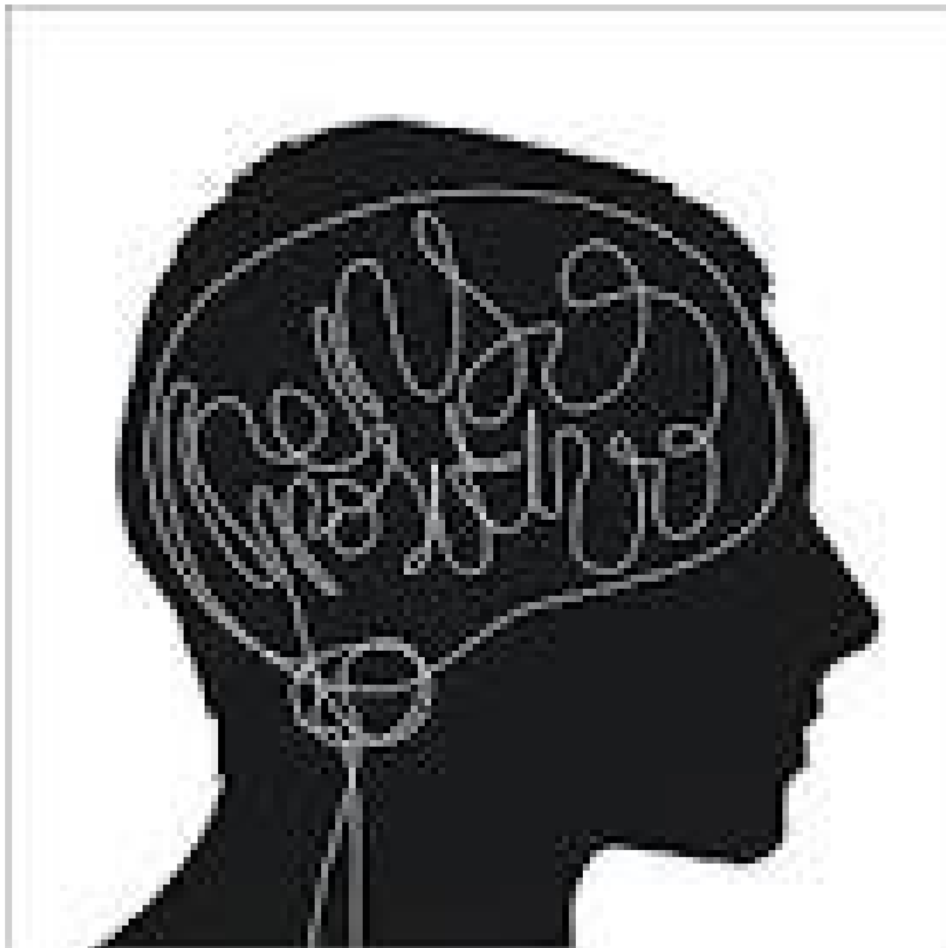
Empirical Support for TBRS

- Complex span more difficult than simple span
- Memory declines as pace of processing increases
- Memory declines as processing steps capture attention for longer
- No effect of number of operations when CL is constant (?)

Barrouillet et al., 2007, Exp. 3



How would you implement TBRS as a model?



Decisions to be made

- Modelling framework
  - symbolic (e.g., ACT-R)
    - concepts & features, propositions, productions
  - connectionist
    - units, connection weights
  - spatial (e.g., SIMPLE, GCM)
    - memory trace = point in space

Decisions to be made

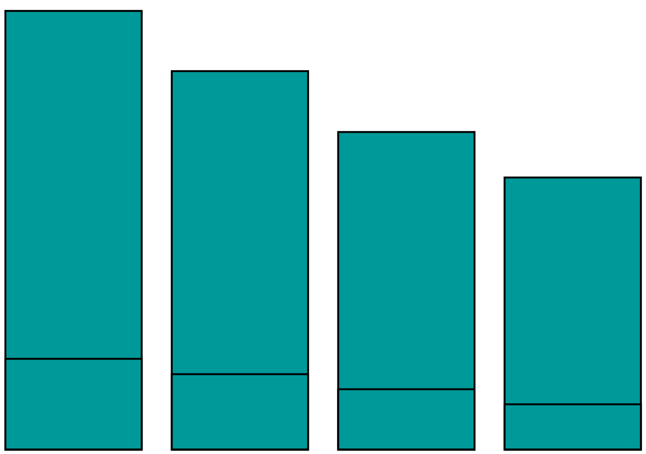
- Modelling framework
  - symbolic, connectionist, spatial
- Representations of items
  - localist
    - each item = single unique unit
  - distributed
    - each item = pattern of activation over all units

Decisions to be made

- Modelling framework
  - symbolic, connectionist, spatial
- Representations of items
  - localist or distributed
- Representation of order
  - primacy gradient or position markers
- tbc...

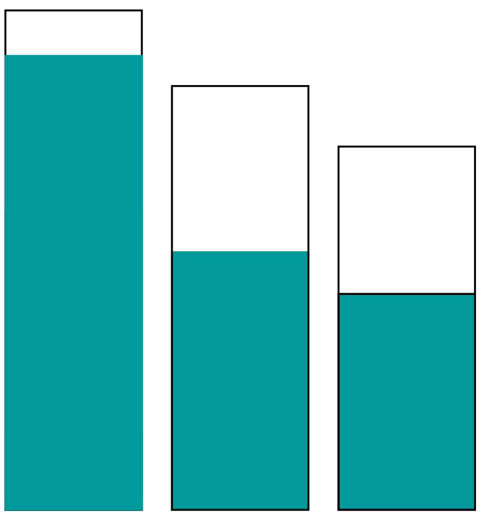
Primacy gradient

- Successive items encoded with decreasing strength
- Recall = pick item with highest strength
- Response suppression → progress



Why a primacy gradient doesn‘t work with rehearsal or refreshing

- Refreshing = retrieval + re-encoding
  - → What to do with response suppression?



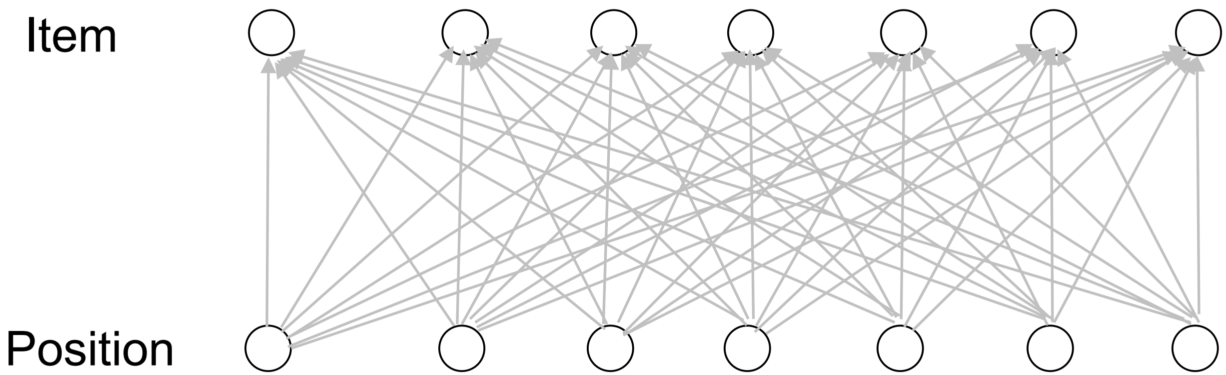
Decisions to be made

- Modelling framework
  - symbolic, connectionist, spatial
- Representations of items
  - localist or distributed
- Representation of order
  - primacy gradient or position markers
- Representation of position markers
  - localist
  - distributed: overlap declining with distance

Decisions to be made

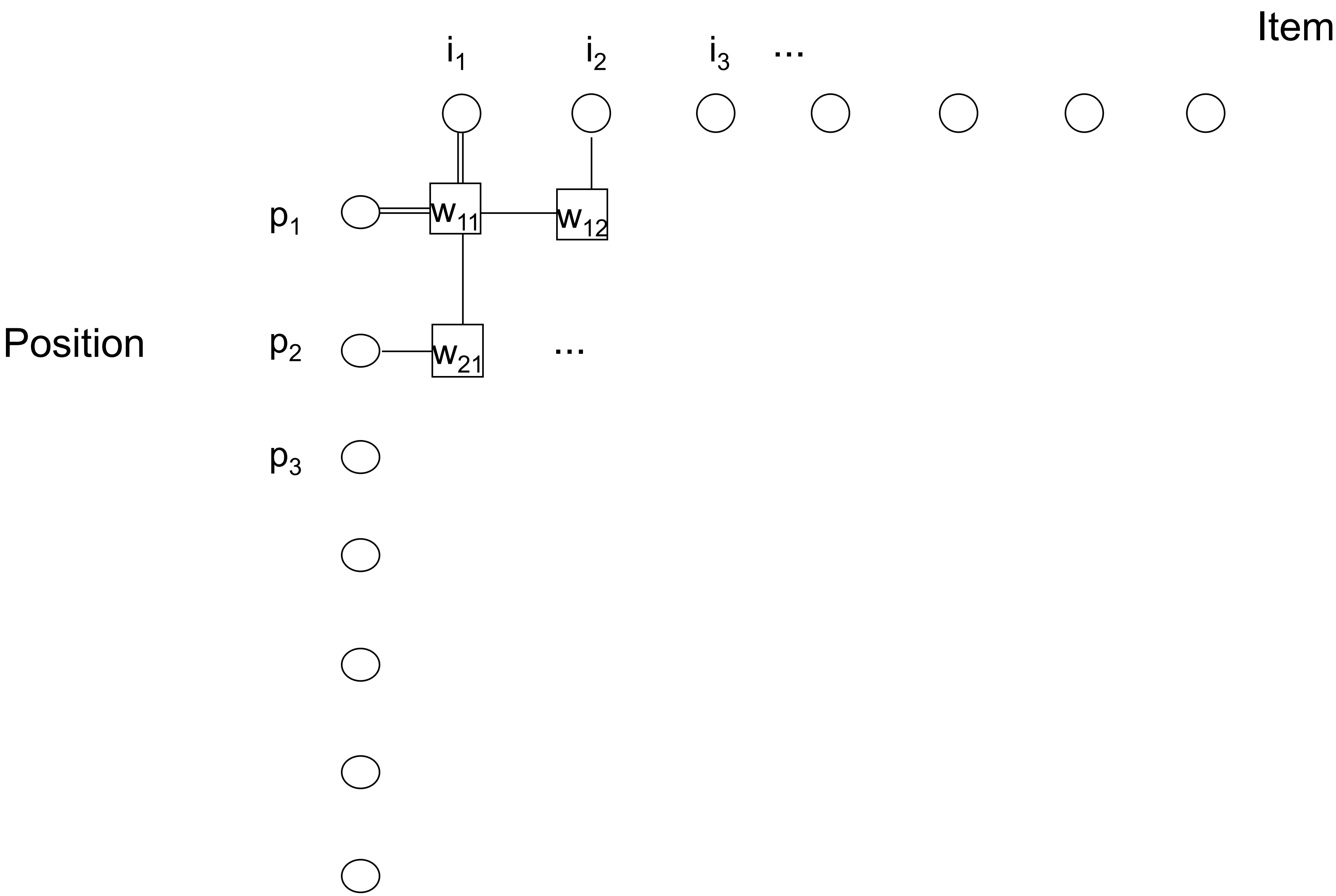
- Modelling framework
  - symbolic, connectionist, spatial
- Representations of items
  - localist or distributed
- Representation of order
  - primacy gradient or position markers
- Representation of position markers
  - localist
  - distributed: overlap declining with distance

TBRS*: Architecture



- Items = localist representations
- Position markers: distributed, overlap declines exponentially with distance

Matrix representation

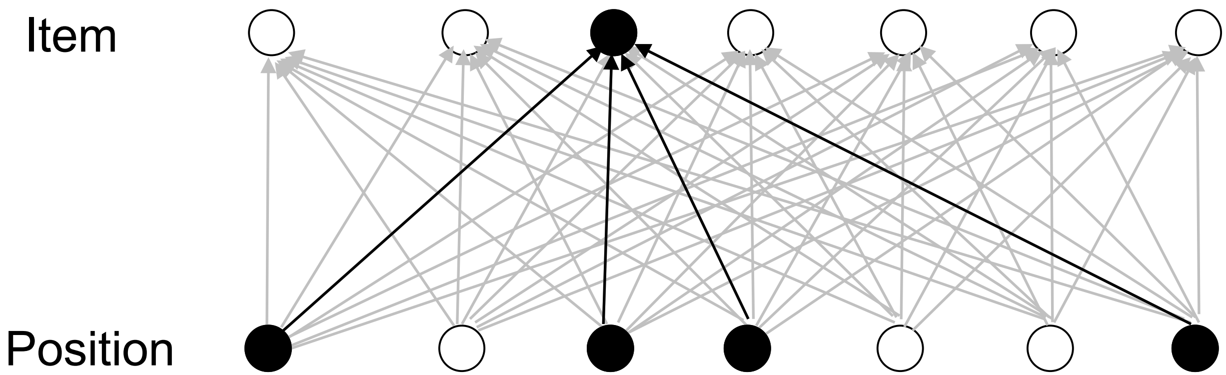


Encoding: Hebbian learning

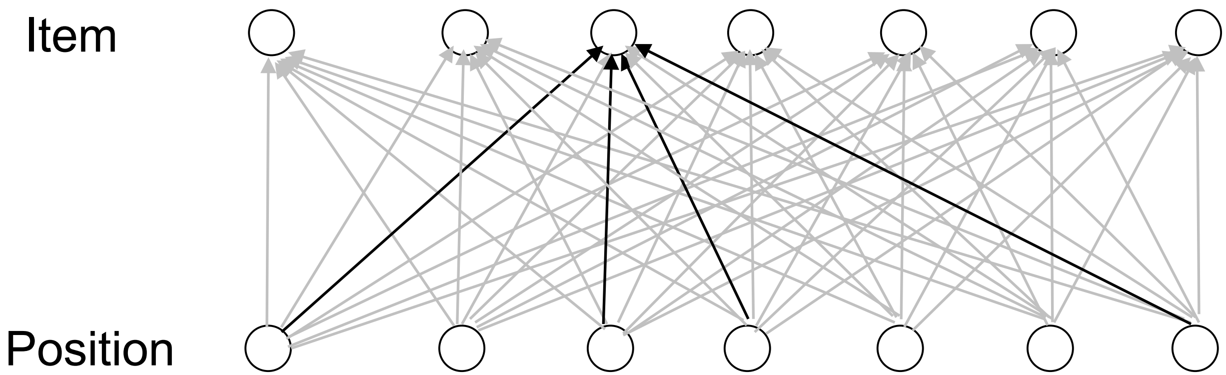
- Association between position markers and items:

$$w_{ij} = w_{ij} + \eta_e a_i a_j$$

TBRS*: Encoding



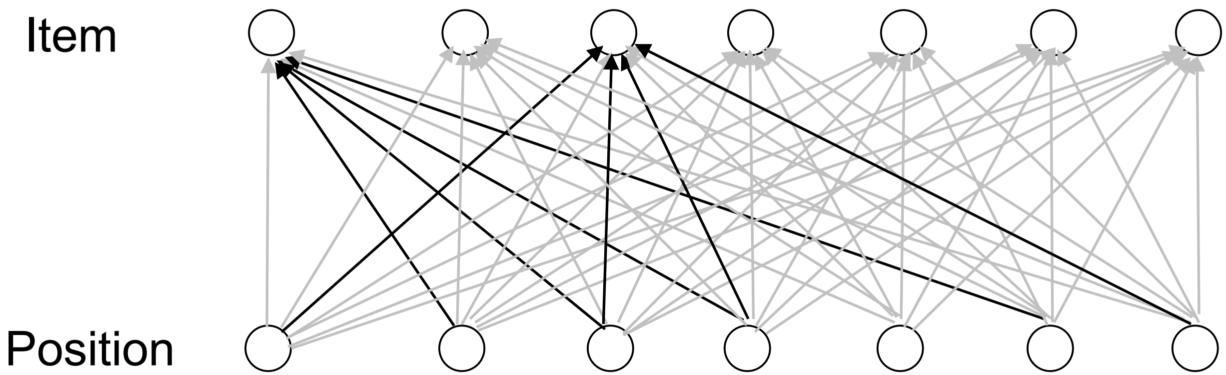
TBRS*: Encoding



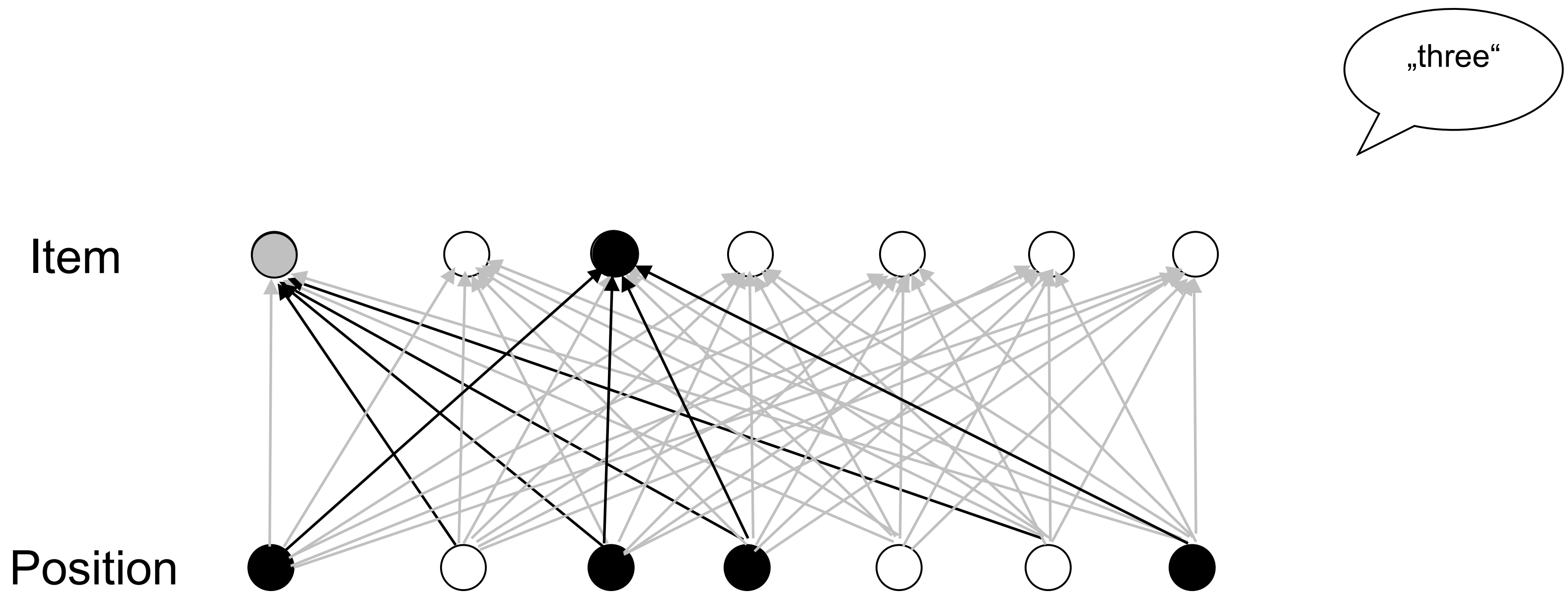
TBRS*: Retrieval

- Activate position markers in forward order
  → reproduce activation in item layer
- Pick the winner in item layer

TBRS*: Retrieval



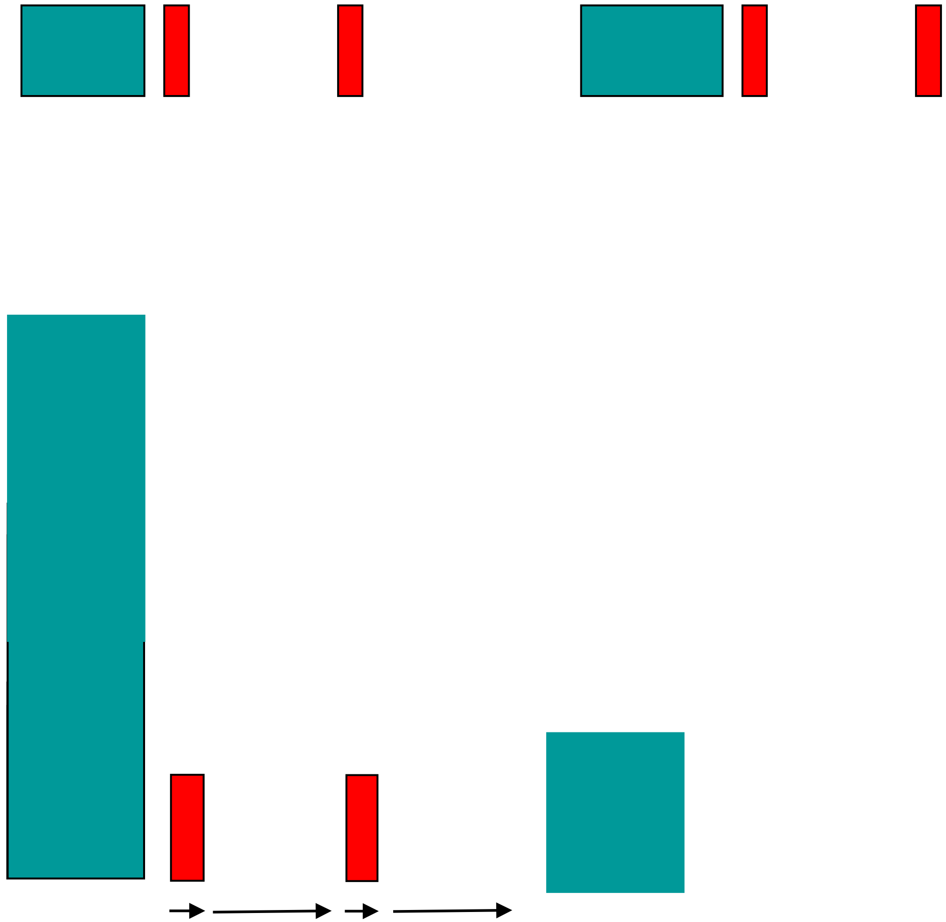
TBRS*: Retrieval



Decisions to be made (2)

- ...
- Cumulative learning (e.g., refreshing):
  - bound: weights saturate
  - unbound: weights grow indefinitely

Trouble with unbound learning



Decisions to be made (2)

- ...
- Cumulative learning:
  - bound or unbound

$$w_{ij} = w_{ij} + \eta_e (1 - w_{ij}) a_i a_j$$

Decisions to be made (2)

- ...
- Cumulative learning:
  - bound or unbound
- How to select item to be recalled
  - pick the winner, select with probability = f(act), ...
- Retrieval threshold?
- Noise added to item layer?

In between encoding and retrieval

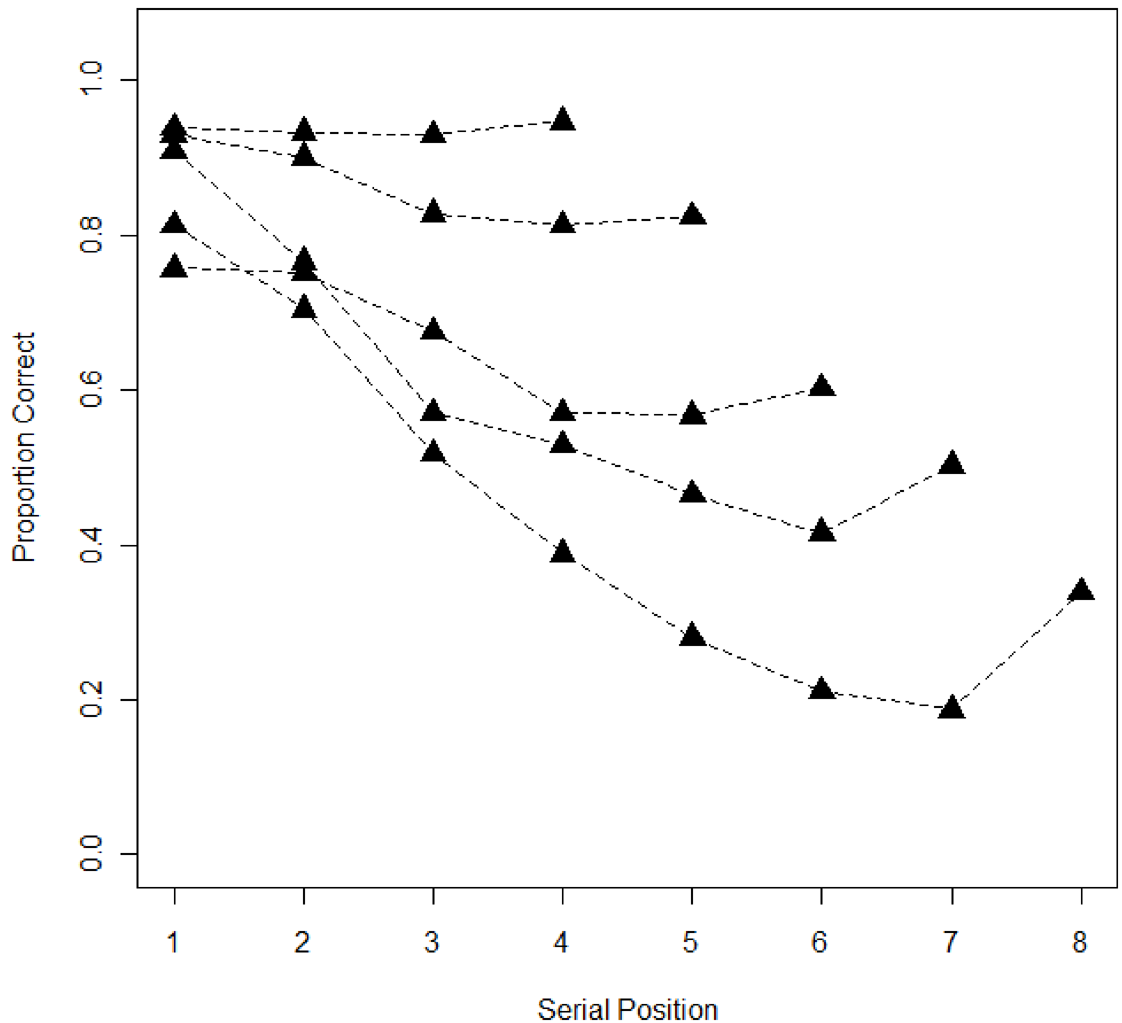
- Decay of weight matrix
  – exponential decline toward 0
- Refreshing
  – retrieve item + re-encode it

Decisions to be made (3)

- ...
- Refreshing schedule
  - only last item encoded
  - random selection
  - cumulative in forward order
    - start over after every interruption
    - start over only after new item

Refreshing the last item encoded

Serial Position Curves of Complex Span



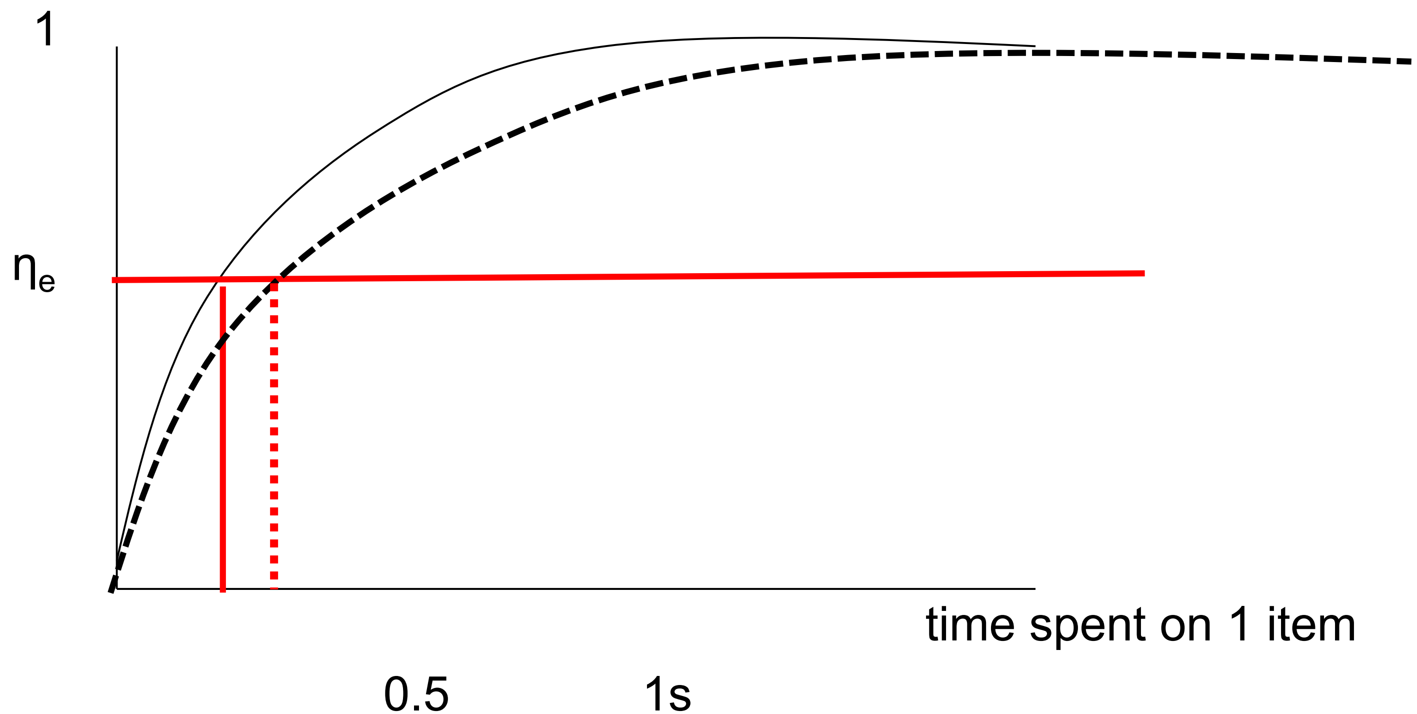
Decisions to be made (3)

- ...
- Refreshing schedule
  - only last item encoded
  - random selection
  - cumulative in forward order
    - start over after every interruption
    - start over only after new item
- Timing of refreshing: when to move on?
  - fixed time
  - fixed criterion of memory strength gained

Decisions to be made (3)

- ...
- Refreshing schedule
  - only last item encoded
  - random selection
  - cumulative in forward order
    - start over after every interruption
    - start over only after new item
- Timing of refreshing: when to move on?
  - fixed time
  - fixed criterion of memory strength gained

The dynamics of (encoding and) refreshing



→ Variability in rate translates into variability in refreshing duration

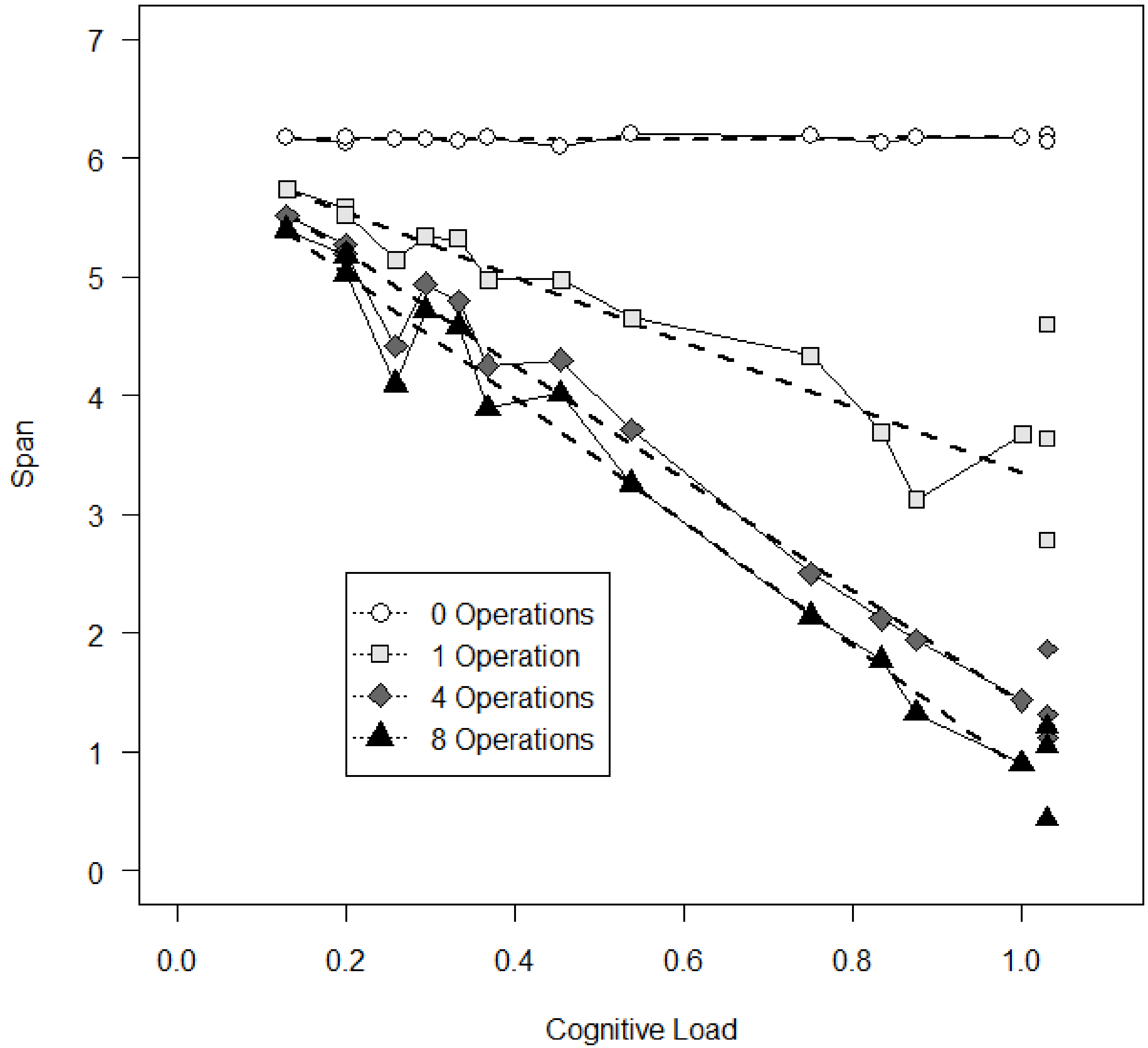
Now we're done

- We have an implementation of TBRS
- So we can run it

Simulation 1:
Big Complex Span Experiment

- Number of operations after each item: 0, 1, 4, 8
- Operation durations: 0.3, 0.5, 0.7 s
- Free times: 0, 0.1, 0.6, 1.2, 2.0 s

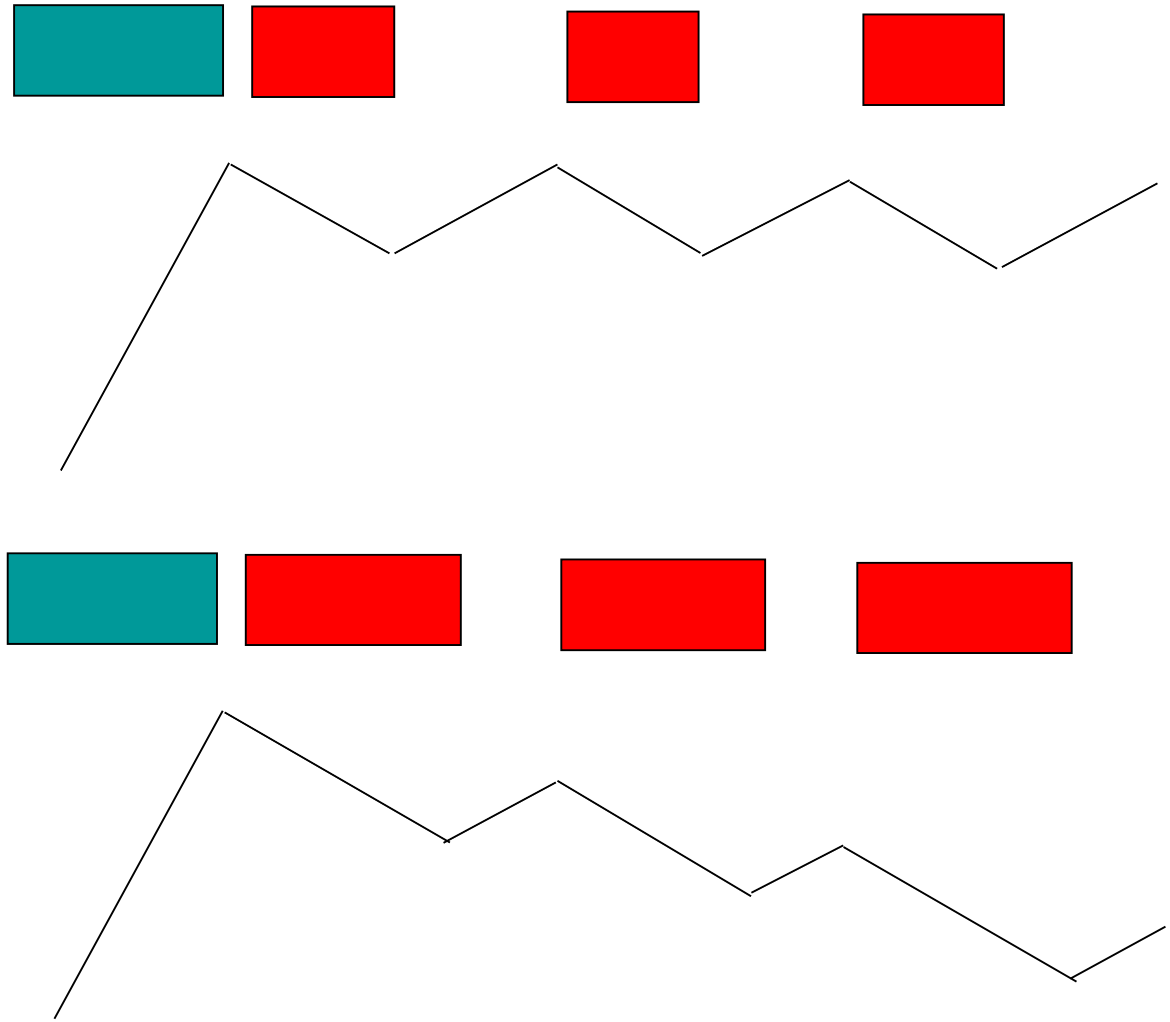
Span over Load



The effect of number of operations

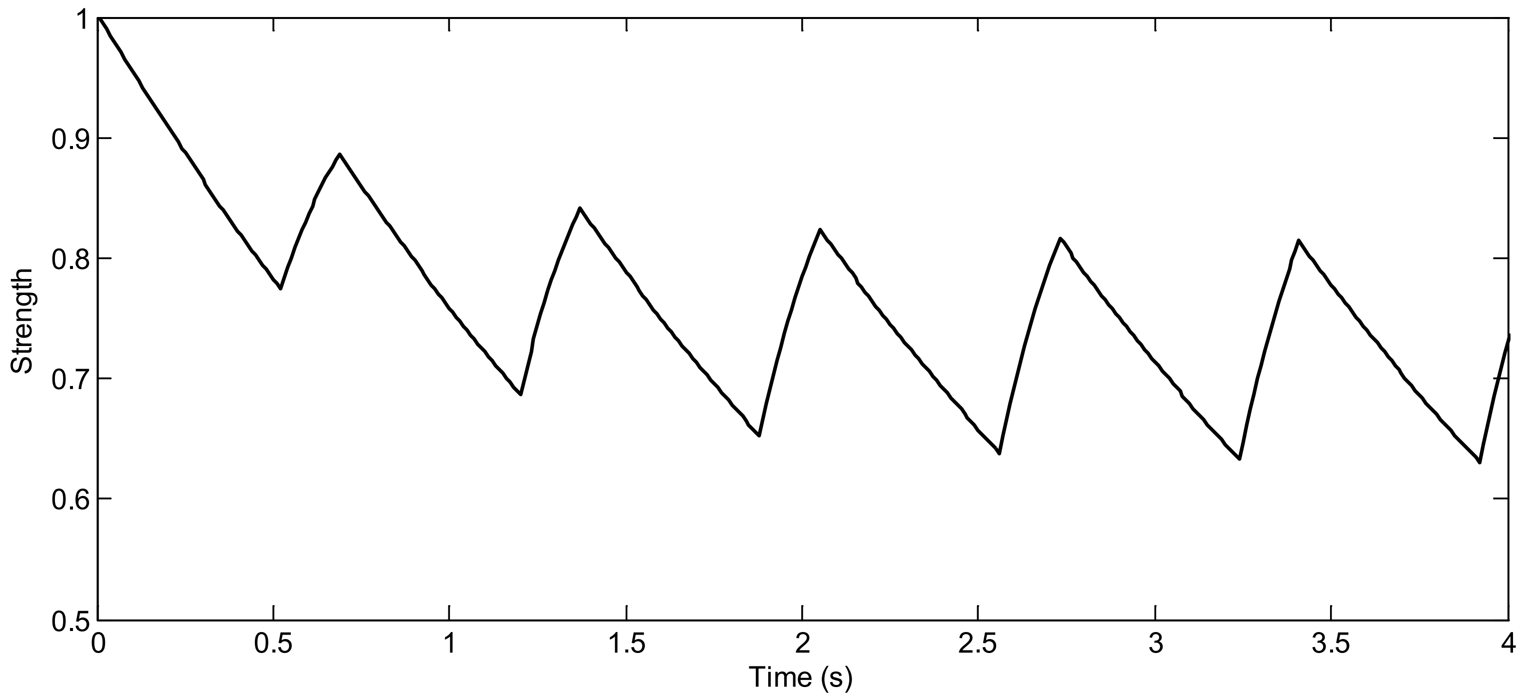
- Remember the intuitive prediction:
  CL = operation duration / total time
  Span declines (linearly) with CL
  At constant CL, no effect of number of distractor operations.

The effect of number of operations

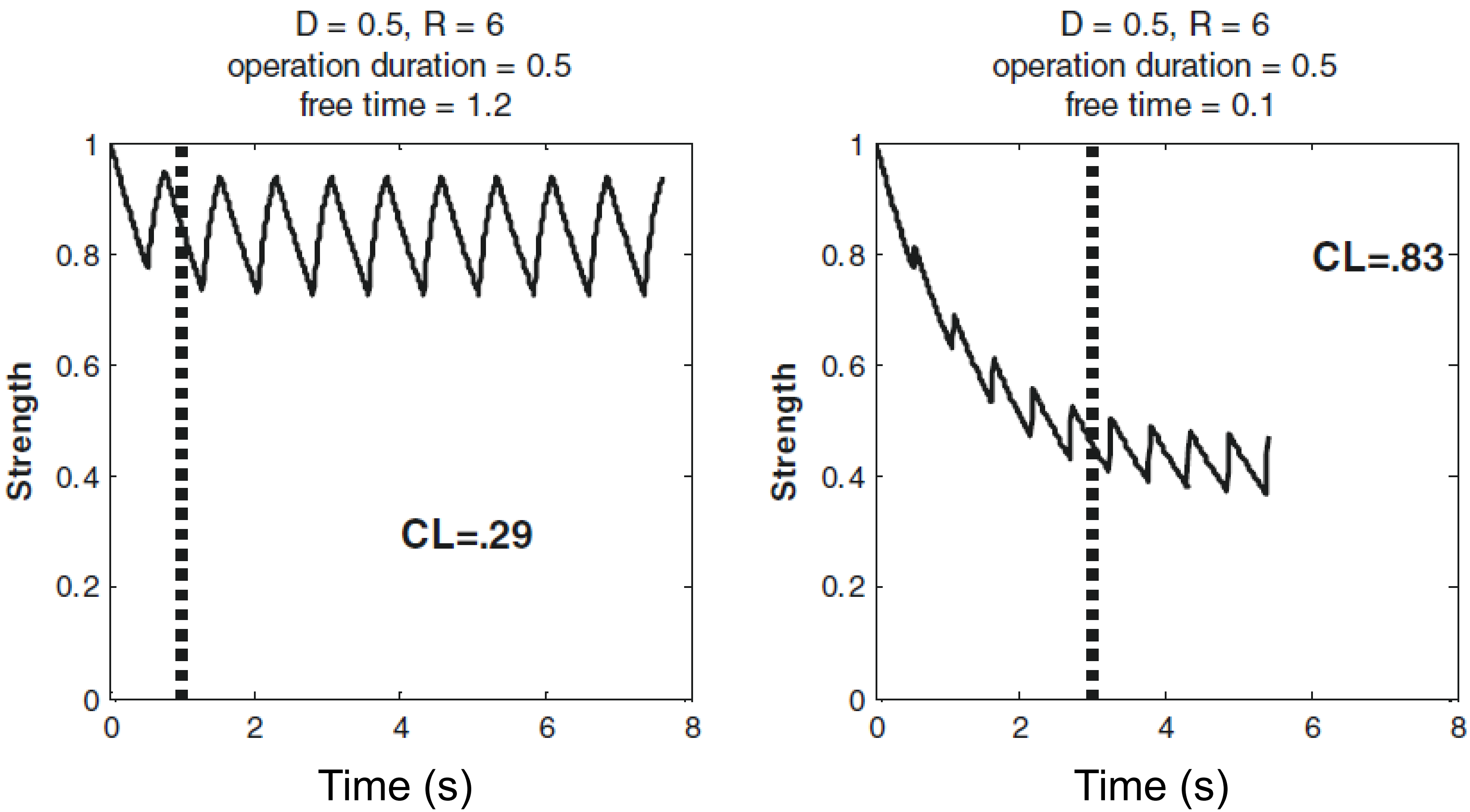


The equilibrium of decay and refreshing

- Decay is non-linear
- Refreshing gain is non-linear



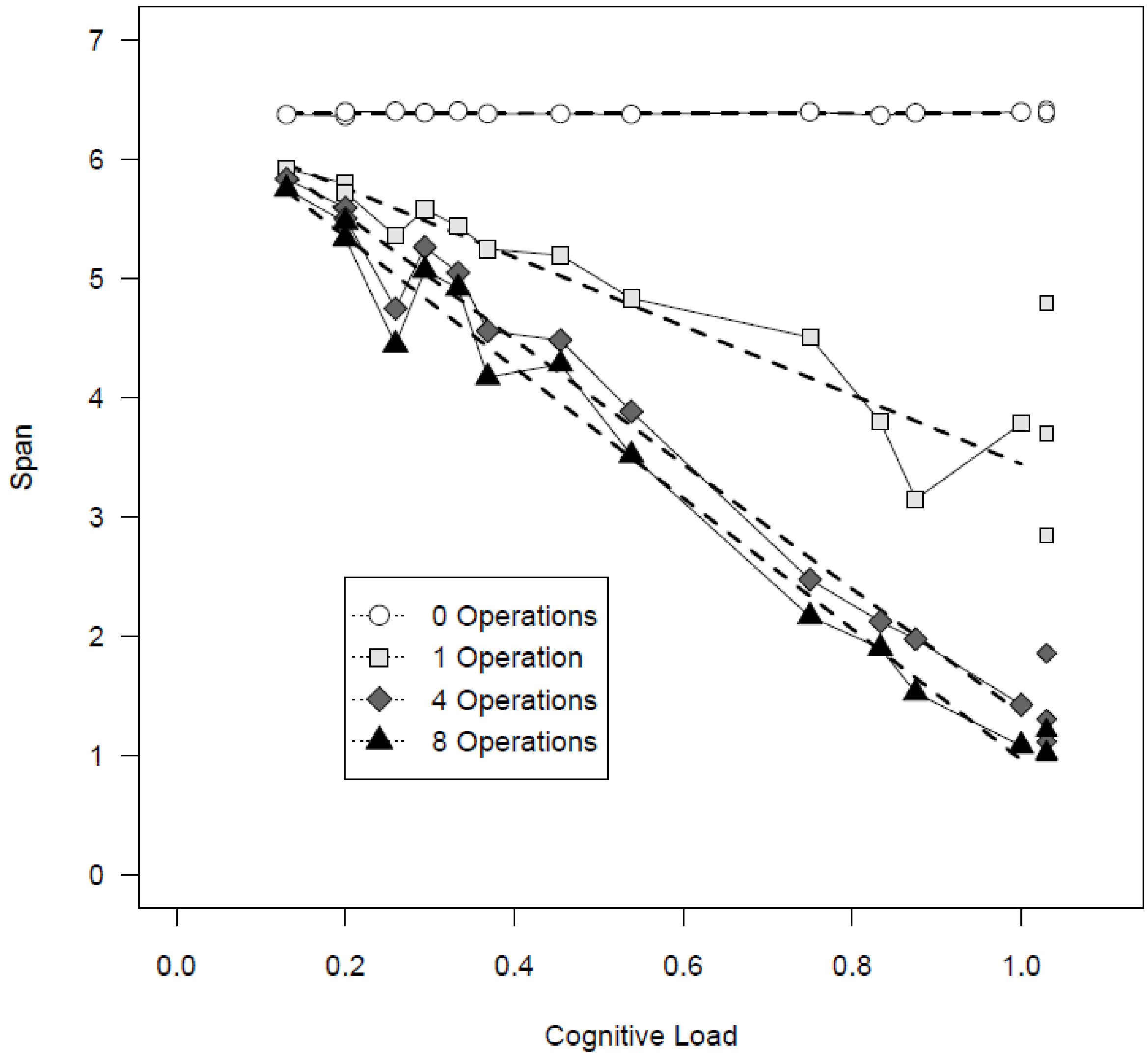
The Actual Effect of Number of Operations



Retrieval threshold?

- Intuition associated with decay: Memory strength falls below threshold → ☹
- So far, threshold = 0.05
- What if we set it to 0?

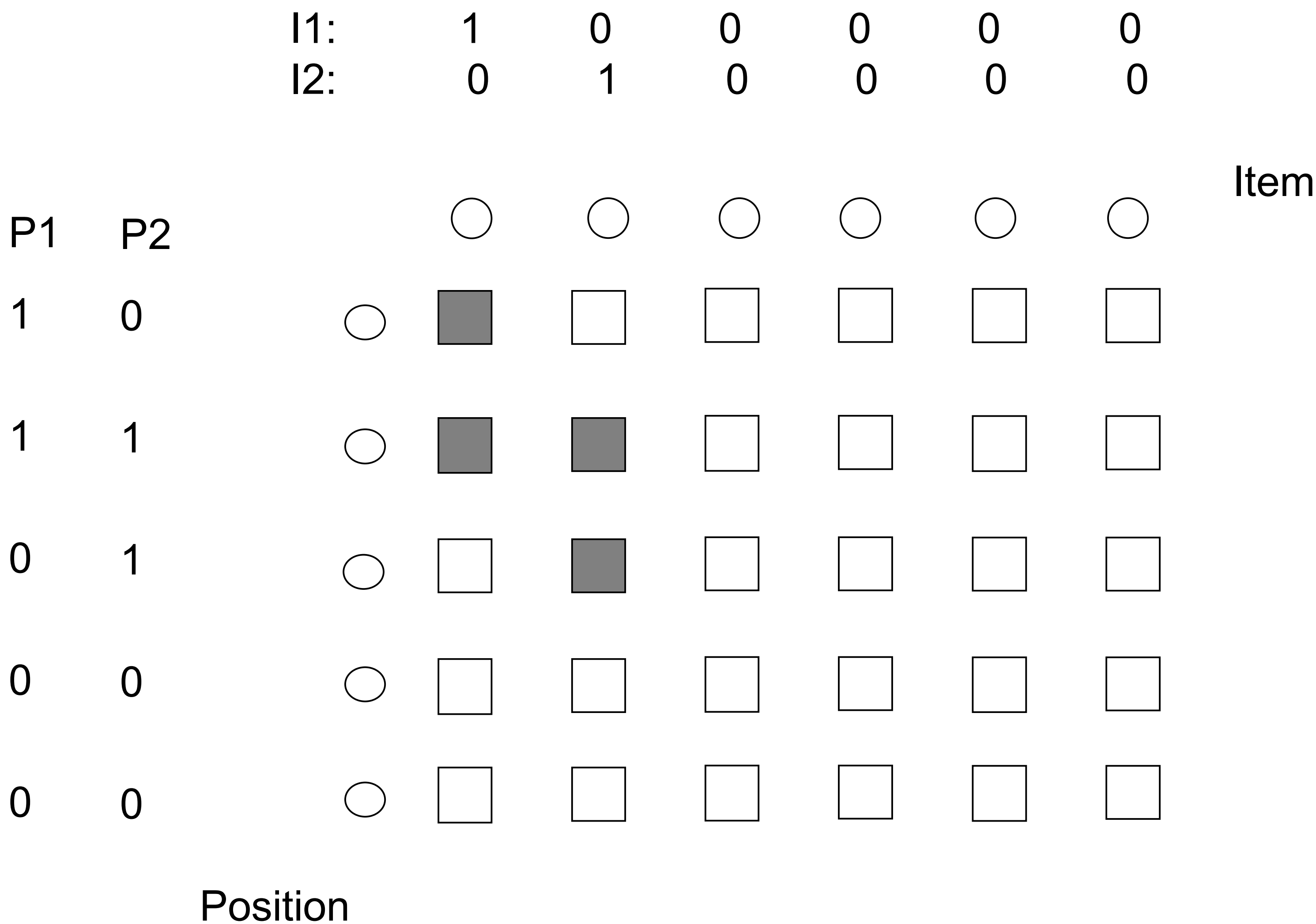
Span over load with threshold = 0



Absolute and relative memory strength

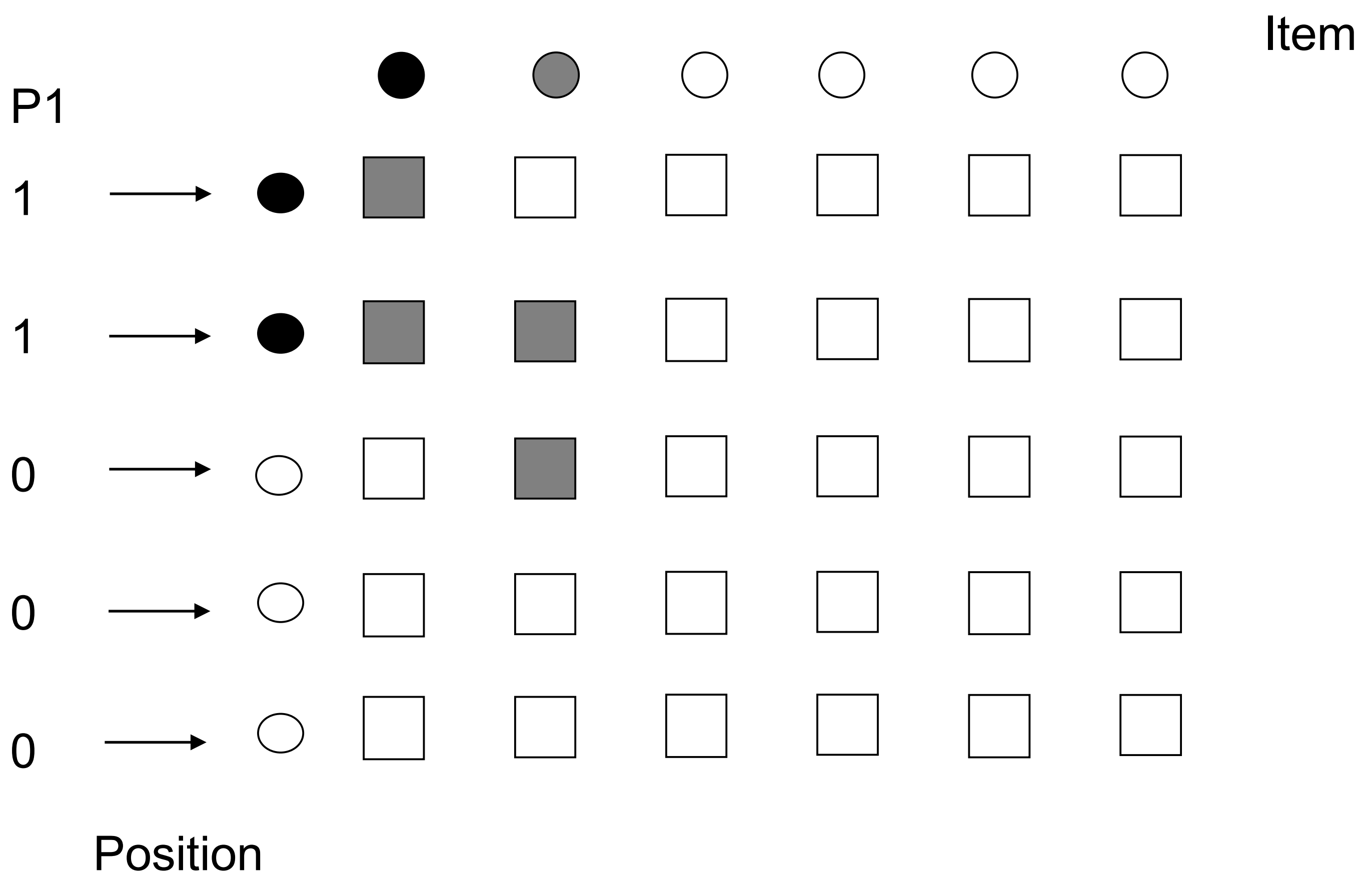
- How can items be forgotten if they never fall below threshold?
- Recall of the right item depends on relative activation in item layer (must be the „winner“!)

Position overlap and co-activation of competitors



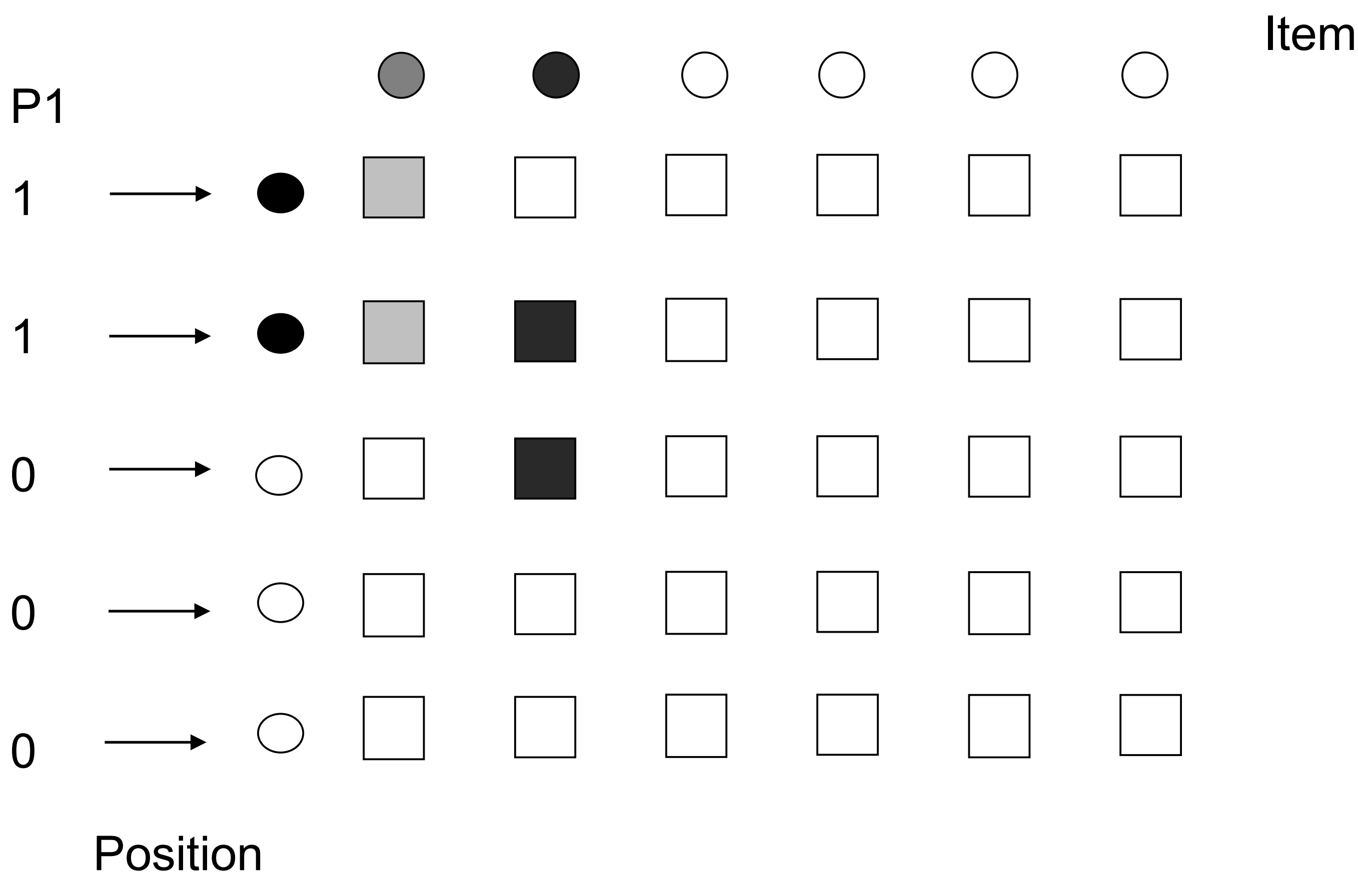
Retrieval, Scenario 1: Equal association strength

I1:	1	0	0	0	0	0
I2:	0	1	0	0	0	0



Retrieval, Scenario 2: Unequal association strength

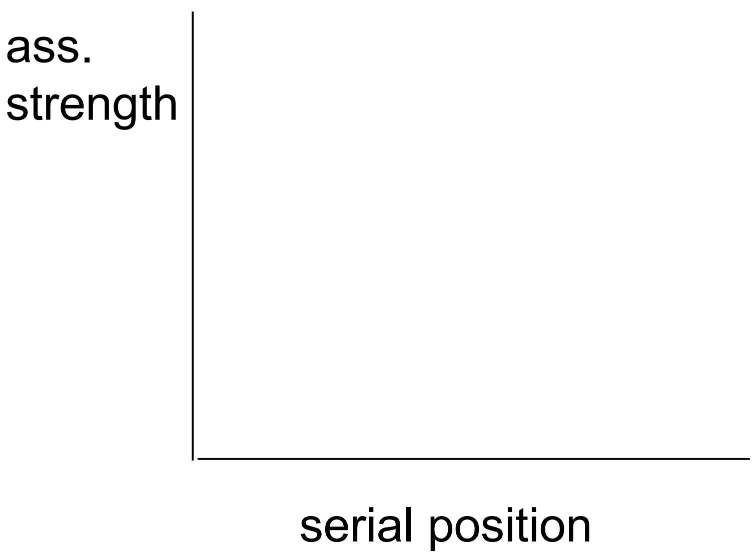
I1:	1	0	0	0	0	0
I2:	0	1	0	0	0	0



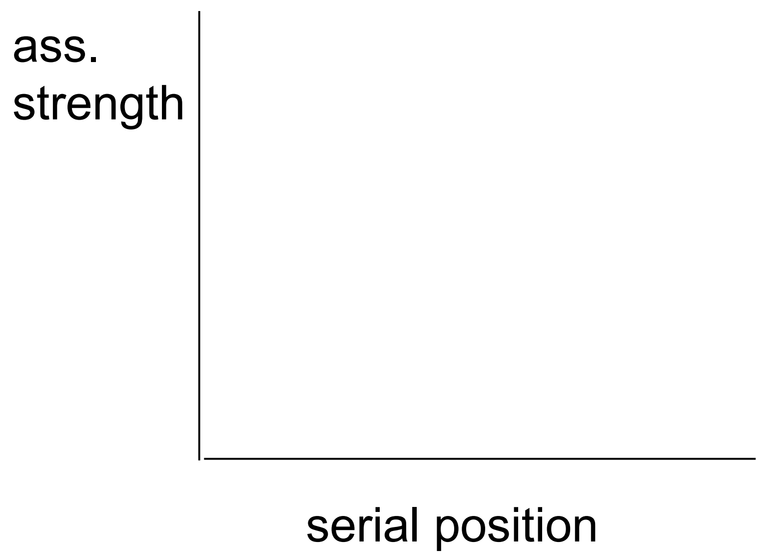
Where does unequal strength come from?

- Items are encoded at different times → different histories of decay and refreshing

At low cognitive load

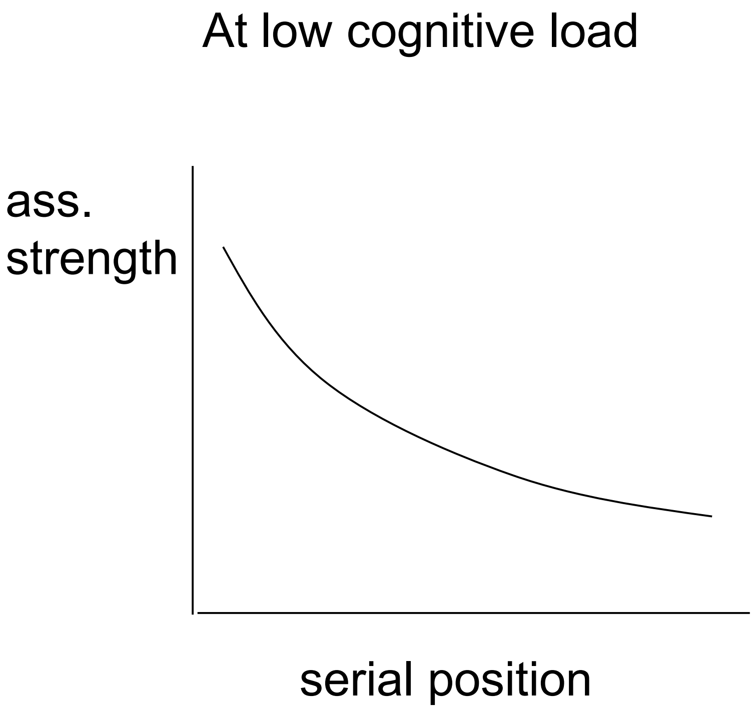


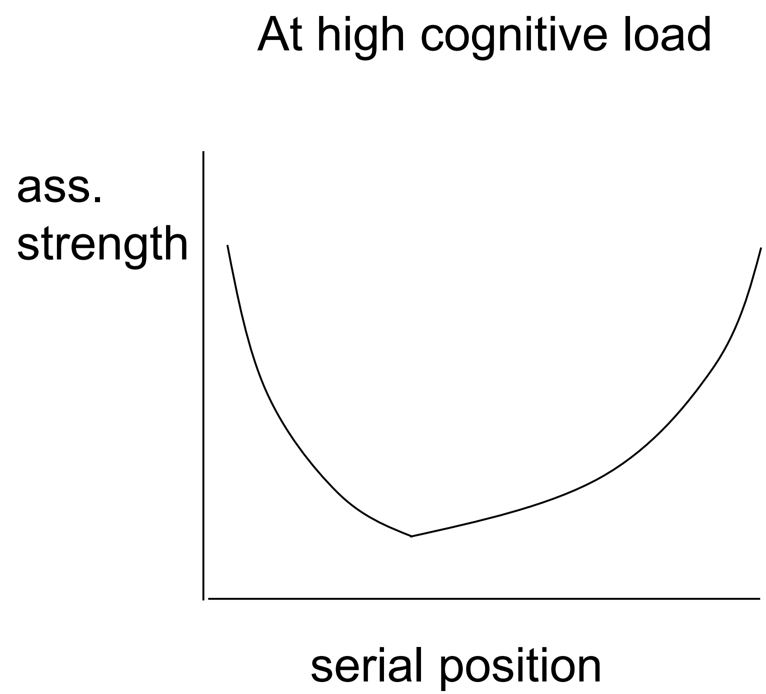
At high cognitive load



Where does unequal strength come from?

- Items are encoded at different times → different histories of decay and refreshing





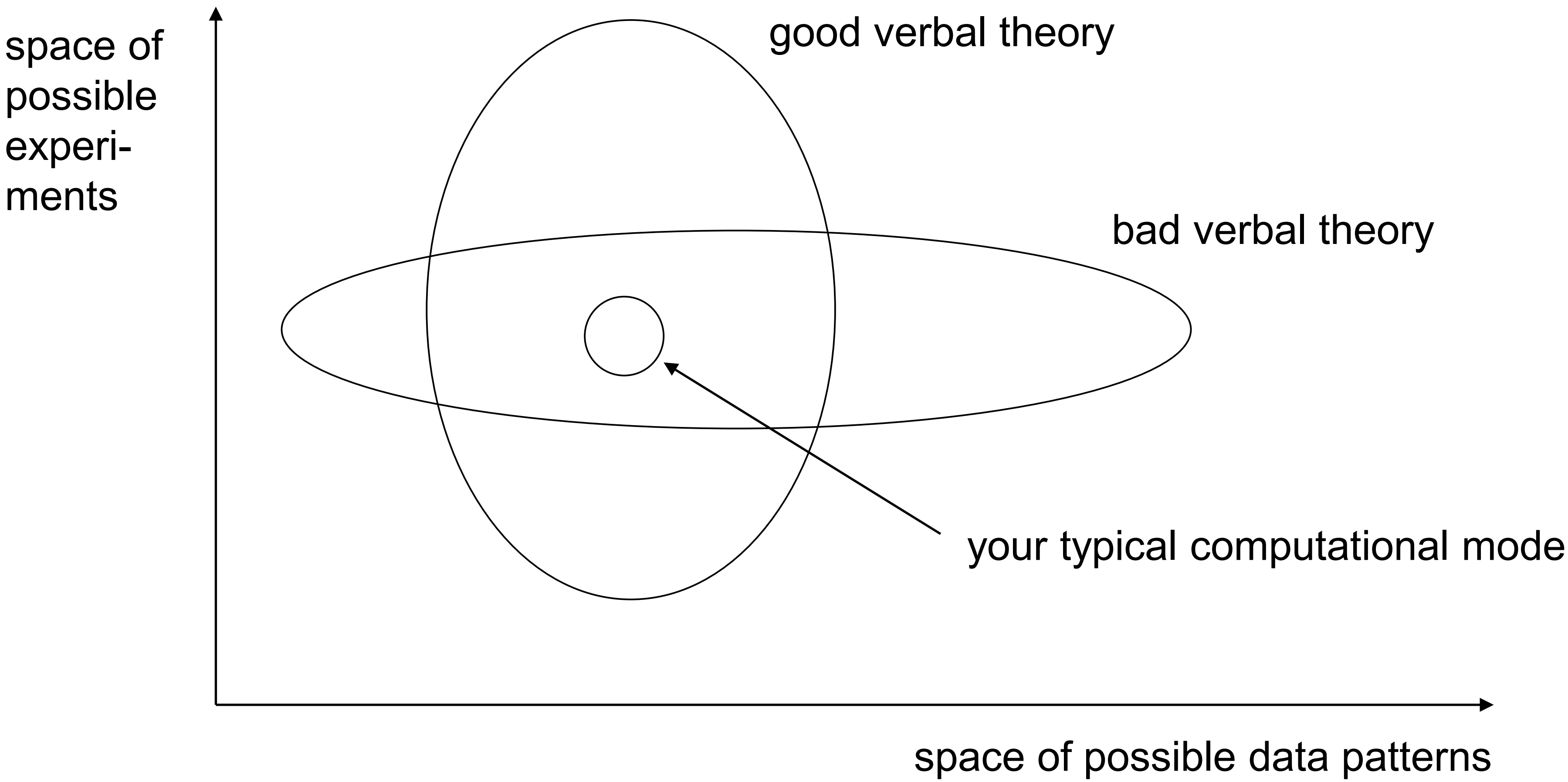
Conclusion

- TBRS* produces the effect of cognitive load predicted by verbal TBRS
- But for a different reason than the intuition underlying verbal TBRS

From TBRS to TBRS*

- 10 decision points → many other models are possible
  - most of them predict very weird data patterns
- There is (at least) one implementation that „works“, but
- It works for reasons not anticipated by intuition
- It makes in 1 case prediction different from intuitive prediction
- Through computational modelling you get to know your own theory

Trade-off btw. scope and precision



Trade-off btw. scope and precision