Introduction to ACT-R

Mike Matessa
NASA Ames Research Center

Rules for ACTSet

In the game of ACTSet, 12 cards are played face up on the table. The goal is to be the first player to identify a so-called Set

What are the cards like?
There are 81 cards, each of which have four attributes:

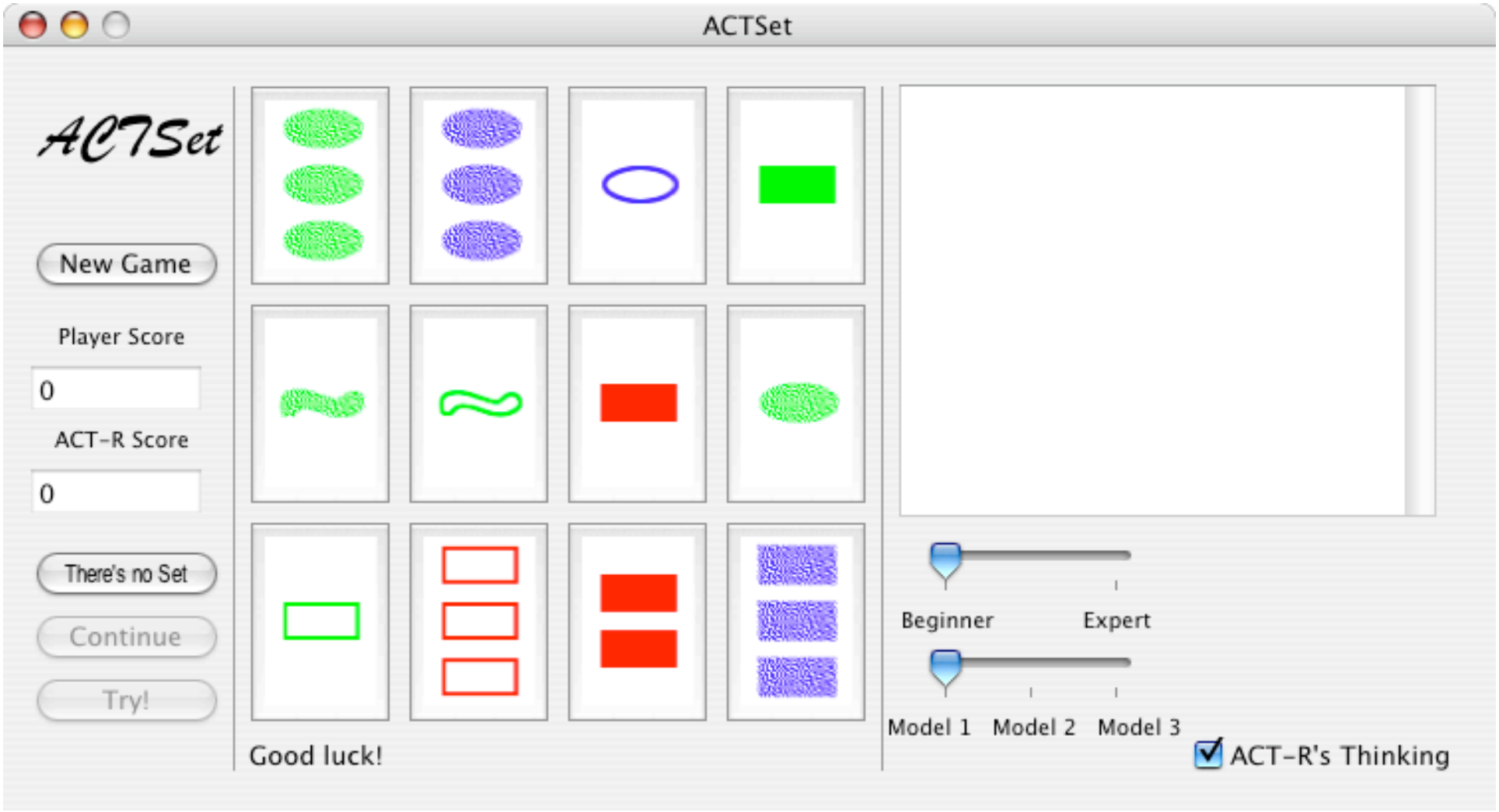
Color: Red, Blue or Green
Shape: Oval, Rectangle or Wave
Number: One, Two or Three
Filling: Filled, Empty or Gray

What is a Set?
A set consist of three cards, in which each of the attributes is **the same** for all three cards, or **different** for all three cards. So the following three cards are a set:

One red filled oval
Two red empty ovals
Three red grey ovals

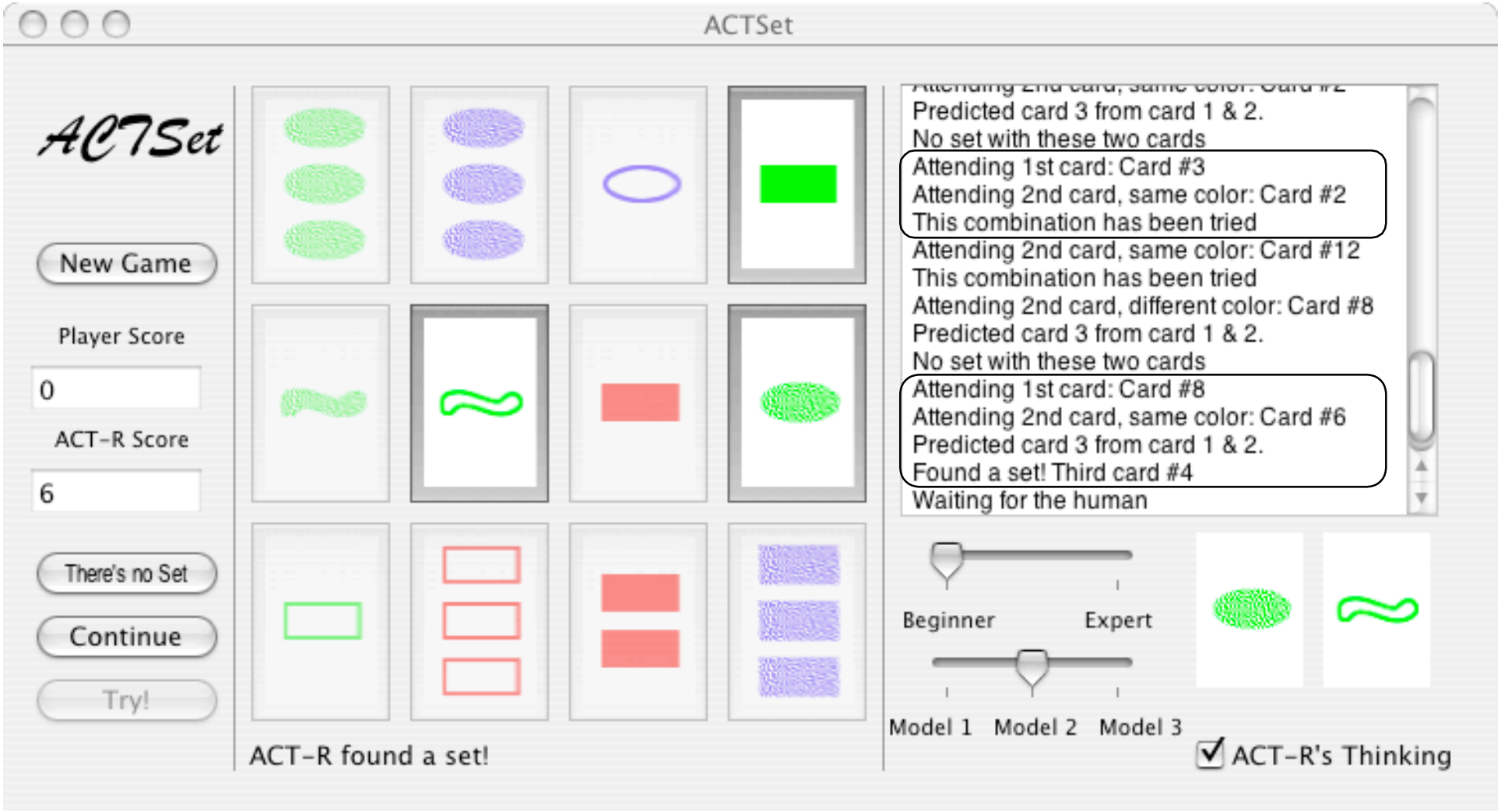
In this case the color and the shape are all the same, and the number and fillings are all different.

Color: Red, Blue or Green
Shape: Oval, Rectangle or Wave
Number: One, Two or Three
Filling: Filled, Empty or Gray



Can you find a set? How did you do it?

Color: Red, Blue or Green
Shape: Oval, Rectangle or Wave
Number: One, Two or Three
Filling: Filled, Empty or Gray



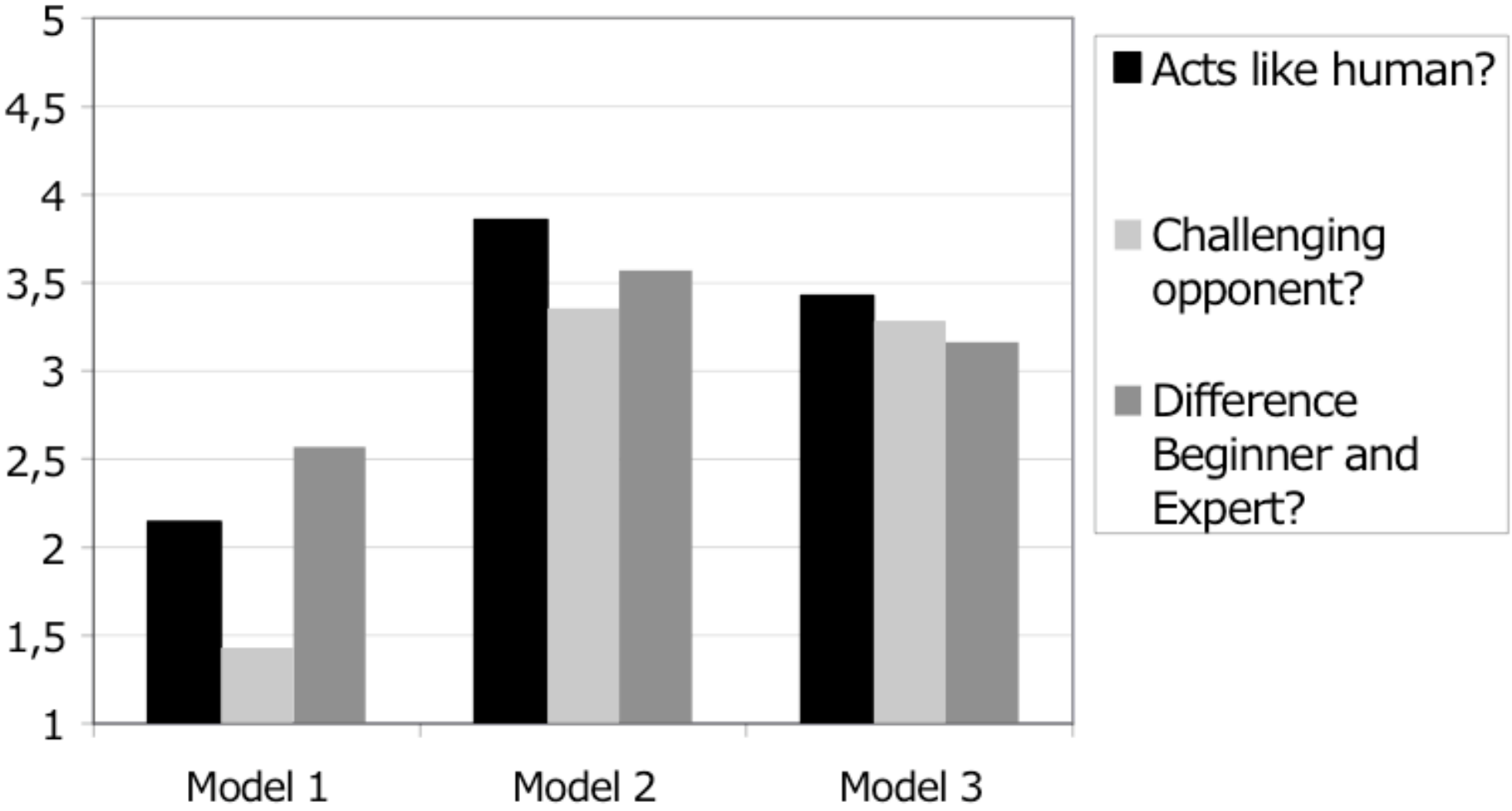
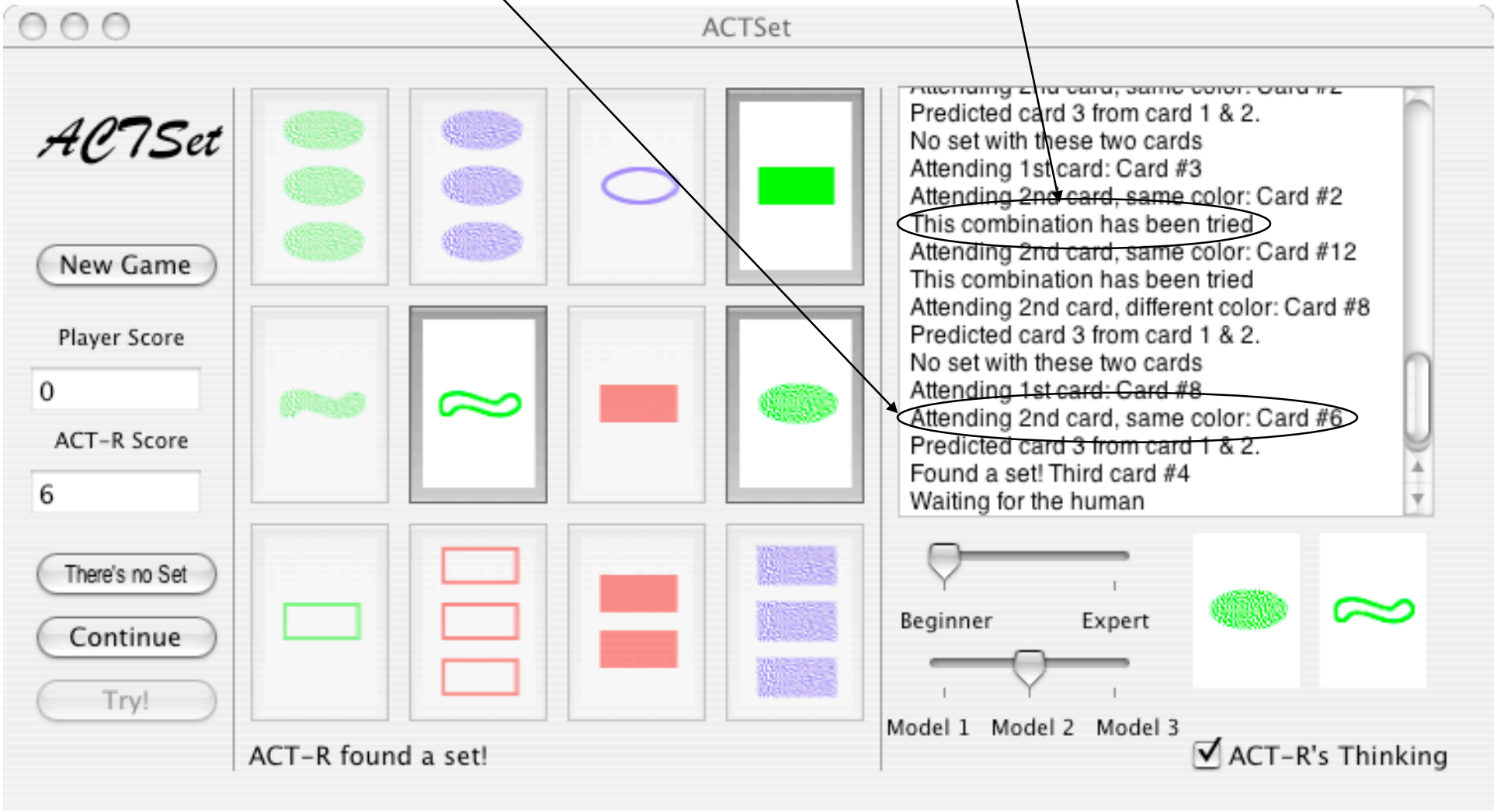


Figure 6. Results of the evaluation of the application: the three questions for each model are rated on a 5 point scale.

Rules for what to do next

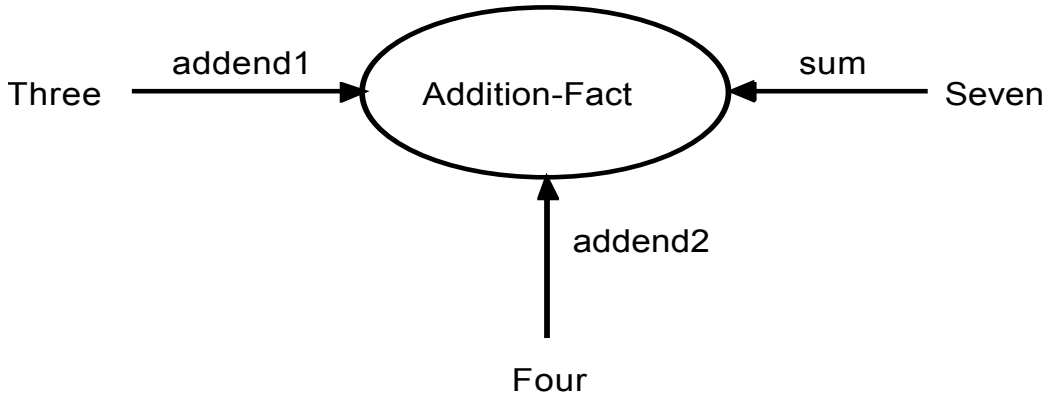
Memory for past events



ACT-R: Knowledge Representation

Declarative-Procedural Distinction

Declarative Knowledge: Chunks
Configurations of small numbers of elements



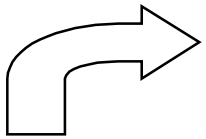
Procedural Knowledge: Production Rules
for retrieving chunks to solve problems.

```
  336
+848
  4
```

IF the goal is to add the numbers in a column
and n1 + n2 are in the column
THEN retrieve the sum of n1 and n2.

Productions serve to coordinate the retrieval of information from declarative memory and the enviroment to produce transformations in the goal state.

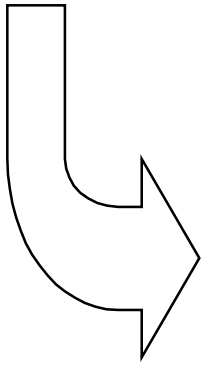
ACT-R: Assumption Space



Performance

	Declarative	**Procedural**
Symbolic	Retrieval of Chunks	Application of Production Rules
Subsymbolic	Noisy Activations Control Speed and Accuracy	Noisy Utilities Control Choice

the "R" in ACT-R



Learning

	Declarative	**Procedural**
Symbolic	Encoding Environment and Caching Goals	Production Compilation
Subsymbolic	Bayesian Learning	Bayesian Learning

Subsymbolic Level

The subsymbolic level reflects an analytic characterization of connectionist computations. These computations have been implemented in ACT-RN (Lebiere & Anderson, 1993) but this is not a practical modeling system.

Chunk Activations are responsible for determining which (if any chunks) get retrieved and how long it takes to retrieve them.

Production Utilities are responsible for determining which productions get selected when there is a conflict.

As with the symbolic level, the subsymbolic level is not a static level, but is changing in the light of experience. Subsymbolic learning allows the system to adapt to the statistical structure of the environment.

Chunk Activation

activation = base activation + (source activation * associative strength) + (mismatch penalty * similarity value) + noise

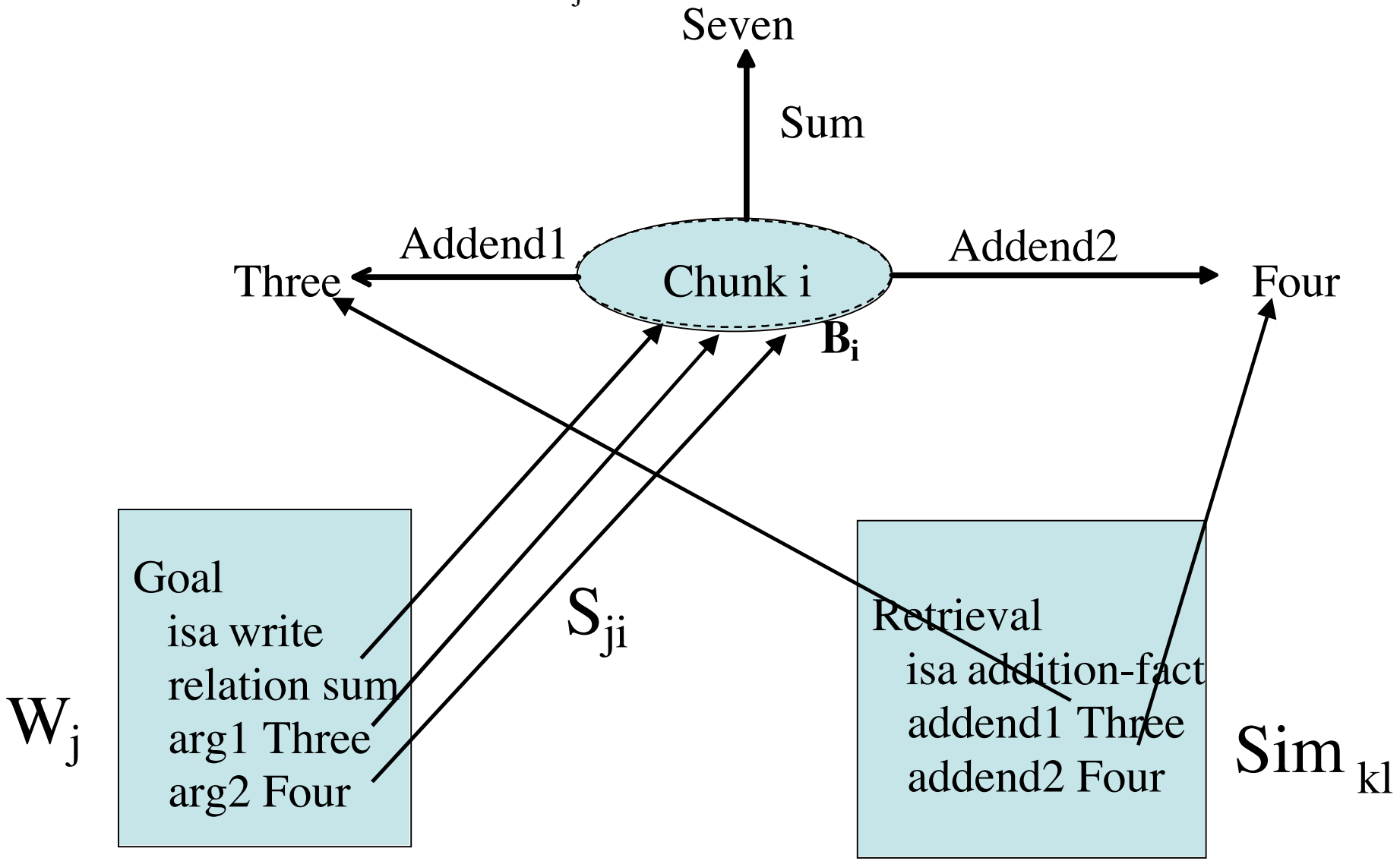
$$A_i = B_i + \sum_j W_j \cdot S_{ji} + \sum_k MP_k \cdot Sim_{kl} + N(0,s)$$

Activation makes chunks available to the degree that past experiences indicate that they will be useful at the particular moment:

- Base-level: general past usefulness
- Associative Activation: relevance to the current context
- Matching Penalty: allows partial matching of context
- Noise: useful to avoid getting stuck in local minima

Activation

$$A_i = B_i + \sum_j W_j \cdot S_{ji} + \sum_k MP_k \cdot Sim_{kl} + N(0,s)$$



Activation, Latency and Probability

- Retrieval time for a chunk is a negative exponential function of its activation:

$$Time_i = F \cdot e^{-A_i}$$

- Probability of retrieval of a chunk follows the Boltzmann (softmax) distribution:

$$t = \sqrt{2} \cdot s = \frac{\sqrt{6} \cdot \sigma}{\pi}$$

$$P_i = \frac{e^{A_i / t}}{\sum_j e^{A_j / t}}$$

- The chunk with the highest activation is retrieved

Base-level Activation

activation = base activation

$A_i = B_i$

The base level activation B_i of chunk C_i reflects a context-independent estimation of how likely C_i is to match a production, i.e. B_i is an estimate of the log odds that C_i will be used.

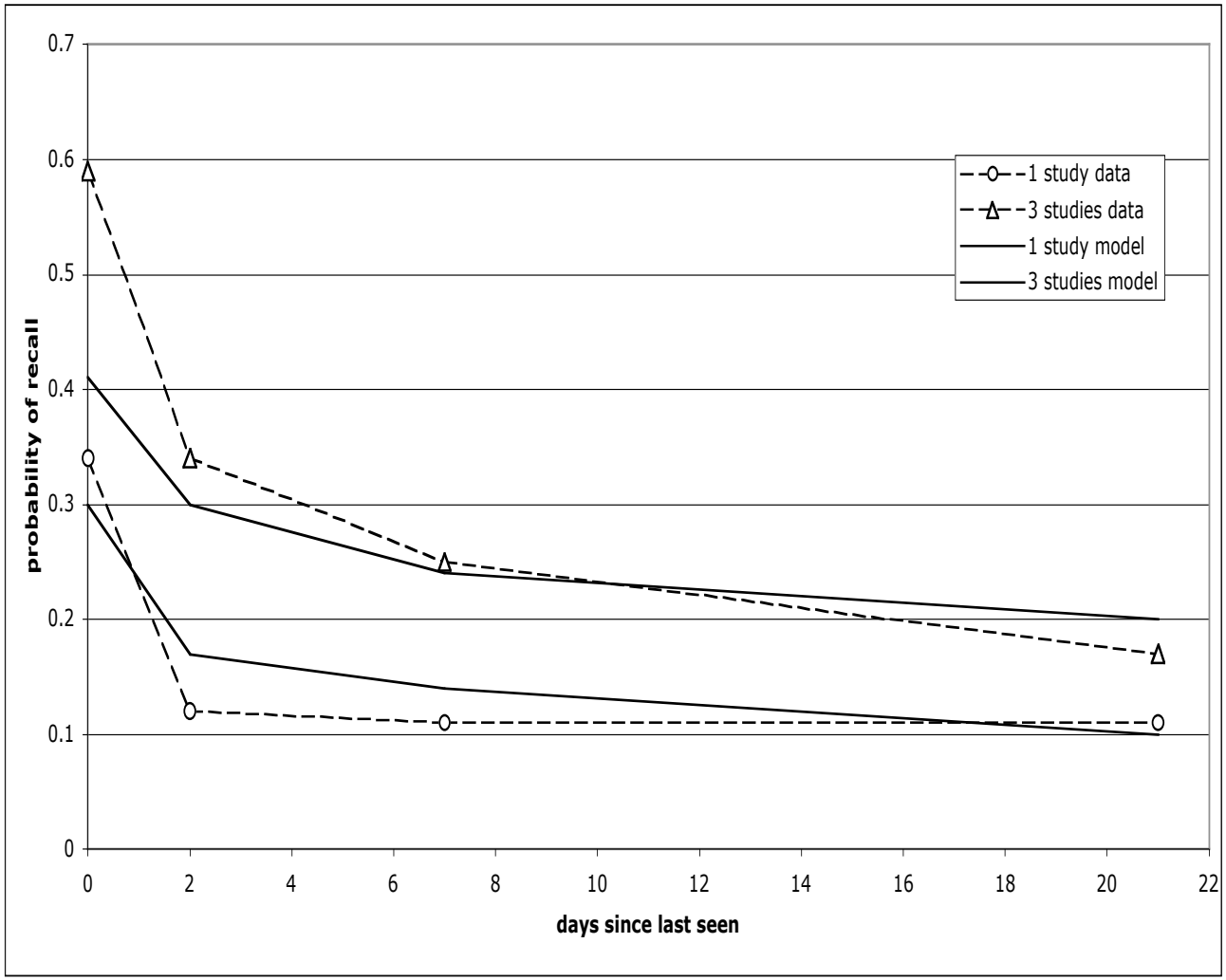
Two factors determine B_i:

- frequency of using C_i
- recency with which C_i was used

$$B_i = \ln\left(\frac{P(C_i)}{P(\overline{C_i})}\right)$$

(talk about learning later)

Application: Paired Associate Recall



Source Activation

$$+ \left(\text{source activation} * \text{associative strength} \right)$$

$$+ \sum_{j} W_j * S_{ji}$$

The source activations W_j reflect the amount of *attention* given to elements, i.e. fillers, of the current goal. ACT-R assumes a *fixed capacity* for source activation

$W = \Sigma W_j$ reflects an individual difference parameter.
People with more W can retrieve more information

Application: Individual Differences

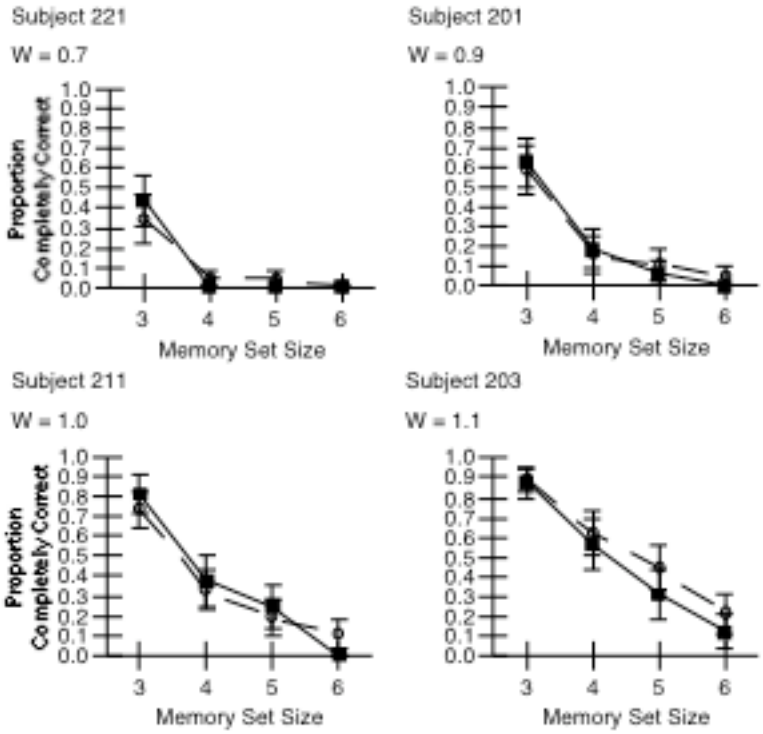


Figure 5. Model fits for four representative subjects from Daily et al. (1999). Filled symbols are subject data, open symbols are the model's predictions.

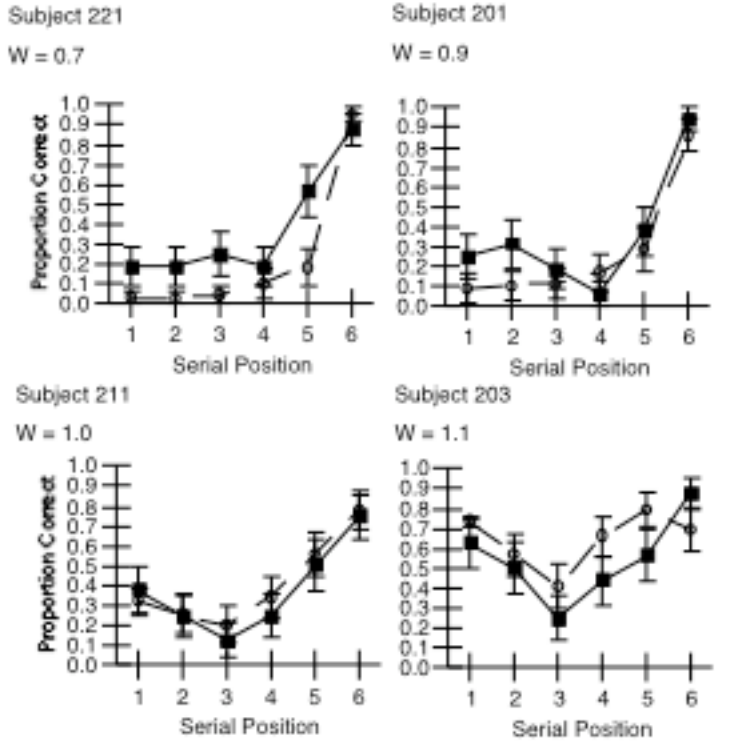


Figure 6. Fits to the serial position data for 4 typical subjects (largest set size only). Filled symbols are subject data, open symbols are the model's predictions.

Associative Strengths

$$+ \left(\text{source activation} * \text{associative strength} \right)$$

$$+ \sum W_j * S_{ji}$$

The association strength S_{ji} between chunks C_j and C_i is a measure of how often C_i was needed (retrieved) when C_j was element of the goal, i.e. S_{ji} estimates the log likelihood ratio of C_j being a source of activation if C_i was retrieved.

$$S_{ji} = \ln \left(\frac{P(N_i|C_j)}{P(N_i)} \right)$$

$$= S - \ln(P(N_i|C_j))$$

Application: Fan Effect

TABLE 3.3
Observed Times (in Seconds) to Accept Targets and Reject Foils
Data from Anderson (1974), with Predictions of ACT-R in Brackets

	Targets, Facts About Person			
Facts About Location	1	2	3	Mean
1	1.11	1.17	1.22	1.17
	[1.08]	[1.14]	[1.18]	[1.13]
2	1.17	1.20	1.22	1.20
	[1.14]	[1.22]	[1.27]	[1.21]
3	1.15	1.23	1.36	1.25
	[1.18]	[1.27]	[1.33]	[1.26]
Mean	1.14	1.20	1.27	1.20
	[1.13]	[1.21]	[1.26]	[1.20]

	Foils, Facts About Person			
Facts About Location	1	2	3	Mean
1	1.20	1.22	1.26	1.23
	[1.22]	[1.27]	[1.31]	[1.27]
2	1.25	1.36	1.29	1.30
	[1.27]	[1.32]	[1.36]	[1.32]
3	1.26	1.47	1.47	1.40
	[1.31]	[1.36]	[1.39]	[1.35]
Mean	1.24	1.35	1.34	1.31
	[1.27]	[1.32]	[1.35]	[1.31]

Partial Matching

$$+\left(\begin{array}{c}\text{mismatch}\\ \text{penalty}\end{array} * \begin{array}{c}\text{similarity}\\ \text{value}\end{array}\right)$$

$$+\sum_k MP_k \cdot Sim_{kl}$$

- The mismatch penalty is a measure of the amount of control over memory retrieval: MP = 0 is free association; MP very large means perfect matching; intermediate values allow some mismatching in search of a memory match.
- Similarity values between desired value k specified by the production and actual value l present in the retrieved chunk. This provides generalization properties similar to those in neural networks; the similarity value is essentially equivalent to the dot-product between distributed representations.

Application: Cognitive Arithmetic

Table 3.1
Data from Siegler & Shrager (1984)
and ACT-R's Predictions

Data

Problem	Answer									Other Including Retrieval Failure
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
1+1	-	.05	.86	-	.02	-	.02	-	-	.06
1+2	-	.04	.07	.75	.04	-	.02	-	-	.09
1+3	-	.02	-	.10	.75	.05	.01	.03	-	.06
2+2	.02	-	.04	.05	.80	.04	-	.05	-	-
2+3	-	-	.07	.09	.25	.45	.08	.01	.01	.06
3+3	.04	-	-	.05	.21	.09	.48	-	.02	.11

Predictions

Problem	Answer									Other Including Retrieval Failure
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
1+1	-	.10	.75	.10	.01	-	-	-	-	.04
1+2	-	.01	.10	.75	.10	.-	-	-	-	.04
1+3	-	-	.01	.10	.78	.06	-	-	-	.04
2+2	-	-	.0	.1	.82	.02	-	-	-	.04
2+3	-	-	-	.03	.32	.45	.06	.01	-	.13
3+3	-	-	-	.04	.04	.08	.61	.08	.01	.18

Noise

\+ noise

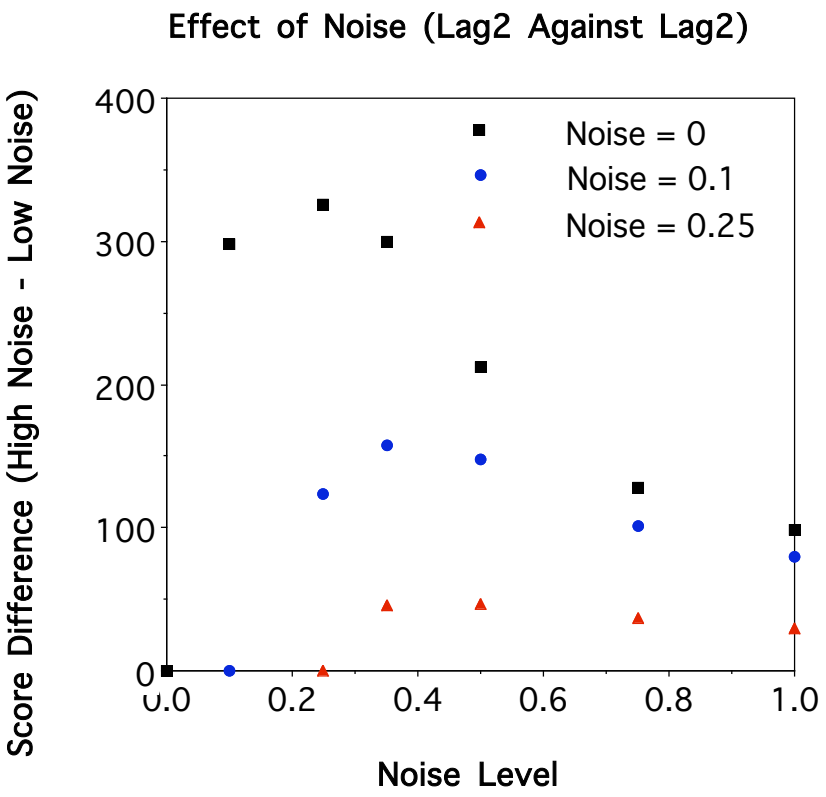
$$+N(0,s)$$

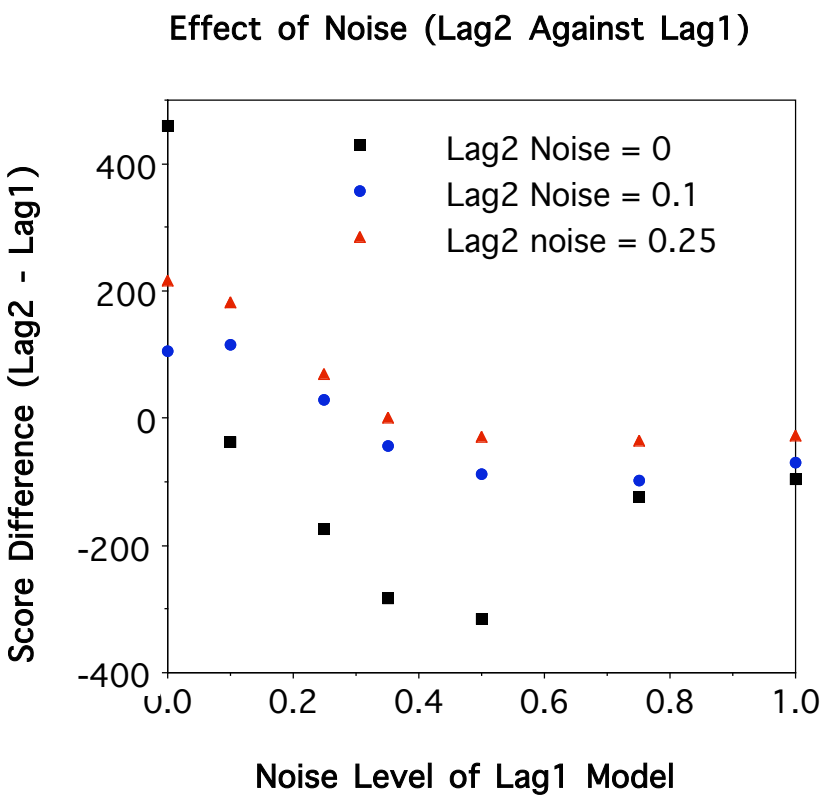
- Noise provides the essential stochasticity of human behavior
- Noise also provides a powerful way of exploring the world
- Activation noise is composed of two noises:
  - A permanent noise accounting for encoding variability
  - A transient noise for moment-to-moment variation

Application: Paper Rocks Scissors

(Lebiere & West, 1999)

- Too little noise makes the system too deterministic.
- Too much noise makes the system too random.
- This is not limited to game-playing situations!





Production Utility

Making Choices: Conflict Resolution

Expected Gain = E = PG-C

P is expected probability of success
G is value of goal
C is expected cost

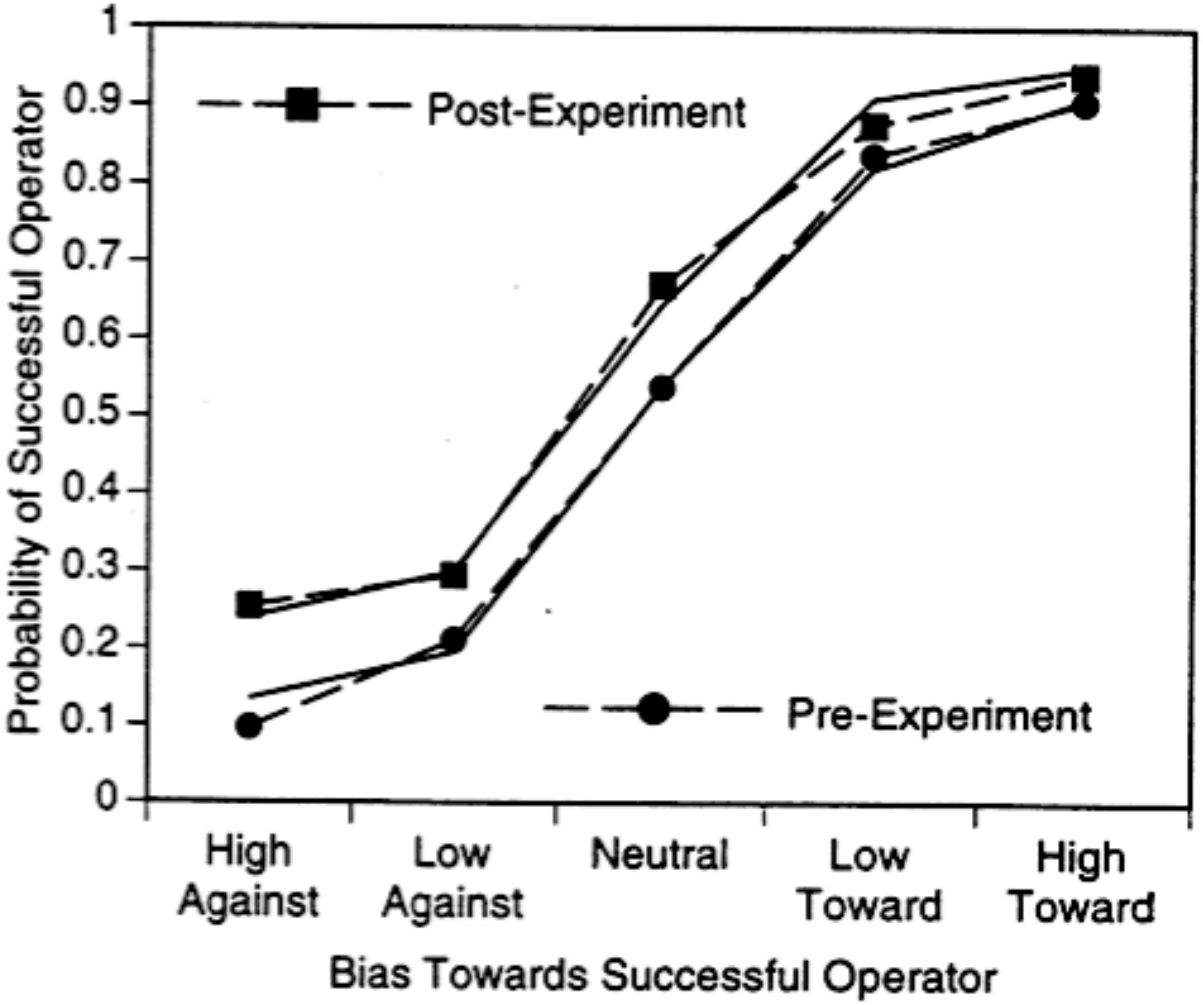
Probability of choosing i = $\frac{e^{E_i/t}}{\sum_j e^{E_j/t}}$

t reflects noise in evaluation and is like temperature in the Bolztman equation

(talk about learning later)

Application: Operator Selection

Probability of Using an Operator as a Function of the Bias Towards That Operator Both Before and After Experiencing Success With That Operator



ACT-R: Assumption Space

	Performance	
	Declarative	**Procedural**
Symbolic	Retrieval of Chunks	Application of Production Rules
Subsymbolic	Noisy Activations Control Speed and Accuracy	Noisy Utilities Control Choice

	Learning	
	Declarative	**Procedural**
Symbolic	Encoding Environment and Caching Goals	Production Compilation
Subsymbolic	Bayesian Learning	Bayesian Learning

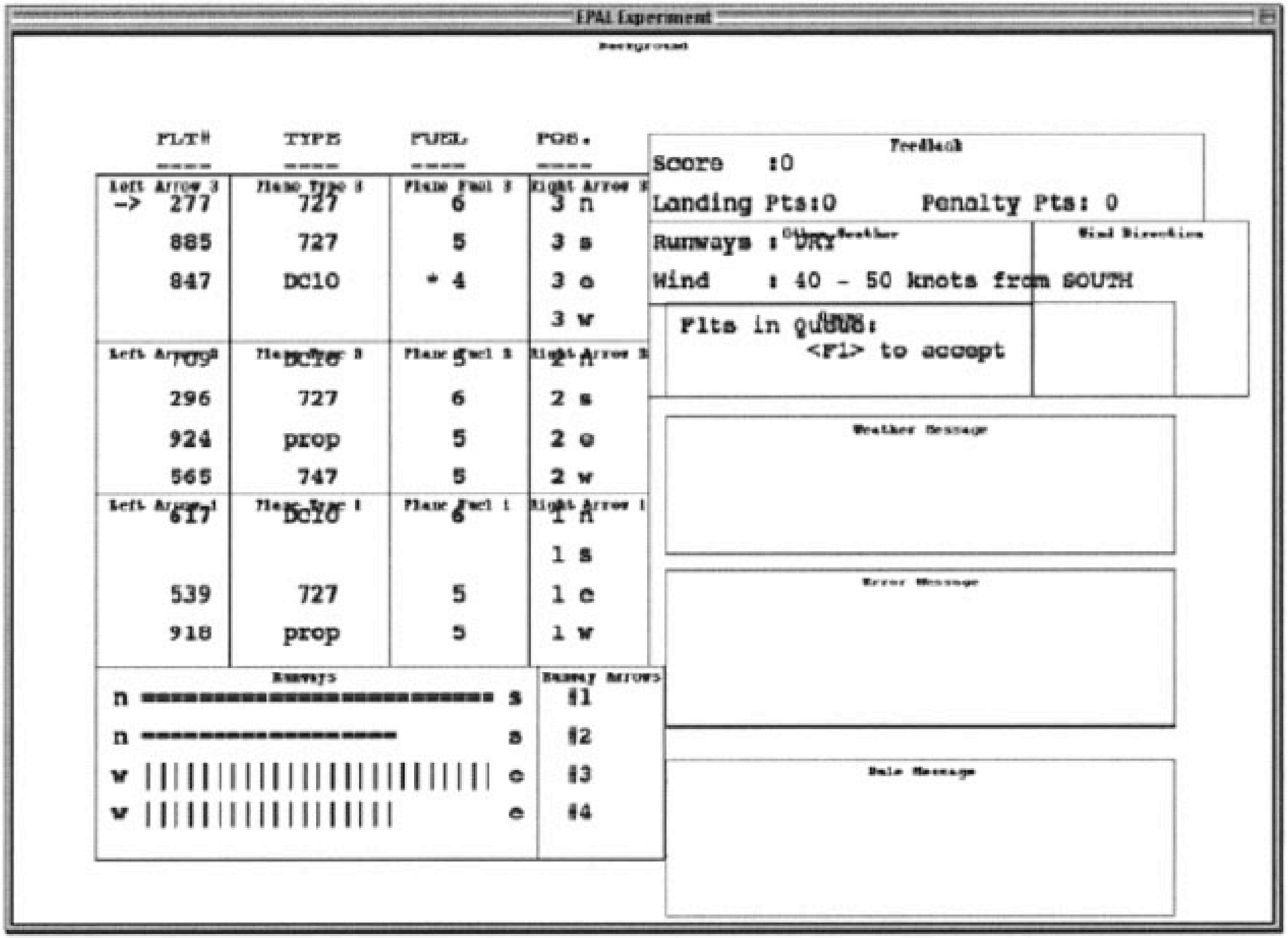
New Chunks by Goal Caching

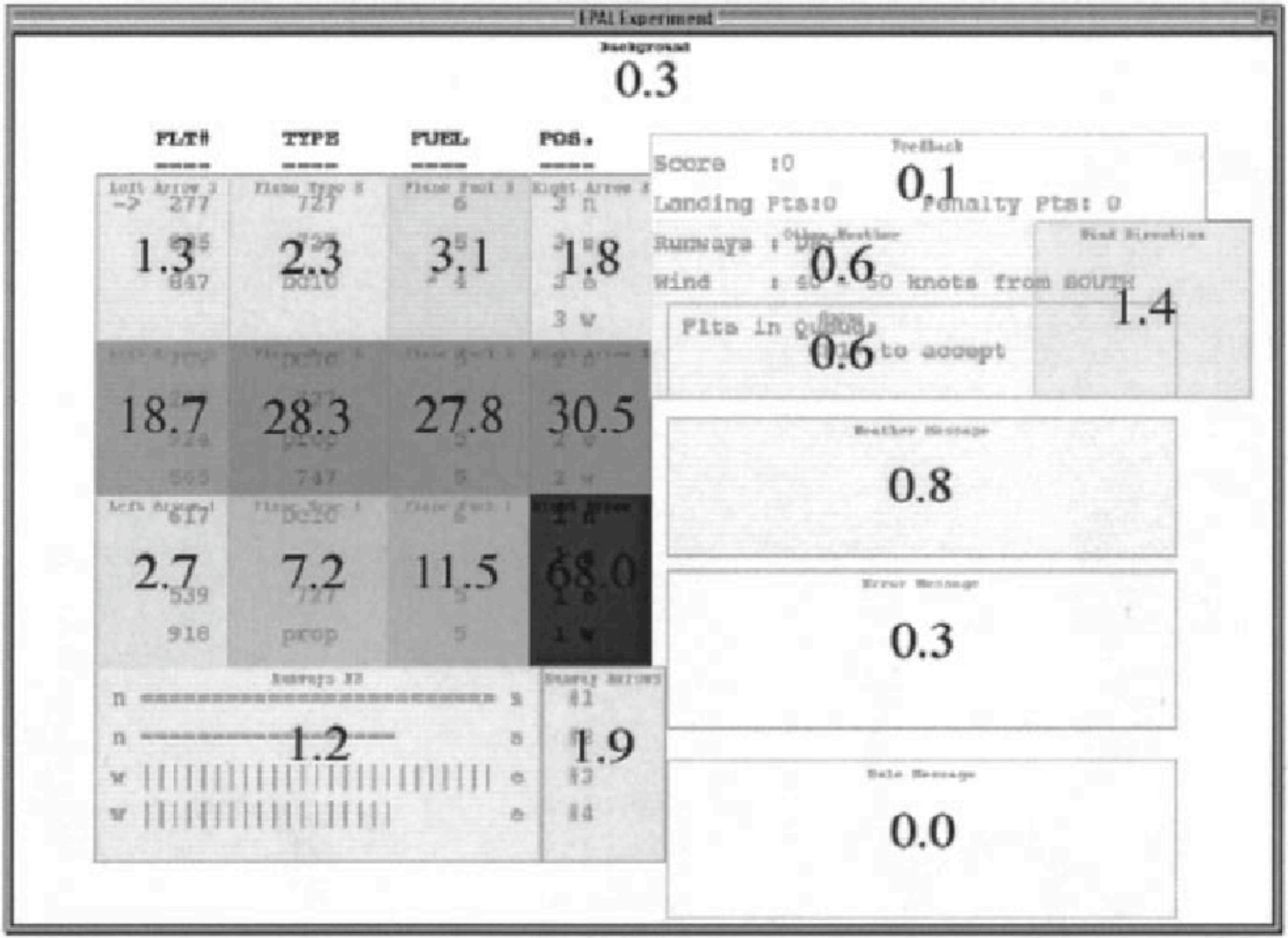
Production Calculate1
IF
    goal=(add,X,Y,__)
THEN
    subgoal=(count,X,Y,Z)
    goal=(add,X,Y,Z)
    write(Z)

Production Retrieve1
IF
    goal=(add,X,Y,__)
THEN
    retrieve(add,X,Y,__)

Production Retrieve2
IF
    goal=(add,X,Y,__)
    retrieved=(add,X,Y,Z)
THEN
    write(Z)

Used in lifetime simulation of arithmetic learning.
Other things to cache: location of visual information, ...


EPAL Experiment
FLT#
TYPE
FUEL
POS.
Feedback
Score :0
Landing Pts:0 Penalty Pts: 0
Runways : DRY
Wind : 40 - 50 knots from SOUTH
Flts in queue:
<F1> to accept
277
727
6
885
727
5
847
DC10
4
296
727
6
924
prop
5
565
747
5
539
727
5
918
prop
5
Runways
Runway Arrows


EPAL Experiment
Background
0.3
FLT#
TYPE
FUEL
POS.
1.3
2.3
3.1
1.8
18.7
28.3
27.8
30.5
2.7
7.2
11.5
68.0
1.2
1.9
Feedback
Score :0
0.1
Landing Pts:0
Penalty Pts: 0
Runways :
0.6
Wind : 50 - 50 knots from SOUTH
1.4
Fits in queue
0.6
to accept
Weather Message
0.8
Error Message
0.3
Info Message
0.0

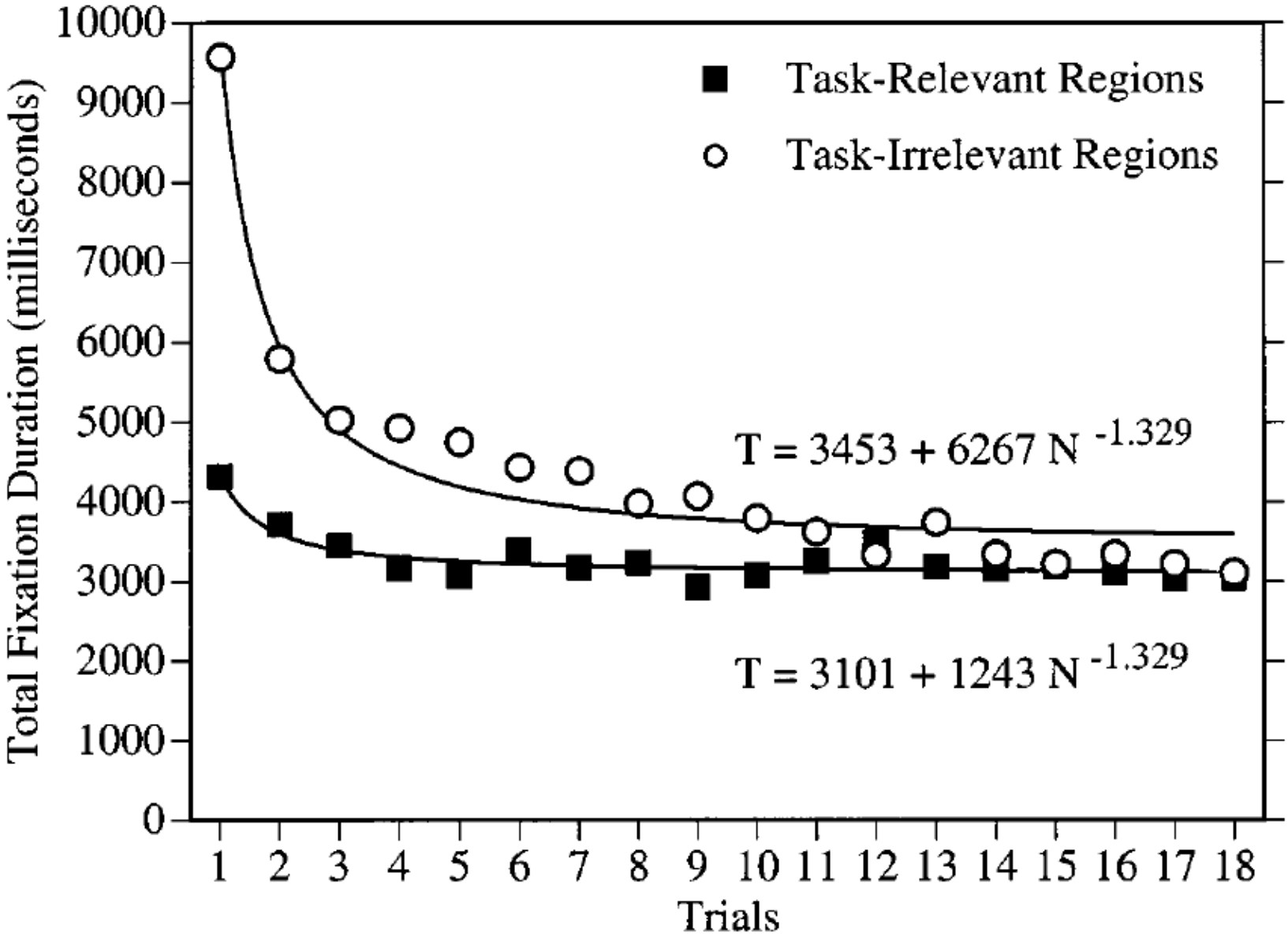


FIG. 19. The mean fixation times in task-relevant and task-irrelevant regions per plane landed in a trial.

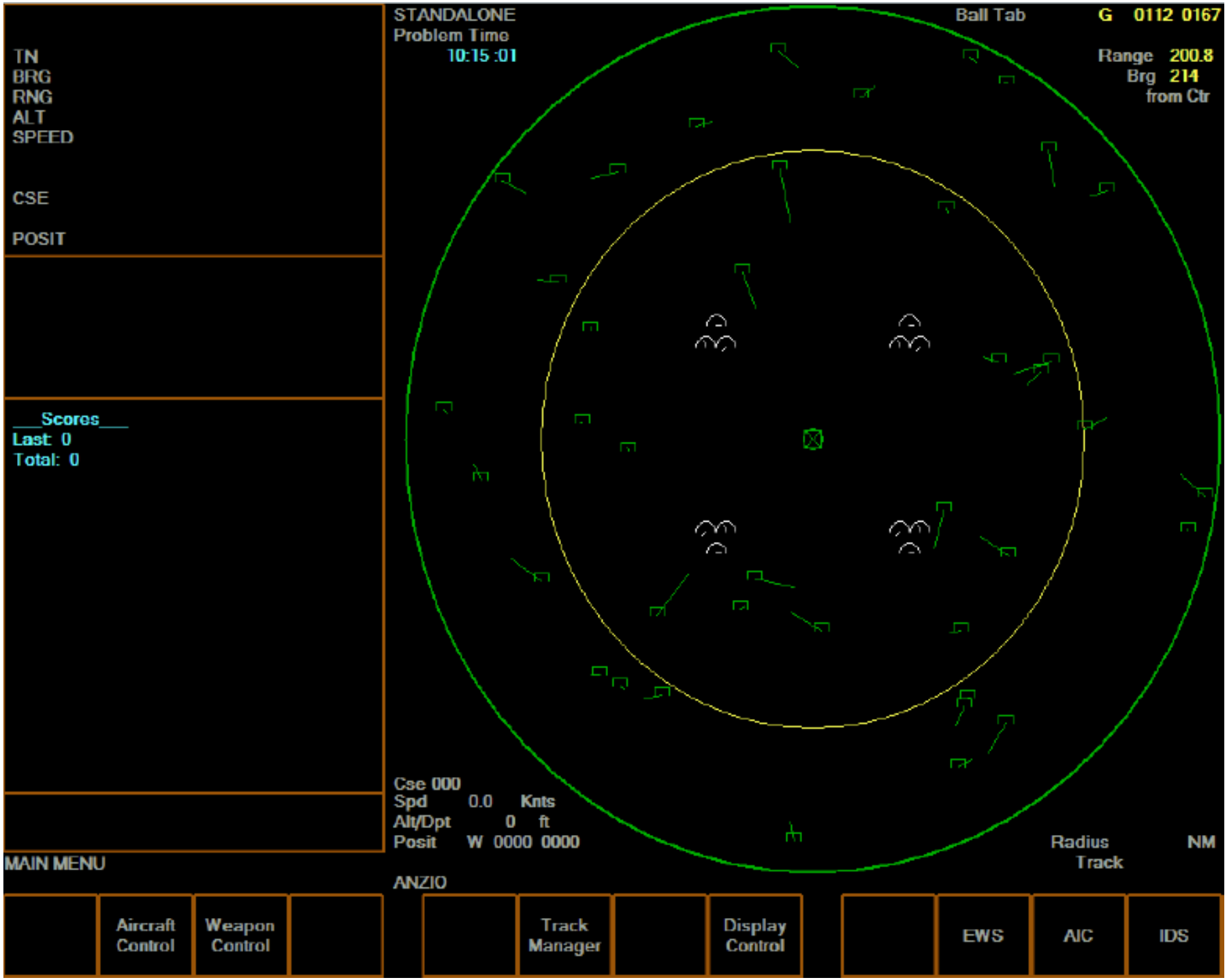
New Productions by Production Compilation

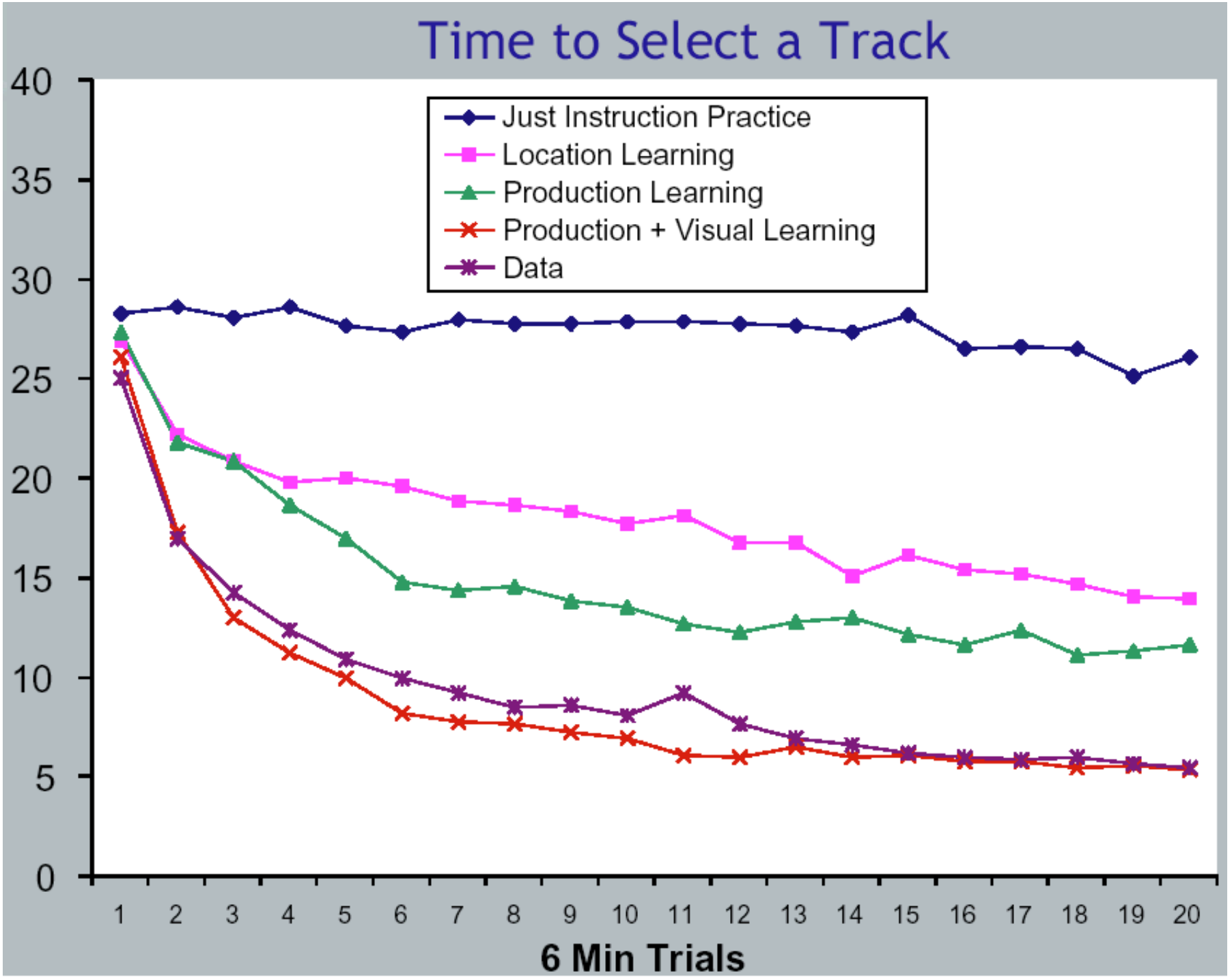
Production Stim
IF
    goal=(press,__,__)
    stimulus = X
THEN
    retrieve map(X,Y)
    goal=(press,X,__)

Production Map
IF
    goal=(press,X,__)
    retrieved map(X,Y)
THEN
    press Y

Production StimMap-uno
IF
    goal=(press,__,__)
    stimulus = “uno”
THEN
    press “1”

Other things to compile: experiment instructions, ...


STANDALONE
Problem Time
10:15 :01
Ball Tab
G 0112 0167
Range 200.8
Brg 214
from Ctr
TN
BRG
RNG
ALT
SPEED
CSE
POSIT
___Scores___
Last: 0
Total: 0
Cse 000
Spd 0.0 Knts
Alt/Dpt 0 ft
Posit W 0000 0000
Radius NM
Track
MAIN MENU
ANZIO
Aircraft Control
Weapon Control
Track Manager
Display Control
EWS
AIC
IDS


Time to Select a Track
Just Instruction Practice
Location Learning
Production Learning
Production + Visual Learning
Data
40
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
6 Min Trials

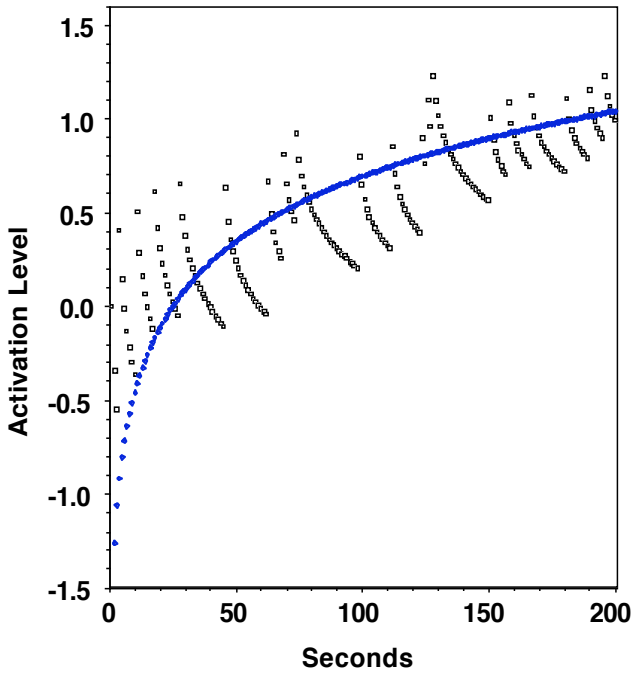
Base-Level Activation Learning

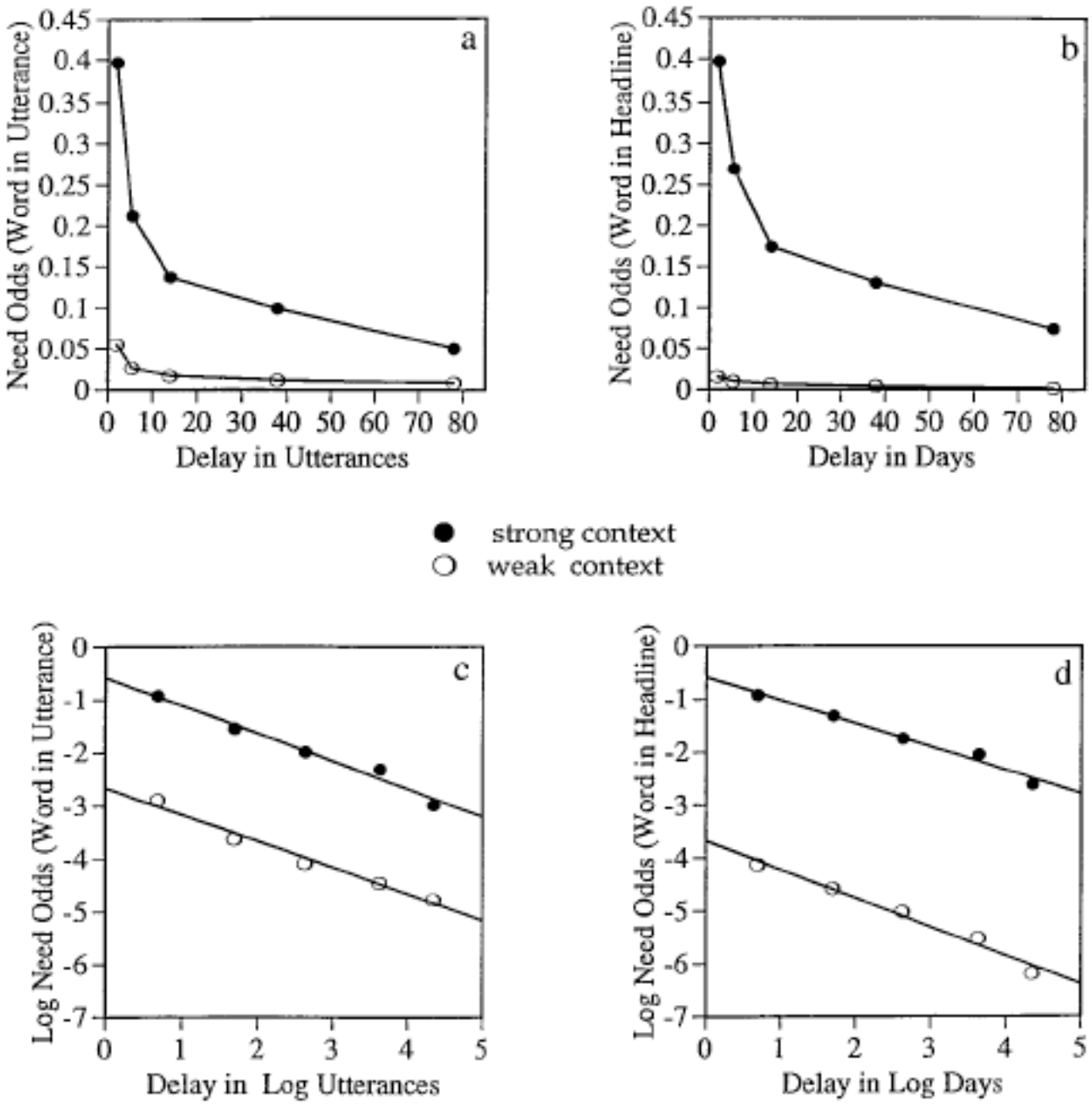
Based on the Rational Analysis of the Environment
(Schooler & Anderson, 1997)

Base-Level Activation reflects the log-odds that a chunk will be needed. In the environment the odds that a fact will be needed decays as a power function of how long it has been since it has been used. The effects of multiple uses sum in determining the odds of being used.

$$B_i = \ln(\sum_{j=1}^{n} t_j^{-d})$$

Base-Level Learning Equation




a
Need Odds (Word in Utterance)
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 10 20 30 40 50 60 70 80
Delay in Utterances
b
Need Odds (Word in Headline)
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 10 20 30 40 50 60 70 80
Delay in Days
strong context
weak context
c
Log Need Odds (Word in Utterance)
0
-1
-2
-3
-4
-5
-6
-7
0 1 2 3 4 5
Delay in Log Utterances
d
Log Need Odds (Word in Headline)
0
-1
-2
-3
-4
-5
-6
-7
0 1 2 3 4 5
Delay in Log Days

Production Utility Learning

Making Choices: Conflict Resolution

Expected Gain = $E = PG - C$

P is expected probability of success
G is value of goal
C is expected cost

Probability of choosing i = $\frac{e^{E_i/t}}{\sum_j e^{E_j/t}}$

t reflects noise in evaluation and is like temperature in the Bolztman equation

$$P = \frac{\text{Successes}}{\text{Successes} + \text{Failures}}$$

Successes = $\alpha + m$
Failures = $\beta + n$

α is prior successes
m is experienced successes
β is prior failures
n is experienced failures

Recent Successes of ACT-R Learning

1. Learning of inflection (English past and German plural).
   Shows that production compilation can come up with generalizations.

2. Learning of air-traffic control task.
   Shows that production compilation can deal with complex perceptual motor skill.

3. Learning of productions for performing paired associate task from instructions.
   Solves mystery of where the productions for doing an experiment come from.

4. Learning to perform an anti-air warfare coordinator task from instructions.
   Shows the same as 2 & 3.

Note all of these examples involve all forms of learning occurring in ACT-R simultaneously – acquiring new chunks, acquiring new productions, activation learning, and utility learning.

Questions?