Context and Learning

In this study, we looked at the effects of context—that is, which stimuli co-occur in the environment—on learning simple, word-like visual stimuli.

We present results from both behavioural studies and computer simulations.

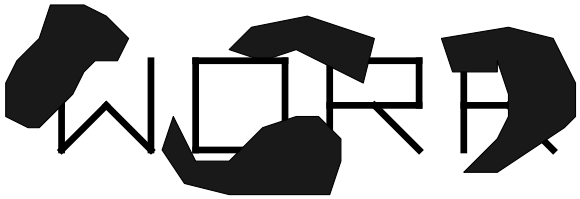
Behavioural results indicate that a "novelty" effect may interact with context effects. One interpretation is that novel items attract attention and hence they "pop-out" at an observer.

Simulations that capture the overall trend of the data suggest that this "novelty" effect has an alternative interpretation. Performance differences are due to the failure to fully represent novel stimuli.

Context and the Environment

The Word Superiority Effect

It is a well established fact that briefly presented, or otherwise obscured, letters are better recognized if they form a word than if they form a pseudo-word, and letters in pseudo-words are better recognized than random letter strings.



One explanation of these results is that the surrounding letters aid in the identification of the unknown letters by adding constraining, or contextual, information.

In these studies, we attempt to address how contextual information interacts with the learning of letter-like characters, both in Bayesian Connectionist Networks and in humans.

Bayesian Approaches to Cognition

Bayesian principles have been regaining popularity within cognitive science, especially in terms of how we learn from the regularities within the environment.

For example, internal representations can be viewed as hypotheses, $\mathcal{H}$, about the data, $\mathcal{D}$, observed from the external world. As such, the problem of defining these internal representations can be reformulated as computing the probability of a given hypothesis (*internal representation*) given the observed data (*external world*)—that is, $P(\mathcal{H}|\mathcal{D})$.

Thus, by specifying $P(\mathcal{D}|\mathcal{H})$, $P(\mathcal{D})$, and $P(\mathcal{H})$, Bayes' theorem provides a mechanism to learn from data.

$$P(\mathcal{H}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{H}) \times P(\mathcal{H})}{P(\mathcal{D})} \tag{1}$$

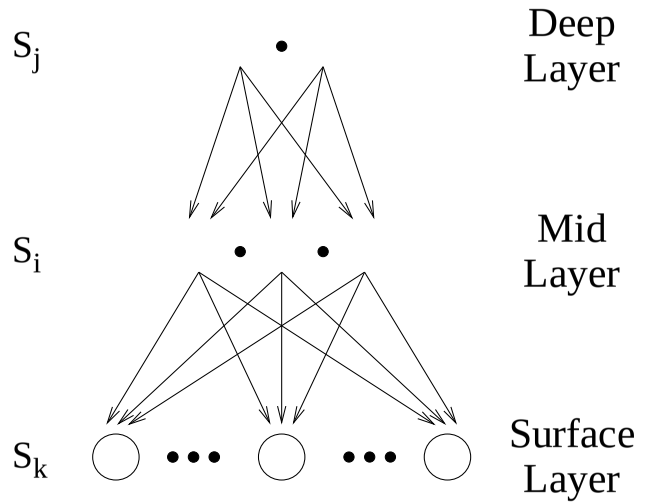
Bayes' Theory and Generative Models

Equation 1 can be rearranged to predict the data given the hypothesis; in other words, this framework can be used to construct a *generative* model, such that the higher-order internal representations predict the lower-level simple features.

In its simplest form, a Bayesian generative network consists of directed connections, where the weight matrix, $\mathbf{W}$, is assumed to encode the probability structure of the network.

Thus, as defined by the connectivity units have

i. parents (**pa**) $S_j > S_i$
ii. children (**ch**) $S_k < S_i$
iii. siblings (**sib**) $S_i = S_i$



Computing Network Probabilities

Being a generative model, the probability of any unit's state is directly computable from the states of its parents:

$$P(S_i = 1 | pa[S_i], \mathbf{W}) = h(\sum_j S_j w_{ij}) \tag{2}$$

where S_j are the parents of S_i and w_{ij} is the weight from unit S_i to S_j.The function h in Equation 2 specifies how these underlying causes are to be combined to produce the probability of $S_i = 1$. One function that can be used for this is the "noisy OR" function:

$$h(u) = 1 - e^{-u} \tag{3}$$

where $u = \sum_j S_j w_{ij}$ is the causal input to S_i. Note that because weights are constrained to be positive, u is never negative, and therefore $0 \leq h(u) \leq 1$.

Sampling Network States

Each state of the network, $\mathcal{S}_m$, is updated iteratively via Gibbs sampling according to the probability of each unit state, S_i, given the states of the remaining units in the network. This conditional probability is computed as

$$P(S_i|S_{j;j\neq i}, \mathbf{W}) \propto P(S_i|pa[S_i], \mathbf{W}) \prod_{j\in ch[S_i]} P(S_j|pa[S_j], S_i, \mathbf{W} \quad (4)$$

Thus, the Gibbs equations as used in this framework can be interpreted in terms of a stochastic recurrent neural network, where the feedback from the deeper layers influences the states at the surface layers.

Hence, the probability of a unit being active given the remaining states of the network is calculated as

$$P(S_i = 1|S_{j;j\neq i}, \mathbf{W}) = \frac{1}{1 + e^{-\Delta x_i}} \quad (5)$$

Weight Estimation

Once we have sampled the activation space, we are in the position to estimate the weights. To control the complexity of the model, a prior is placed on the weights. In using the "noisy OR" function where all weights are constrained to be positive, it is assumed that the weight prior is a product of independent gamma distributions parameterized by α and β. Hence, the objective function we wish the maximize becomes

$$\mathcal{L} = P(D_{1:N}|\mathbf{W})P(\mathbf{W}|\alpha,\beta)$$

Using the maximization step from the EM algorithm, we want to set $\partial\mathcal{L}/w_{ij} = 0$ and solve for w_{ij}. This can be accomplished by using the transformations $f_{ij} = 1 - e^{-w_{ij}}$ and $g_i = 1 - e^{-u_i}$ and solving for

$$f_{ij} = \frac{\alpha - 1 + 2f_{ij} + \sum_n S_i^{(n)} S_j^{(n)} f_{ij}/g_j^{(n)}}{\alpha + \beta + \sum_n S_i^{(n)}} \tag{6}$$

Network Simulations

Networks were trained and tested on the same data sets that participants were given.

For testing purposes, the first pattern was presented to the network, and the network was allowed to settle to a stable state. The internal representation at the deepest layer was then used to generate the probabilities, $g(y_i)$, of a unit being active at the surface layer.

The second pattern was then presented, and the settling/generation process repeated to produce a new set of probabilities, $f(y_i)$.

A variation on the Kullback-Leibler divergence measure was then used to compute whether the network detected a change. The added component is used to ensure that the probabilities sum to 1.

$$KL = \sum_{i=1}^{n} g(y_i) log \frac{g(y_i)}{f(y_i)} + g(1 - y_i) log \frac{g(1 - y_i)}{f(1 - y_i)} \qquad (7)$$

d' were approximated for the networks by simply taking the difference between the *KL* measures for the same (+) and different (-) trials.

The Task: Contextual Learning

The current study was designed to simulate how the word superiority effect may develop. Specifically, we were interested in

i. *the learning of novel, letter-like stimuli*,

ii. *whether stimuli were learned in whole, or in parts*,

iii. *the effects of context on learning*.

To explore these questions, we employed a same/different judgement task between two sequentially presented stimuli.

The Task: Design

Training: Two context groups (A & B) were trained, where each context group consisted of eight stimuli: (3 positions $\times$ 2 characters/position)

Testing: 288 different stimuli were tested:

1. *Familiar Stimuli*: AAA or BBB
2. *Crossed Stimuli*: BAA or ABB
3. *Novel Stimuli*: CAA or DBB

Training and testing trials were intermixed, with no cue as to which was a training trial and which was a testing trial. There were a total of 1440 trials per day (1152 Train, 288 Test). Participants were tested over 10 days.

The Task: Stimuli

Twenty-four letter-like characters were constructed from 16 the line segments of Rumelhart & Siple's (1974) feature based alphabet.

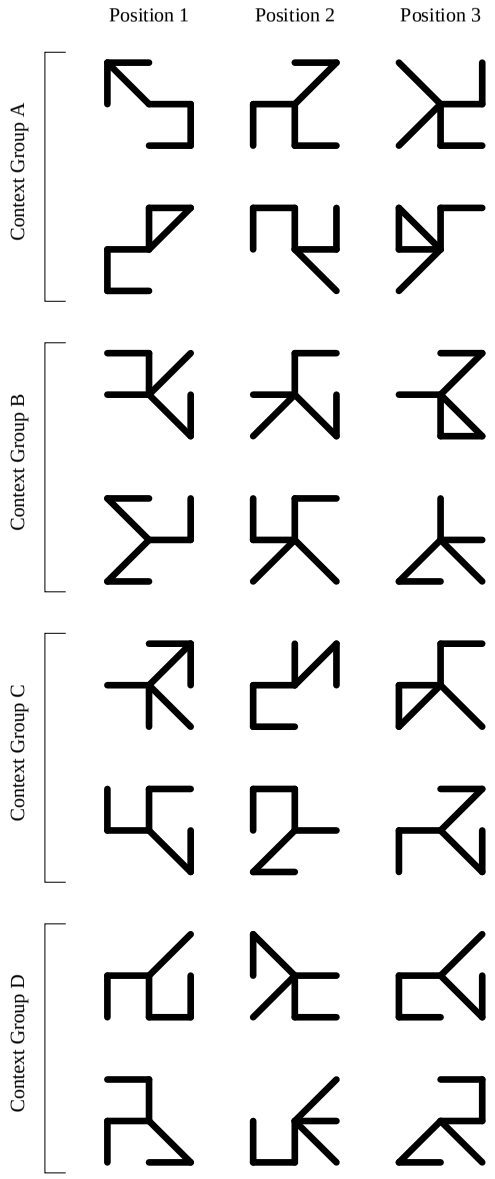
Characters were composed of simple visual features such as horizontal, vertical, and oblique lines.

Each character had six line segments, with the following constraints:

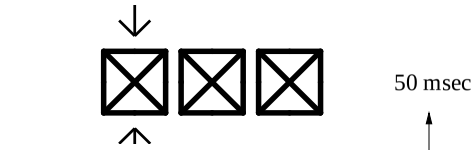
(i) characters were continuous (no stray line segments)

(ii) no two line segments formed a continous straight line

(iii) no character was a mirror image nor rotation of another



The Task: Presentation Method



50 msec

200 msec

250 msec

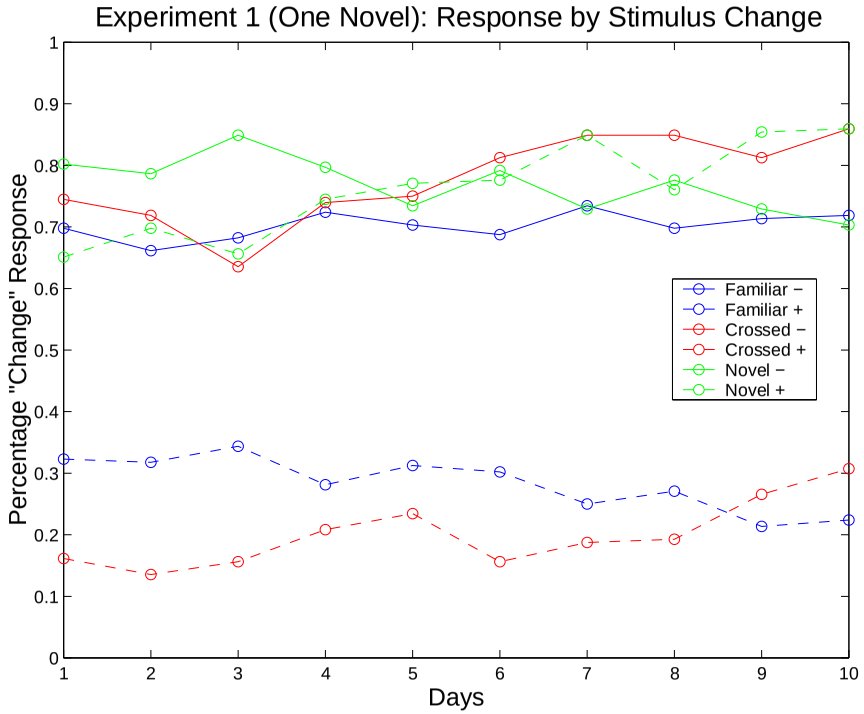
200 msec

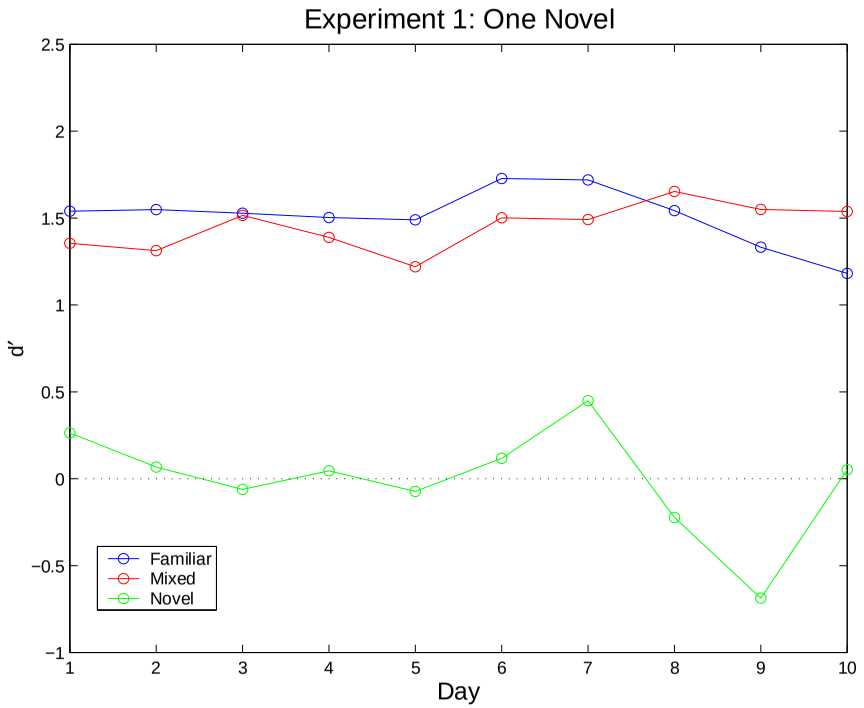
250 msec

500 msec

Experiment 1: Results

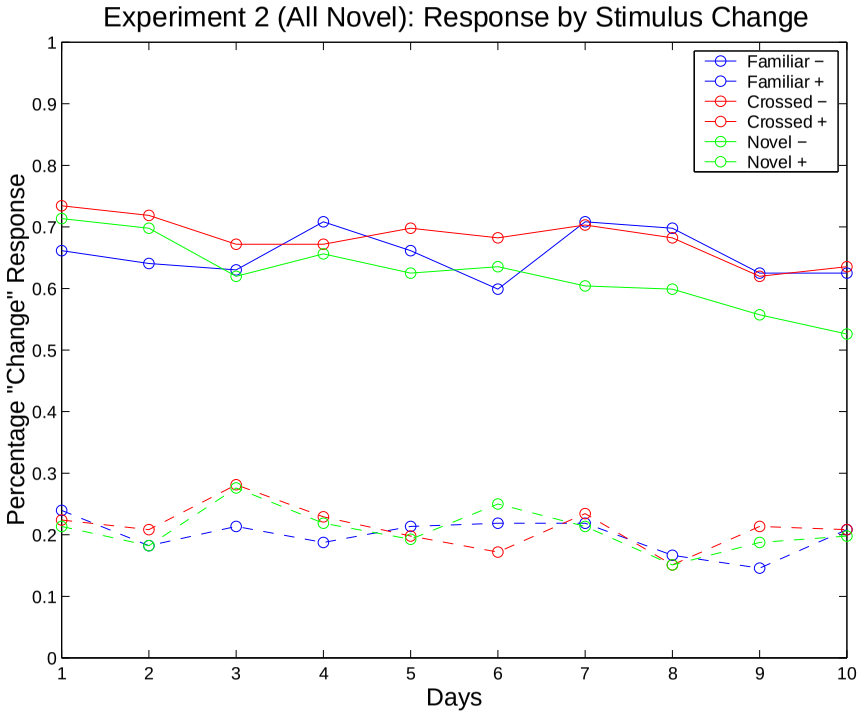
Testing is intermixed with training trials. Novel stimuli contain only one novel character (e.g., *CAA* or *DBB*).

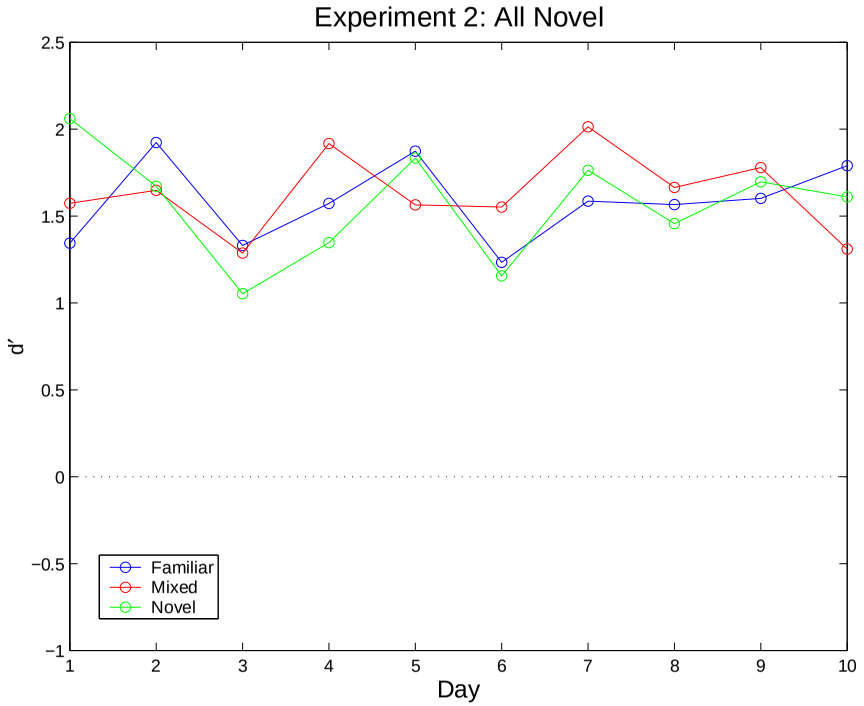




Experiment 2: Results

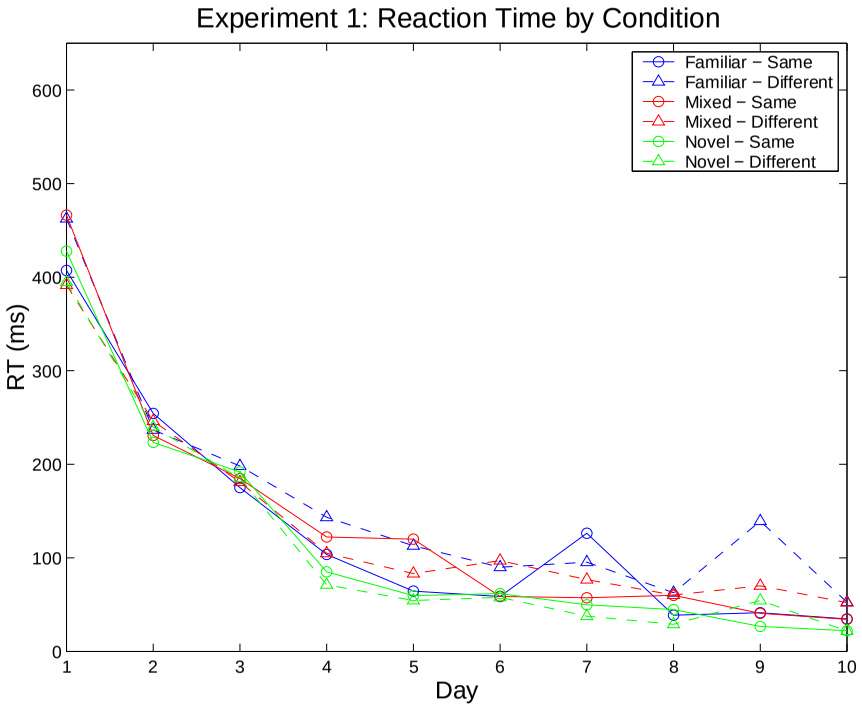
Testing is intermixed with training trials. Novel stimuli contain three novel characters (e.g., *CCC* or *DDD*).

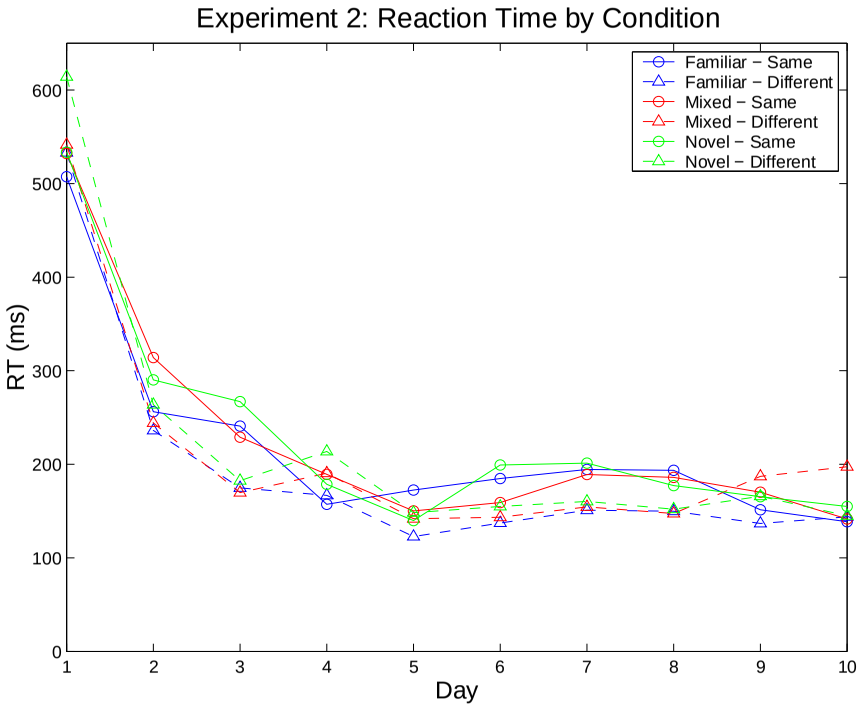




Experiments 1 & 2: Reaction Times

Reaction times were recorded for each of the experiments. Participants in the "one-novel" condition responded significantly faster than participants in the "all-novel" condition.





These discrepancies in RT's suggest that it was more difficult for the participants in Experiment 2 to process the stimuli.

Preliminary Conclusions

- Preliminary results indicate that participants learn context sensitive representations. Participants were more accurate at detecting when characters changed for the familiar stimuli than for either of the other types.

- It would appear that there is a "novelty" effect that interacts with the context effect.

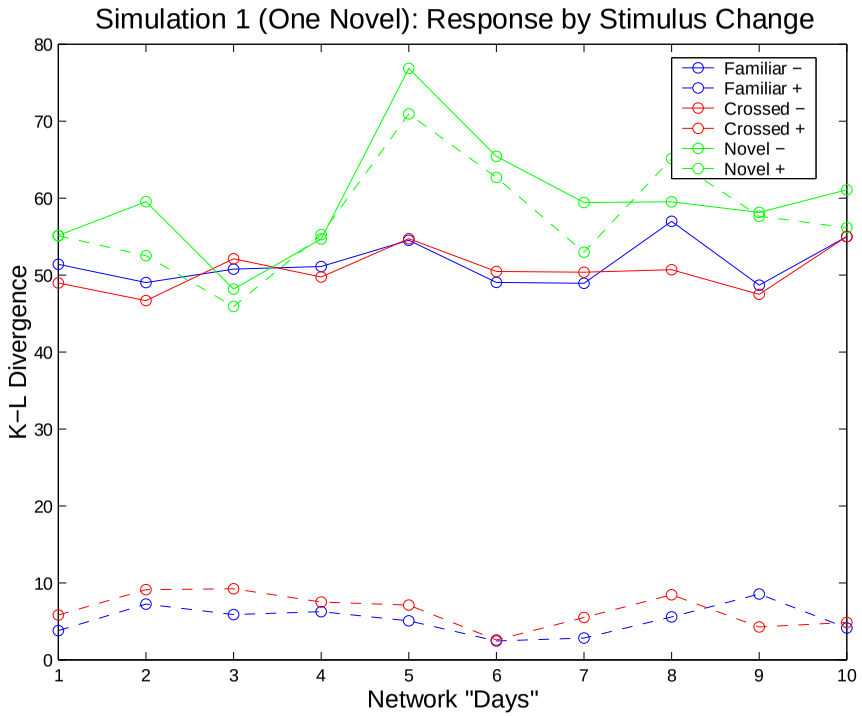
  - When novel stimuli contain only one new character, participants almost always indicate "change" regardless if there was a change or not. Overall processing for all stimuli is relatively rapid.

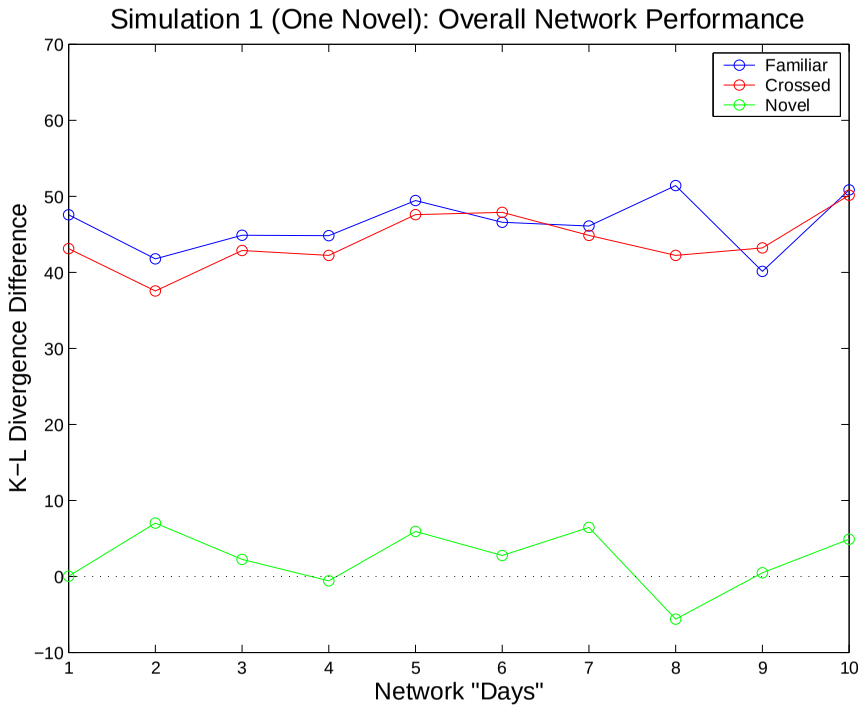
  - When novel stimuli contain all new characters, there appears to be a very limited effect of context. Overall processing for all stimuli, however, is relatively slower.

- One possible explanation for the "novelty" effect would be a shift in attention to the novel stimuli. That is, novel items "pop-out" and are easier to detect.

Simulation 1: Results

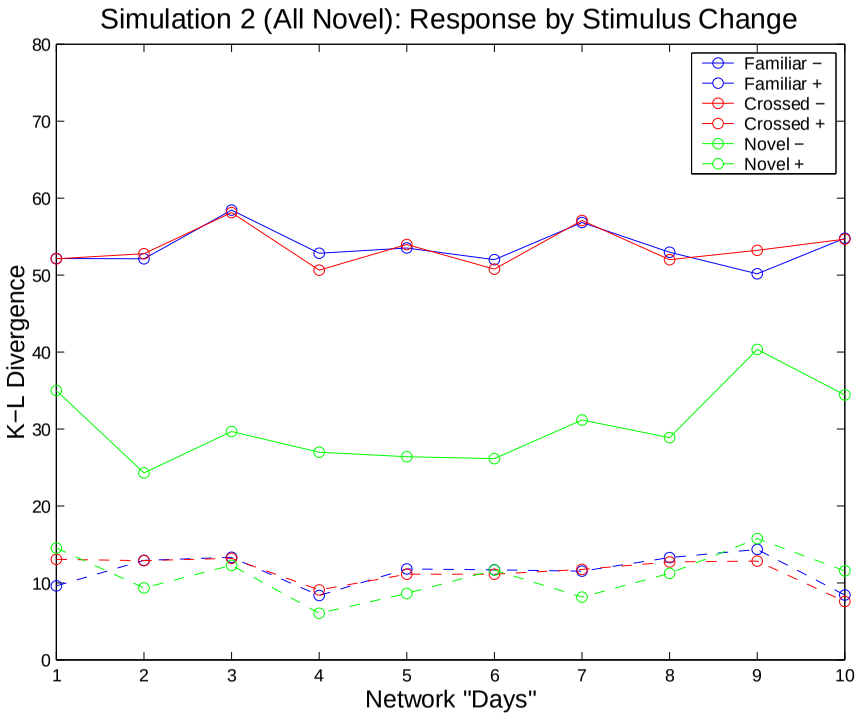
Testing is intermixed with training trials. Novel stimuli contain only one novel character (e.g., *CAA* or *DBB*).

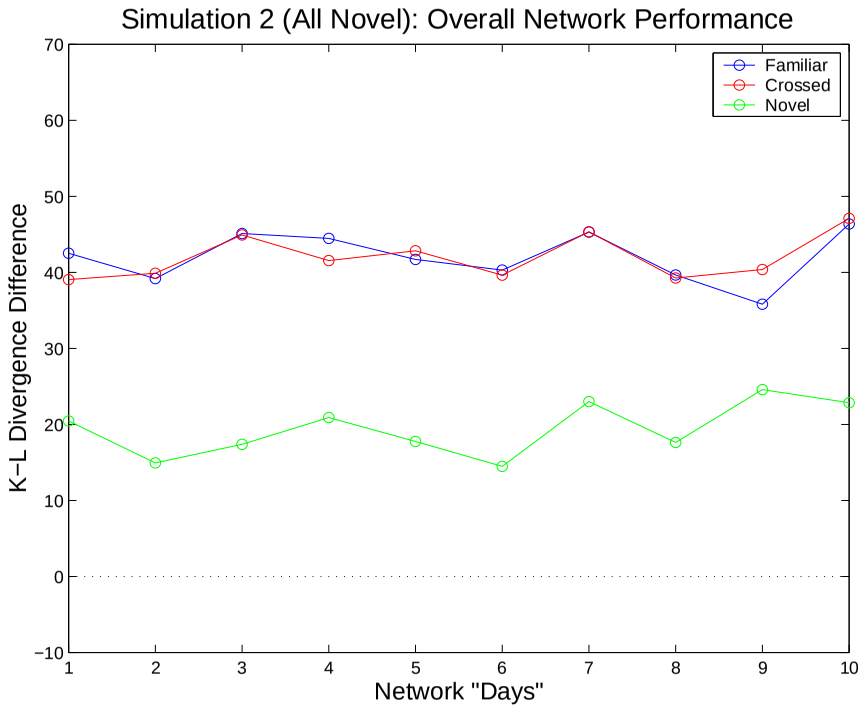




Simulation 2: Results

Testing is intermixed with training trials. Novel stimuli contain three novel characters (e.g., *CCC* or *DDD*).



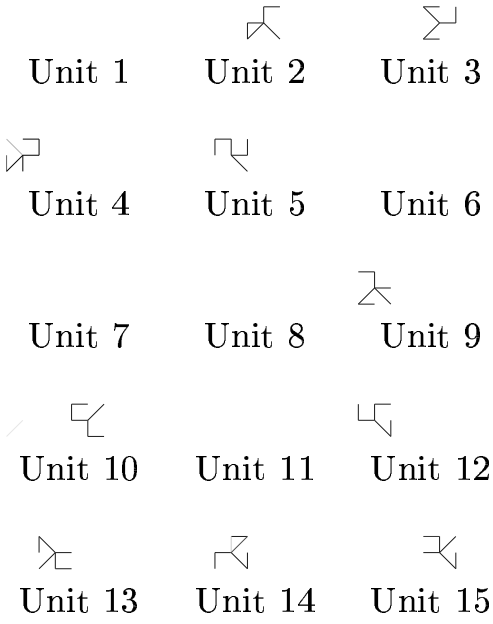


Simulations 1 & 2: Internal Structure

To understand why there is a difference in performance on the novel task between the two networks, we turn to the learned internal representations.

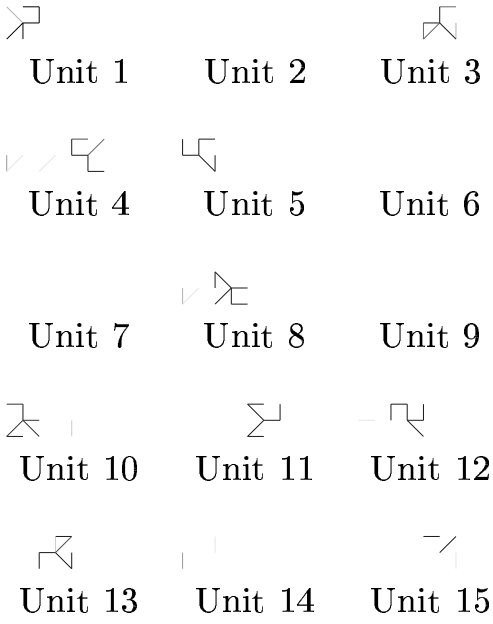
One Novel

Hidden Unit Structure



All Novel

Hidden Unit Structure



The "One Novel" network cleanly learned the characters from the *A* & *B* data sets only; thus, all new stimuli would be processed as changes. While the "All Novel" network developed fuzzier representations, it did learn something about the *C* & *D* data sets; thus, the network could process new characters.

General Conclusions

- Much like participants, Bayesian Connectionist Models learn to be sensitive to environmental context. Results suggest that the network is able to dectect—especially early in training—when context is violated.

- The "novelty" effect may not necessarily be an attentional shift. Simulations indicate that this may be due to failure to learn novel characters.

- This novelty effect is captured in Simulation 1 which qualitatively captures the results of Experiment 1. Analysis of the internal structure of the network shows that it failed to learn the novel characters.

- Results from Simulation 2 are somewhat consistent with Experiment 2. The network is able to detect when there is a change in all novel stimuli. Analysis of the network structure indicates that certain feature of the novel stimuli are learned.

- Importantly, Simulation 2 suggests that participants should detect changes in novel stimuli at a lower rate than either the familiar or crossed stimuli.

- Results from Experiment 2 show that as training progresses, participants are less likely to report "change" for novel stimuli.