A compositional shape code explains how we read jumbled words

Aakash Agrawal[1], K.V.S. Hari[2] & S. P. Arun[3*]

[1]Centre for BioSystems Science & Engineering, [2]Department of Electrical Communication Engineering & [3]Centre for Neuroscience

Indian Institute of Science, Bangalore, 560012, India

*Correspondence to: S. P. Arun (sparun@iisc.ac.in)

ABSTRACT

We raed jubmled wrods effortlessly, yet the visual representations underlying this remarkable ability remain unknown. Here, we show that well-known principles of neural object representations can explain orthographic processing. We constructed a population of neurons whose responses to single letters matched perception, and whose responses to multiple letters was a weighted sum of its responses to single letters. This simple compositional letter code predicted human performance both in visual search as well as on explicit word recognition tasks. Unlike existing models of word recognition, this code is neurally plausible, seamlessly integrates letter shape and position, and does not invoke any specialized detectors for letter combinations. Our results suggest that looking at a word activates a compositional shape code that enables its efficient recognition.

SIGNIFICANCE STATEMENT

Reading is a recent cultural invention, but we are remarkably good at reading words and even jubmeld words. It has so far been unclear whether this ability is due to a representation specialized for letter shapes, or is inherited from basic principles of visual processing. Here we show that a large variety of word recognition phenomena can be explained by well-known principles of object representations, whereby single neurons are selective for the shapes of single letters and respond to longer strings according to a compositional rule.

INTRODUCTION

Reading is a recent cultural invention, yet we are remarkably efficient at reading words and even jmulbed wrods (Fig. 1A). What makes a jumbled word easy or hard to read? This question has captured the popular imagination through demonstrations such as the purported Cambridge University effect (*1*, *2*), depicted in Fig. 1A. It has also been investigated extensively, leading to the identification of a variety of factors (*3*, *4*). The simplest factors are visual or letter-based (Fig. 1B): word reading is easy when similar shapes are substituted (*5*, *6*), when the first and last letters are preserved (*7*), when there are fewer transpositions (*8*) and when word shape is preserved (*3*, *4*). Despite these advances, it is unclear how these factors combine since we do not understand how word representations are related to letters. The more complex factors are lexical and linguistic (Fig. 1B): word recognition is easier for frequent words, and for shuffled words that preserve intermediate units such as consonant clusters and morphemes (*3*, *4*). Yet these manipulations inevitably also affect the letter-based factors, and so whether they have a distinct contribution remains unclear.

Addressing these fundamental questions will require understanding how letter shape and position combine to form word representations. To this end, we performed visual search tasks in which subjects were required to find an oddball target. We chose visual search since it does not require any explicit reading, and because it is closely linked to shape representations in visual cortex (*9*, *10*). An example search array containing two oddball targets is shown in Fig. 1C. It can be seen that finding OFRGET is easy among FORGET whereas finding FOGRET is hard (Fig. 1C). This difference in visual similarity (Fig. 1D) explains why a word with middle letters jumbled are easier to read than a word with the edge letters jumbled.

The above observation suggests that many reading phenomena can be explained using shape representations that drive visual search. Alternatively, even visual search may have been influenced by lexical and linguistic factors. To overcome this confound, we developed a neurally plausible model to predict word discrimination exclusively using visual considerations. We drew upon two well-known principles of object representations in high-level vision. First, images that are perceptually similar elicit similar patterns of activity in single neurons (*9*–*11*). We used this principle to create neural responses to single letters. Second, the neural response to multiple objects is a linear combination of the response to the individual objects, a phenomenon known as divisive normalization (*10*, *12*, *13*). We used this to create responses to longer strings and words from letter responses. Thus, this neural model incorporates only visual aspects of a word (letter shape and position) but not higher order statistical features of language such as the occurrence of bigrams, trigrams or words. It is also devoid of any knowledge of linguistic features of words, such as phonemes, morphemes, words or semantics. The resulting model elucidates the initial visual representation of a word that forms the basis for further linguistic processing.

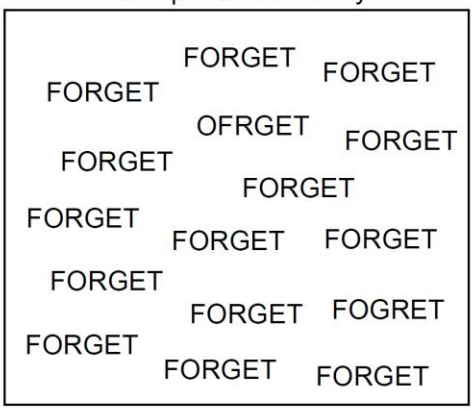
A

AOCCDRNIG TO A RSEEARCH AT CMABRIGDE UINERVTISY, IT DEOSN'T MTTAER IN WAHT OREDR THE LTTEERS IN A WROD ARE, THE OLNY IPRMOETNT TIHNG IS TAHT THE FRIST AND LSAT LTTEER BE AT THE RGHIT PCLAE. THE RSET CAN BE A TOATL MSES AND YOU CAN SITLL RAED IT WOUTHIT A PORBELM. TIHS IS BCUSEAE THE HUAMN MNID DEOS NOT RAED ERVEY LTETER BY ISTLEF, BUT THE WROD AS A WLOHE.

B

	Factors that facilitate word reading	EASY	HARD
Visual	Fewer transpositions	*FGROET*	*FGEORT*
Visual	First letter transposition	*FOGRET*	*OFRGET*
Visual	Preserving Word shape	*froget*	*fogret*
Visual	Similar letter substitution	*FORCET*	*FORXET*
	Familiarity	*TARGET*	*FORGET*
	Linguistic factors	*FGORET*	*FROGET*

C



D

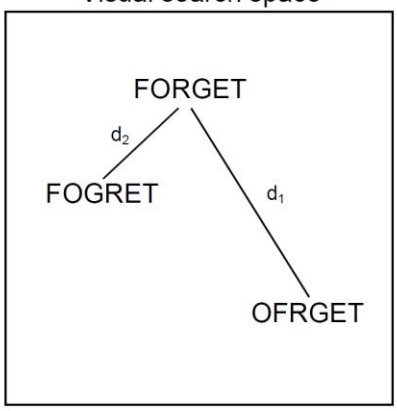


Figure 1. Reading scrambled words

(A) We are extremely good at reading scrambled words, as illustrated by the purported Cambridge University effect where every word is jumbled while leaving the first and last letters intact.

(B) Factors thought to facilitate jumbled word reading.
Fewer transpositions: transposing only two letters (G & O in FORGET) is easy to read whereas many transpositions (G & O, E & R) is hard.
Middle letter transposition: transposing the middle letters (G & R) is easy whereas transposing edge letters (O & F) is hard.
Preserving word shape: a jumbled word such as “froget” is easy because its overall shape envelope matches with “forget”.
Similar letter substitution: – Replacing G in FORGET with a similar letter makes the resulting word easier to read than substituting the dissimilar letter X.

Familiarity: A frequent word like 'TARGET' is easier to read compared to 'FORGET' which is relatively less frequent.
Linguistic factors: A jumbled word like FROGET which includes a new word (FROG) will slow down reading compared to one that doesn't, such as FGORET.

(C) Visual search array showing two oddball targets (OFRGET & FOGRET) among many instances of FORGET. It can be seen that OFRGET is easy to find whereas FOGRET is harder to find.

(D) Schematic representation of these three words in visual search space. The search difficulty suggests that FOGRET is closer to FORGET compared to OFRGET (i.e. $d_1 > d_2$). Thus jumbled word reading might be driven by visual dissimilarity.

RESULTS

We investigated whether visual word representations can be understood using single letter representations. In Experiment 1, we characterized the shape representation of single letters using visual search and demonstrate how search data can be used to construct a population of neurons whose responses predict perception. In Experiment 2, we show how bigram search can be predicted using this neural population together with a simple compositional rule. In Experiment 3, we show that visual search for compound words can be predicted using this neural model. Finally we show that this neural model can account for human performance on jumbled word recognition (Experiment 4) as well as word/nonword discrimination (Experiment 5).

Experiment 1: Single letter searches

We recruited 16 subjects to perform an oddball visual search task involving pairs of English uppercase letters, lowercase letters and numbers. Since there were a total of 62 items, subjects performed all possible pairs of searches (${}^{62}C_2$ = 1,891 searches). An example search is shown in Fig. 2A. Subjects were highly consistent in their responses (split-half correlation between average search times of odd- and even-numbered subjects: $r = 0.87$, $p < 0.00005$). We calculated the reciprocal of search times for each letter pair which is a measure of distance between them (*14*). These letter dissimilarities were significantly correlated with subjective dissimilarity ratings reported previously (Section S1).

Since shape dissimilarity in visual search matches closely with neural similarity in visual cortex (*9*, *10*), we asked whether these letter distances can be used to reconstruct the underlying neural responses to single letters. To do so, we performed a multidimensional scaling (MDS) analysis, which finds the n-dimensional coordinates

of all letters such that their distances match the observed visual search distances. In the resulting plot for 2 dimensions for uppercase letters (Fig. 2B), nearby letters correspond to small distances i.e. long search times. The coordinates of letters along a particular dimension can then be taken as the putative response of a single neuron. For example, the first dimension represents the activity of a neuron that responds strongest to the letter O and weakest to X (Fig. 2C). Likewise the second dimension corresponds to a neuron that responds strongest to L and weakest to E (Fig. 2C). We note that the same set of distances can be obtained from a different set of neural responses: a simple coordinate axis rotation would result in another set of neural responses with an equivalent match to the observed distances. Thus, the estimated activity from MDS represents one possible solution to how neurons should respond to individual letters so as to collectively produce behaviour.

As expected, increasing the number of MDS dimensions led to increased match to the observed letter dissimilarities (Fig. 2D). Taking 10 MDS dimensions, which explain nearly 95% of the variance, we obtained the single letter responses of 10 such artificial neurons. We used these single letter responses to predict their response to longer letter strings in all the experiments. Analogous results for all letters and numbers are shown in Section S1.

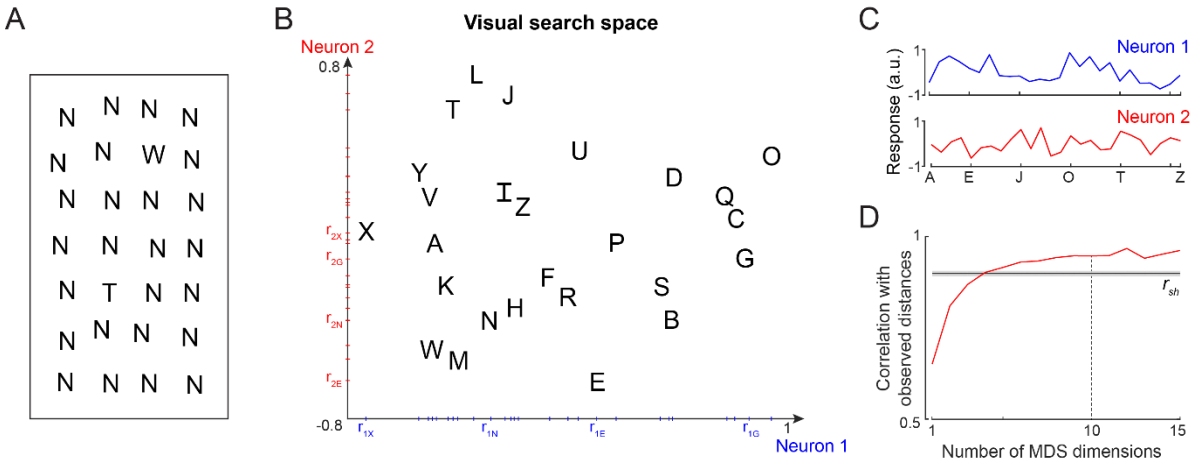


Figure 2. Single letter discrimination (Experiment 1)

(A) Visual search array showing two oddball targets (W & T) among many Ns. It can be seen that finding W is harder compared to finding T. The actual experiment comprised search arrays with only one oddball target among 15 distractors.

(B) Visual search space for uppercase letters obtained by multidimensional scaling of observed dissimilarities. Nearby letters represent hard searches. Distances in this 2D plot are highly correlated with the observed distances ($r = 0.82$, $p < 0.00005$). Letter activations along the x-axis are taken as responses of Neuron 1 (*blue*), and along the y-axis are taken as Neuron 2 (*red*), etc. The tick marks indicate the response of each letter along that neuron.

(C) Responses of Neuron 1 and Neuron 2 shown separately for each letter. Neuron 1 responds best to O, whereas Neuron 2 responds best to L.

(D) Correlation between observed distances and MDS embedding as a function of number of MDS dimensions. The dashed line represents the split-half correlation with error bars representing s.d calculated across 100 random splits.

Experiment 2: Bigram searches

Next we proceeded to ask whether searches for longer strings can be explained using single letter responses. A total of 8 subjects performed an oddball search experiment involving bigrams. An example search is depicted in Fig. 3A. It can be seen that, finding TA among AT is harder than finding UT among AT. Thus, letter transpositions are more similar compared to letter substitutions, in keeping with the classic results on reading (*3*, *4*). We created all possible 49 bigrams from a subset of 7 letters (Fig. 3A): these bigrams included both frequent bigrams (e.g. IN, TH) and infrequent bigrams (e.g. MH, HH). Subjects performed all possible searches involving these bigrams (${}^{49}C_2$ = 1176 searches). As before, subjects were highly consistent in their performance (split-half correlation between odd and even numbered subjects: $r = 0.82$, $p < 0.00005$).

Next we asked whether bigram search performance can be explained using single letter responses estimated from Experiment 1. The essential principle is depicted in Fig. 3B. In monkey visual cortex, the response of single neurons to two simultaneously presented objects is an average of the single object responses (*10*, *12*, *15*). This averaging can easily be biased through changes in divisive normalization (*13*). Therefore we took the response of each neuron to a bigram to be a weighted sum of its responses to the constituent letters (Fig. 3B). Specifically, the response to the bigram AB is given by $r_{AB} = w_1 r_A + w_2 r_B$. Note that if $w_1 = w_2$, the bigram response to AB and BA will be identical. Thus, discriminating letter transpositions requires asymmetric summation. Thus the neural model for bigrams has two unknown spatial weighting parameters for each neuron, and we used a total of 10 neurons throughout, which accounted for 95% of the variance in single letter dissimilarities. Varying this choice yielded qualitatively similar results. We optimized these weights to match the

observed bigram dissimilarities using standard nonlinear fitting algorithms (see Methods).

This neural model yielded excellent fits to the observed data ($r = 0.85$, $p < 0.00005$; Fig. 3C). To assess whether the model explains all the systematic variance in the data, we calculated an upper bound estimated from the inter-subject consistency (see Methods). This consistency measure ($r_{data} = 0.90$) was close to the model fit, suggesting that the model captured nearly all the systematic variance in the data. As predicted in the schematic figure (Fig. 3B), the estimated spatial summation weights were unequal (average absolute difference between w_1 and w_2: 0.07 ± 0.04). To assess whether this difference was statistically significant we randomly shuffled the observed dissimilarities and estimated these weights. The absolute difference between weights for the shuffled data was significantly smaller (average absolute difference: 0.03 ± 0.02; $p < 0.005$, sign-rank test).

If reading expertise leads to the formation of specialized bigram detectors, we predicted that searches involving frequent bigrams (e.g. TH, ND) or two letter words (e.g. AN, AM) would produce larger model errors compared to other bigrams. Contrary to this prediction, we observed no visually obvious difference in model fits for frequent bigram pairs or word-word pairs compared to other bigram pairs (Fig. 3C). To quantify this pattern, we asked whether the model error for each bigram pair, calculated as the absolute difference between observed and predicted dissimilarity, covaried with the average bigram frequency of the two bigrams (for both frequent bigrams and words). This revealed a weak negative correlation whereby frequent bigram pairs showed smaller errors ($r = -0.06$, $p = 0.04$ across 1176 bigram pairs). This is the opposite of what would be expected if there were specialized detectors. To further investigate possible bigram frequency effects, we compared the model error for the 20 bigram

199 pairs with the largest mean bigram frequency with the 20 pairs with the lowest mean
200 bigram frequency. This too revealed no systematic difference (mean ± sd of residual
201 error: 0.10 ± 0.08 for the 20 most frequent bigrams and words; 0.11 ± 0.09 for 20 least
202 frequent bigrams; p = 0.80, rank-sum test). Thus, model errors are not systematically
203 different for frequent compared to infrequent bigram pairs.
204
205

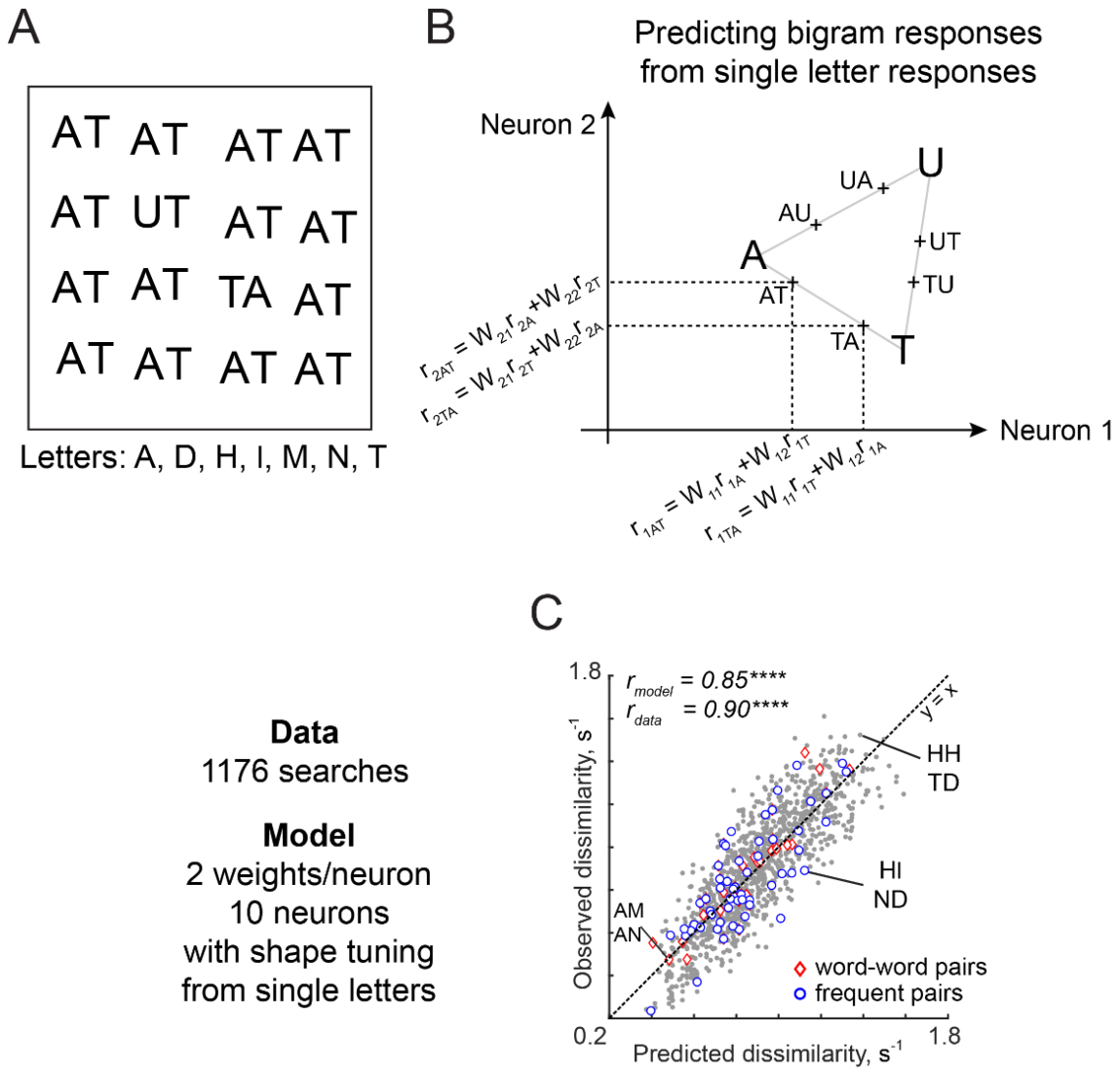


Figure 3. Bigram experiment (Experiment 2)

(A) Example search array with two oddball targets (UT & TA) among the bigram AT. It can be seen that UT is easier to find than TA, showing that letter substitution causes a bigger visual change compared to transposition.

(B) Prediction of bigram responses from single letters: The response of each neuron to a bigram is given by a weighted sum of its response to single letters. The weights are depicted by W with subscripts indicating the neuron and letter location. Note that the bigrams AT and TA can be distinguished only if there is unequal summation. In the schematic, the first position is assumed to have higher magnitude.

(C) In the neural model (*left*), the two weights for each neuron are taken as unknown but the single letter responses are fixed. *Right:* observed dissimilarities between bigram pairs plotted against predictions of the neural model for word-word pairs (*red diamonds*), frequent bigram pairs (*blue circles*) and all other bigram pairs (*gray dots*). Model correlation is shown at the top left, along with the data consistency for comparison.

Generalization to longer strings

To investigate whether these results would generalize to longer strings which can contain frequent words, we performed several additional visual search experiments using 3, 4, 5 and 6-letter uppercase strings. The neural model yielded excellent fits across all string lengths (Section S2). We also tested lowercase and mixed-case strings because word shape is thought to play a role when letters vary in size or have upward and downward deflections (*16*). Even here, the neural model, without any explicit representation of overall word shape, was able to accurately predict most of the search performance (Section S2).

We conclude that dissimilarity between longer strings can be explained using simple spatial summation of single letter responses.

Can letter dissimilarities be estimated directly from bigrams?

The neural model described is neurally plausible and compositional, but is based on dissimilarities between letters presented in isolation. It could be that the representation of a letter within a bigram, although compositional, differs from its representation when seen in isolation. Likewise the representation of the first letter in a bigram, although compositional, might differ from that of the second letter. To explore these possibilities we developed an alternate model in which bigram dissimilarities can be predicted using a sum of (unknown) part dissimilarities at different locations. The resulting model, which we denote as the part sum model yields comparable fits to the data (Section S3). It is completely equivalent to the neural model under certain conditions. Unlike the neural model which is nonlinear and could suffer from multiple local minima, the part sum model is linear and its parameters can be estimated

uniquely using standard linear regression. Its complexity can be drastically reduced using simplifying assumptions without affecting model fits (Section S3).

Effect of familiarity on spatial summation

In the neural model, sensitivity to letter transpositions is increased with asymmetry in spatial summation. We therefore predicted that readers might be more sensitive to letter transpositions due to asymmetric summation. To test this prediction we compared visual search for upright strings with inverted strings, which have identical visual features but differ in their familiarity to the reader. Alternatively, if readers had developed specialized detectors for longer strings, we predicted that model fits would be worse for upright strings compared to inverted strings. We found that the neural model yielded equally good fits for both upright and inverted strings, thereby ruling out the presence of specialized detectors. Further, the estimated spatial weights were more asymmetric for upright compared to inverted strings (Section S4). Thus, the neural model explains how letter familiarity shapes word representations.

Experiment 3: Compound words

Having shown that visual discrimination of longer strings can be explained using single letters, we performed an additional experiment to detect the presence of specialized word detectors. We created compound words by combining two valid words such as FORGET from FOR and GET (Fig. 4A). This resulted in some valid words (e.g. FORGET, TEAPOT) and many invalid words (e.g. FORPOT and TEAGET). The full stimulus set is shown in Section S5.

If valid words are driven by specialized detectors, responses to valid words should be less predictable by the single letter model. We formulated two specific

predictions. First, we predicted that the dissimilarity between valid words (e.g. FORMAT vs TEAPOT) would yield larger model errors compared to invalid word pairs (e.g. DAYFOR vs ANYMAT). Second, we predicted that the dissimilarity between two invalid compound words (e.g. DAYFOR vs ANYMAT) should be explained better by their constituent trigrams (DAY, FOR, ANY, MAT) rather than by their constituent letters (Fig. 4B).

We recruited 8 subjects to perform oddball search involving pairs of trigrams as well as compound words. In all there were 12 three-letter words which resulted in ${}^{12}C_2$ = 66 searches and 36 compound 6-letter strings which resulted in ${}^{36}C_2$ = 630 searches. We also included 12 three-letter nonwords created by transposing each three-letter words, resulting in an additional ${}^{12}C_2$ = 66 searches. An example search involving two 6-letter strings is shown in Fig. 4C. As before, subjects were highly consistent in their responses (split-half correlation between odd and even subjects: $r = 0.54$, $p < 0.00005$ for 3-letter words; $r = 0.46$, $p < 0.00005$ for 3-letter nonwords; $r = 0.65$, $p < 0.00005$ for 6-letter words).

We started by using the single letter model as before to predict compound word responses. We took single neuron responses as before from Experiment 1, and took the response of each neuron to a compound word to be a weighted sum of its responses to the individual letters. Using these compound word responses, we calculated the dissimilarity between pairs of compound words, and used nonlinear fitting to obtain the best model parameters. The single letter model yielded excellent fits to the data ($r = 0.68$, $p < 0.00005$; Fig. 4D). This performance was comparable to the data consistency estimated as before ($r_{data} = 0.72$).

Next we asked whether discrimination between compound words can be explained better as a combination of two valid three-letter words, or as a combination

of all the constituent six letters. To address this question we constructed a new compositional model based on trigrams, and asked if its performance was better than the single letter model (Fig. 4B). The trigram-based neural model used trigram dissimilarity to construct neurons with trigram tuning, and spatial summation over the two trigrams to predict the 6-gram responses. To compare the performance of both models even though they have different numbers of free parameters, we used cross-validation: we fit both models on the even-numbered subjects and tested their performance on odd-numbered subjects. The letter model outperformed the trigram model (Fig. 4E). Because both models were trained on half the subjects and tested on the other half, the upper bound on their performance is simply the split-half correlation between the two halves of the data (denoted by r_{sh}). Indeed the letter model performance was close to this upper bound (r_{sh} = 0.56; Fig. 4E), suggesting that it explained nearly all the explainable variance in the data. Thus, compound word discrimination can be understood from single letters.

We next asked whether the single letter model could explain 3-letter word and nonword dissimilarity. The single letter model again yielded excellent fits to the data that were comparable to the data consistency (Section S5). Thus, compound word responses can be understood in terms of single letters regardless of word status.

Finally, we looked at the spatial summation weights of the single letter neural model for further insights. The spatial summation weights of the first neuron, whose activity itself explains 65% of the variance in letter dissimilarities, showed a U-shaped profile (Section S5). This is a characteristic profile for letter importance observed in reading studies (*17*). Thus, neural responses are dominated by the first and last letters.

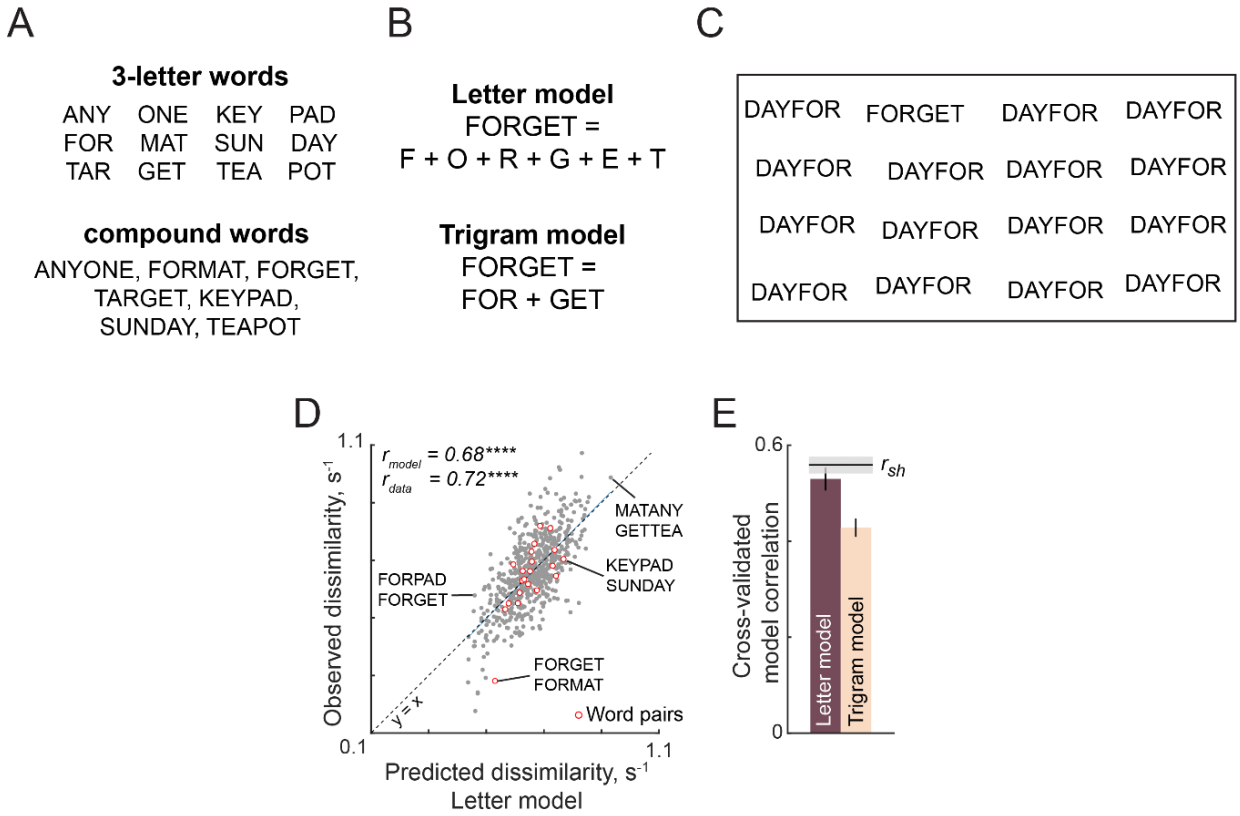


Figure 4. Compound words (Experiment 3)

(A) 3-letter words (*top*) used to create compound words (*bottom*).

(B) Illustration of letter and trigram models. In the letter model, the response to a compound word is a weighted sum of responses to the six single letters. In the trigram model, the response to a compound word is a weighted sum of its two trigrams.

(C) Example search array involving compound words, with one oddball target (FORGET) among identical distractors (DAYFOR).

(D) Observed dissimilarity for compound words plotted against predicted dissimilarity from the letter model for word pairs (*red*) and other pairs (*gray*).

(E) Cross-validated model correlations for the letter and the trigram models. The upper bound on model fits is the split-half correlation (r_{sh}), shown in black with shaded error bars representing standard deviation across 30 random splits.

Experiment 4: Scrambled word reading

The above experiments show that multi-letter string discrimination in visual search can be explained by neurons that embody single letter shape tuning together with a simple compositional rule. Put differently, looking at a word activates a compositional shape representation for the entire word that then drives search. This finding raises the intriguing possibility that the same shape representation might drive reading behaviour. We evaluated this possibility using experiments in which we asked subjects to perform two separate word recognition tasks.

In this experiment, we recruited 16 subjects to perform a scrambled word experiment. On each trial, subjects saw a scrambled word comprising 3, 4 or 5 letters and were asked to press a key as soon as they could unscramble the word. Following this they entered the unscrambled word which we later rated for accuracy. Each scrambled word was presented exactly once to each subject. Of a total of 300 scrambled words tested, we selected for further analysis 238 words that were correctly unscrambled by more than two-thirds of the subjects. Subjects responded quickly and accurately to these words (mean ± std of accuracy: 71 ± 9%; response time: 2.13 ± 0.33 s across 238 words). Subjects took longer to respond to some scrambled words (e.g. REHID) compared to others (e.g. DBTOU), as seen in the sorted response times (Fig. 5A). These patterns were consistent across subjects, as evidenced by a significant split-half correlation ($r = 0.55$, $p < 0.00005$ between odd- and even-numbered subjects).

Can these patterns in unscrambling time be explained using the neural model? To do so, we reasoned that scrambled words with large dissimilarity to the original word will take longer to elicit a response (Fig. 5B). Accordingly we took the average response times to each scrambled word and asked whether it can be predicted using

the single letter model described previously. For each word length, we optimized the weights of the single letter model to find the best fit to this data, and then combined the predictions across all word lengths to obtain a composite measure of performance. The single letter model yielded excellent fits to the data ($r = 0.76$, $p < 0.00005$; Fig. 5C). This model fit was comparable to the data consistency ($r_{data} = 0.70$).

The above finding shows that human performance on unscrambling words is driven primarily by the visual dissimilarity between the scrambled and original word. However it does not rule out the presence of lexical factors. To assess this possibility we formulated a model to predict the unscrambling time as a linear sum of many lexical factors. We used five lexical properties: log word frequency, log mean letter frequency, log mean bigram frequency of the scrambled word, log mean bigram frequency of the unscrambled i.e. original word, and the number of orthographic neighbours (see Methods). To avoid overfitting by either model, we trained both models on one-half of the subjects and tested it on the other half. This lexical model yielded relatively poor fits ($r = 0.30$, $p < 0.00005$, Fig. 5D) compared to visual dissimilarity from the single letter model. The difference in model fit between the lexical model and single letter model was statistically significant ($p < 0.05$, Fisher's z-test). Among the lexical factors, word frequency and letter frequency contributed the most compared to the others (partial correlation of each lexical factor after accounting for all others: $r = -0.23$, $p < 0.0005$ for log word frequency, $r = 0.18$, $p < 0.05$ for log mean letter frequency; $r = .05$, $p = 0.49$ for log mean bigram frequency of scrambled word; $r = -0.02$, $p = 0.77$ for log mean bigram frequency in original word; $r = 0.04$, $p = 0.58$ for number of orthographic neighbours).

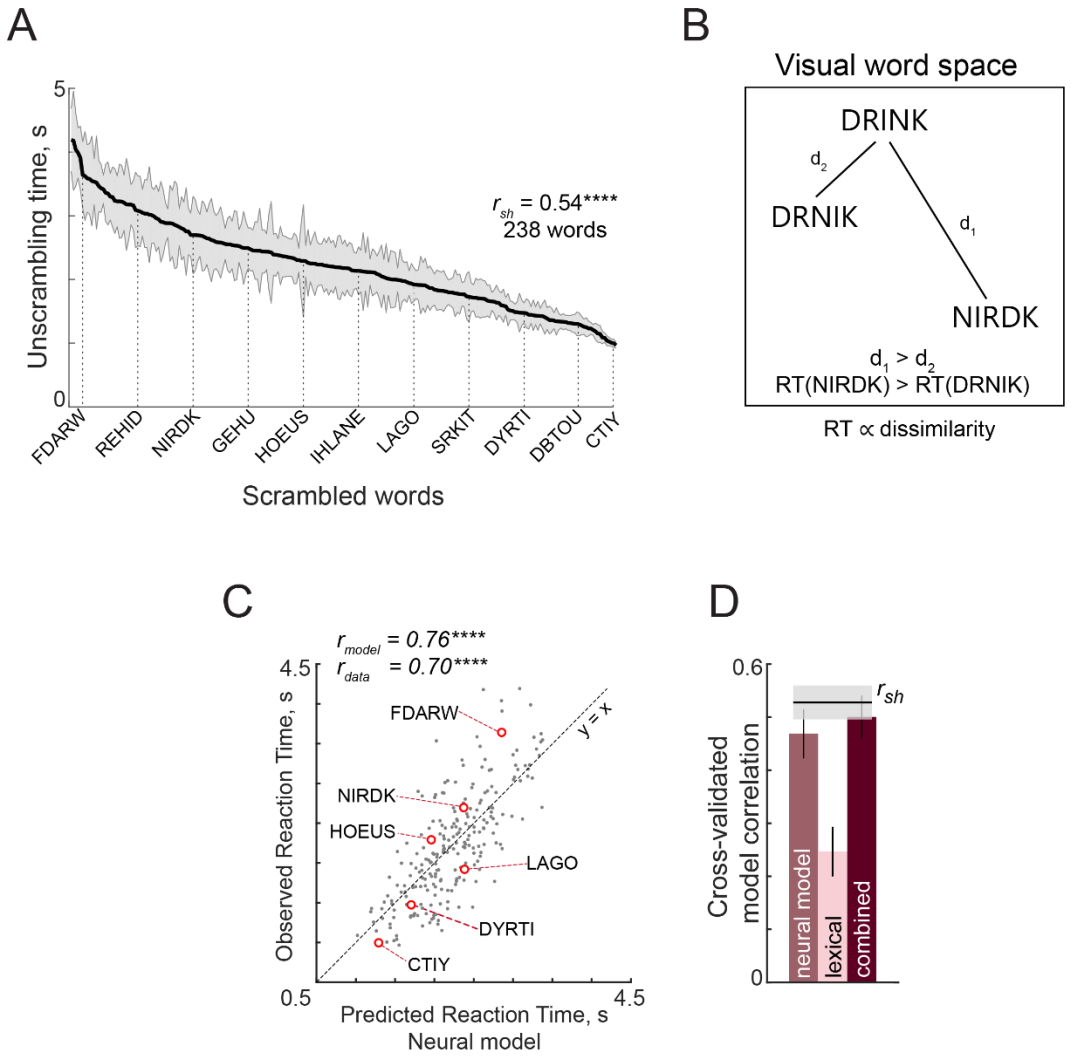


Figure 5. Scrambled word task (Experiment 4)

(A) Response times in the scrambled word task sorted in descending order. Shaded error bars represent s.e.m. Some example words are indicated using dotted lines. The split-half correlation between subjects (r_{sh}) is indicated on the top left.

(B) Schematic of visual word space, with one stored word (DRINK) and two jumbled versions (DRNIK & NIRDK). We predicted that the time taken by subjects to unscramble a jumbled word would be proportional to its dissimilarity to the stored word. Thus, subjects would take longer to unscramble NIRDK compared to DRINK.

(C) Observed response times in the scrambled word task plotted against predictions from the neural model based on single letters with spatial summation. Each point represents one word. Asterisks indicate statistical significance (**** is $p < 0.00005$).

(D) Cross-validated model correlations for the neural model, lexical model and combined model. Model correlations were obtained by training each model on one half of subjects, and evaluating the correlation on the other half (error bars represent standard deviation across 30 random splits). The upper bound on model fits is the split-half correlation (r_{sh}), shown in black with shaded error bars representing standard deviation across the same random splits.

To assess the extent of shared variance in the two models, we calculated the partial correlation between the observed data and the lexical model predictions after factoring out the contribution from visual dissimilarity. This revealed a small partial correlation ($r = 0.31$, $p < 0.00005$). Conversely, the partial correlation for the single letter model after factoring out the lexical model was much higher ($r = 0.75$, $p < 0.00005$). Thus, visual dissimilarity from the single letter model dominates jumbled word reading.

Finally we asked whether both visual dissimilarity and lexical factors contribute to the jumbled word task. We created a combined model in which the scrambled word response times were a linear combination of the predictions of both models. This combined model yielded better predictions than either model by itself ($r = 0.78$, $p < 0.00005$, Fig. 5D). To assess the statistical significance of these results, we performed a bootstrap analysis. On each trial, we trained three models on the dissimilarity obtained from considering only one randomly chosen half of subjects: the visual dissimilarity model, the lexical model and the combined model. We calculated the correlation between all three model predictions on the other half of the data, and repeated this procedure 100 times. Across these samples, the lexical model fits never exceeded the visual dissimilarity model, suggesting that the visual dissimilarity model was significantly better ($p < 0.05$). Likewise the combined model was only marginally better than the visual model (fraction of combined < visual: $p = 0.07$) but was significantly better than the lexical model (fraction of combined < lexical: $p = 0$).

We conclude that performance on the jumbled word task primarily on visual dissimilarity. We propose that this initial visual representation of a word allows the subject to make a quick guess at the correct word without explicit symbolic manipulation.

Experiment 5: Lexical decision task

Here we used a widely used paradigm for word recognition, a lexical decision task, in which subjects have to indicate whether a string of letters is a word (*3*, *4*). We recruited 16 subjects for this task. We used a total of 900 letter strings (450 words, 450 nonwords) made of 4, 5 or 6 letters. Subjects were fast and highly accurate on this task (mean ± std of accuracy: 96 ± 2 % for words, 95 ± 3% for nonwords; response times: 0.58 ± 0.05 s for words, 0.61 ± 0.05 s for nonwords). Importantly their response times were consistent as evidenced by a significant split-half correlation (correlation between odd- and even-numbered subjects: $r = 0.59$, $p < 0.00005$ for words, $r = 0.73$, $p < 0.00005$ for nonwords). Subjects responded faster to some words compared to others (Fig. 6A). Likewise, they responded faster to some nonwords compared to others (Fig. 6B).

Responses in lexical decision tasks are typically thought to depend on accumulation of evidence towards or against word status (*18*, *19*). We reasoned that looking at a string of letters will trigger a compositional neural representation that activates nearby stored patterns that correspond to words. If the string is a word, the response time will depend on the strength of the stored pattern, which in turn would depend on lexical factors such as word frequency (*18*, *19*). This was indeed the case (Section S6). However, if the string is a nonword, the response will be slow if there is a nearby stored pattern corresponding to a word, and fast otherwise (*20*, *21*). Thus, nonword responses may depend on the visual dissimilarity to the nearest word. Specifically, we reasoned that response time for nonwords should be inversely proportional to the dissimilarity between the nonword and the nearest word (Fig. 6C), and also inversely proportional to the frequency of the nearest word.

To test this prediction we took the neural model with 10 neurons with single letter tuning and optimized the spatial summation weights to match the reciprocal of the nonword responses for each word length. To avoid overfitting, we calculated the cross-validated model performance by training the model on one-half of the subjects and testing it on the other half of the subjects. This model yielded excellent fits to the data (mean correlation between observed and predicted 1/RT: r = 0.57, p < 0.00005; Fig. 6D), which was close to the upper bound given by the split-half correlation (r_{sh} = 0.70; Fig. 6D).

To assess the contribution of lexical factors to the nonword responses, we performed a linear regression of the nonword reciprocal RTs against a number of lexical factors. This lexical model yielded relatively poorer fits to the data (r = 0.35, p < 0.00005; Fig. 6D). The difference in model fit between the lexical model and single letter model was statistically significant (p < 0.005, Fisher's z-test). To further establish that the superior fit of the neural model was not simply due to having more free parameters, we compare the lexical model fits with a reduced version of the neural model with only 5 free parameters (Section S3). Even this reduced model showed comparable fits to the neural model that were better than the lexical model (r = 0.49, p < 0.00005; Section S3).

Among the lexical factors, word frequency was the single largest contributor towards both word and nonword responses (Section S6). To assess the degree of shared variance between the lexical and visual models, we performed a partial correlation analysis. The lexical model contributed distinctly to the observed responses even after factoring out the contribution from visual dissimilarity (partial correlation: r = 0.35, p < 0.00005). However the visual dissimilarity from the single letter model had a larger contribution after factoring out the lexical model predictions (partial correlation:

480 r = 0.62, p < 0.00005). We conclude that visual dissimilarity is the dominant driver of
481 the nonword responses.

482 Next we asked whether the response times for nonwords could be entirely
483 explained using a combined model that included both model predictions. This
484 combined model indeed yielded the best prediction (Fig. 6D). The combined model
485 performance approached the theoretical upper bound, given by the split-half
486 consistency of the data (r_{sh} = 0.70; Fig. 6D). To assess statistical significance, we
487 performed a bootstrap analysis as before. The combined model performance was
488 significantly better than both the visual dissimilarity model (fraction of combined <
489 visual: p = 0), and the lexical model (fraction of combined < lexical: p = 0). The visual
490 model was consistently better than the lexical model (fraction of visual < lexical: p =
491 0).

492 We conclude that word response during lexical decisions is driven by lexical
493 factors but nonword responses are strongly influenced by visual factors as well.

494

495 **Can the compositional neural code explain orthographic processing?**

496 Finally, we asked whether the compositional neural model can predict classic
497 phenomena in orthographic processing. In the lexical decision task, subjects took
498 longer to respond to nonwords obtained by transposing the letter of a word, compared
499 to nonwords obtained by substituting a letter (Fig. 6E). Similarly, subjects took longer
500 when the middle letters were transposed compared to when the edge letters were
501 transposed (Fig. 6E). These effects replicate the classic orthographic processing
502 effects reported across many studies (*3*, *4*, *22*, *23*). Importantly, the neural model
503 predictions showed exactly the same trends (Fig. 6E).

504

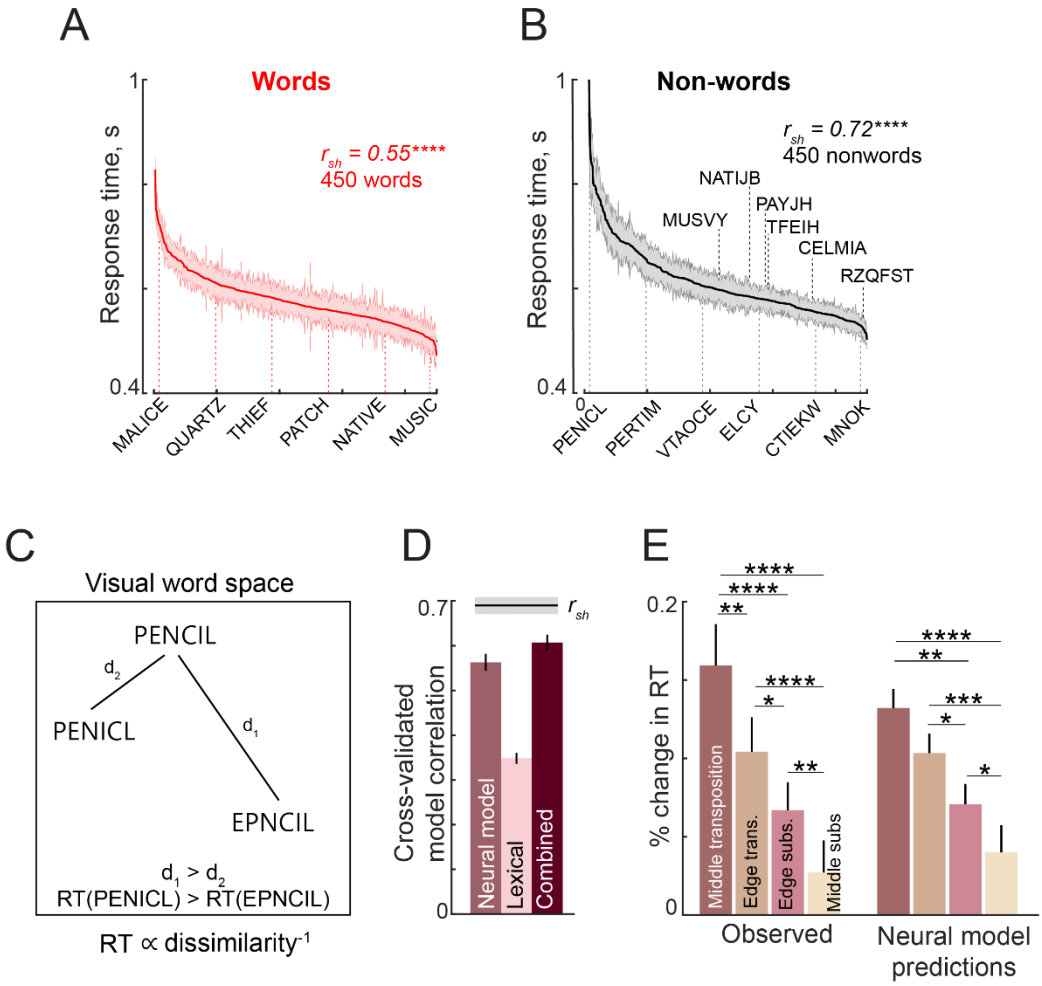


Figure 6. Lexical decision task (Experiment 5)

(A) Response times for words in the lexical decision task, sorted in descending order. The solid line represents the mean categorization time for words and the shaded bars represent s.e.m. Some example words are indicated using dotted lines. The split-half correlation between subjects (r_{sh}) is indicated on the top left.

(B) Same as (A) but for nonwords used in the task.

(C) Schematic of visual word space, with one stored word (PENCIL) and two nonwords (PENICL & EPNCIL). We predicted that subjects would take longer to categorize a nonword when it is similar to a word. Thus, they would take longer to respond to PENICL compared to EPNCIL.

(D) Cross-validated model correlation for the neural model based on single letters, lexical model and combined models. Model correlations were calculated by training each model on one half of the subjects, and evaluating the correlation on the other half. Error bars represent standard deviations across 30 random splits. The upper bound on model fits is the split half correlation (r_{sh}), shown in black with shaded error bars representing standard deviation across the same random splits.

(E) Change in response time (nonword RT – word RT)/word RT for letter transpositions and substitutions for observed responses (*left*) and for neural model predictions (*right*). For the observed data, asterisks represent statistical significance of the main effect of condition in an ANOVA with subject and condition as factors. For the predicted data, the asterisks represent statistical significance using a rank-sum test on the two conditions. In both cases, * is $p < 0.05$, ** is $p < 0.005$ etc.

DISCUSSION

We have shown that our remarkable ability to read jumbled words can be understood using a simple compositional shape code. This code consists of single neurons with fixed shape tuning for single letters, together with a compositional rule whereby the response to longer strings is a linear sum of single letter responses. This code accurately explained human performance on both visual search and word recognition tasks. Below we discuss its implications and its relation to the existing literature.

This code is based on two well-known principles of object representations in the visual cortex. The first principle is that images that elicit similar activity across neurons in high-level visual cortex will appear perceptually similar (*9–11*). This is non-trivial because it is not necessarily true in lower visual areas or in image pixels (*24*). We have turned this principle around to construct artificial neurons whose shape tuning matches visual search. The second principle is that the neural response to multiple objects is typically the average of the individual object responses (*12*, *25*) that can be biased towards a weighted sum (*13*, *26*). Thus both guiding principles of the neural model are strongly grounded in empirical evidence from the visual system.

How does reading expertise affect visual processing?

The success of this letter-based compositional code challenges the widely held belief that reading expertise should lead to the formation of specialized bigram detectors (*4*, *27*, *28*). The presence of these specialized detectors should have caused larger model errors for valid words and frequent n-grams, but we observed no such trend (Fig. 3, 4). So what happens to visual letter representations upon expertise with reading? Our comparison of upright and inverted bigrams suggests that reading

554 should increase letter discrimination and increase the asymmetry of spatial summation
555 (Section S4). This is consistent with differences in letter position effects for symbols
556 and letters (*17*, *29*). We propose that both processes may be driven by visual
557 exposure: repeated viewing of letters makes them more discriminable (*30*), while
558 viewing letter combinations induces asymmetric spatial weighting. Whether these
559 effects require active discrimination such as letter-sound association training or can
560 be induced even by passive viewing will require comparing letter string discrimination
561 under these paradigms.
562
563 **Can compositional shape coding explain orthographic processing?**
564 This neural code can explain many orthographic processing phenomena
565 reported in the literature. Consider the myriad factors thought to influence reading (Fig.
566 7A – same as Fig. 1B). To elucidate how various scrambled versions of a word are
567 represented according to this neural code, we calculated responses of the neural
568 model trained on data from Experiment 3, and visualized the distances using
569 multidimensional scaling (Fig. 7B). It can be seen transposing the edge letters
570 (OFRGET) results in a bigger change than transposing the middle letters (FOGRET).
571 Likewise, it can be seen that substituting a dissimilar letter (FORXET) leads to a large
572 change compared to substituting a similar letter (FORCET), thus explaining many
573 transposed letter effects (*3*). Replacing G with C in FORGET leads to a smaller change
574 than replacing with X, thus explaining how priming is stronger when similar letters are
575 substituted (*31*). Finally, the letter subset FRGT is closer to FORGET than the same
576 letters reversed (TGRF), thereby explaining subset priming (*2*, *27*). Finally, as a
577 powerful demonstration of this code, we used it to arbitrarily manipulate reading
578 difficulty along a sentence (Fig. 7C), or across multiple transpositions and even

579 number substitutions (Fig. 7D). We propose that this compositional neural code can
580 serve as a powerful baseline for the purely visual shape-based representation
581 triggered by viewing words, thereby enabling the study of higher order linguistic
582 influences on reading processes.
583
584 **Relation to other models of word recognition**
585 Our compositional neural code stands in stark contrast to existing models of
586 reading. Existing models of reading assume explicit encoding of letter position and do
587 not account for letter shape (*3*, *8*, *32*, *33*). By contrast, our model encodes letter shape
588 explicitly and position implicitly through asymmetric spatial summation. Our model can
589 be applied to any language by incorporating the corresponding letter dissimilarities.
590 The implicit coding of letter position avoids the complication of counting transpositions
591 (*20*, *34*), while explaining a variety of letter transposition effects (Fig. 7B). The
592 asymmetric spatial weighting shows a larger weight for the first letter (Section S6),
593 which explains the first-letter advantage observed previously (*17*). It also explains why
594 increasing letter spacing can benefit reading in poor readers, presumably because it
595 increases asymmetry in spatial summation (*35*). The integrated representation of both
596 letter shape and position explains both letter transposition and substitution effects and
597 their relative importance (Fig. 7D).
598 Our results offer additional insights into how letter-based visual representations
599 and lexical factors combine during word recognition. In both our scrambled word and
600 lexical decision tasks, visual dissimilarity between a non-word and its nearest word
601 explained a large fraction of the response time variance, and the remaining variance
602 was explained by lexical factors (primarily word frequency). This finding is consistent
603 with a spreading activation account whereby looking at a string of letters activates a

compositional visual representation that is then matched with stored word patterns. Lexical factors contribute here because they modulate the strength of the stored pattern. By contrast, word responses in the lexical decision task are driven only by lexical factors (Section S6), presumably because the response depends only on the strength of the stored pattern since the visual match is almost instantaneous. Our finding that word frequency is a major driver of lexical decision times are consistent with previous work (*19–21*). Our finding that visual dissimilarity influences non-word response times is consistent with the fact that they are influenced by the number of orthographic neighbours (*20*). However our findings demonstrate that visual dissimilarity is the predominant influence on common reading tasks. We propose that the compositional shape code provides a quick match to unscramble a word, failing which subjects may initiate more detailed symbolic manipulation.

Relation between word recognition and reading

We have shown that word recognition can be explained using a compositional visual code based on single letters. While this is an important first step, reading often involves sampling many words with each eye movement (*36*). Our ability to sample multiple letters or words at a single glance is limited by two factors. The first is our visual acuity, which reduces with eccentricity. The second is crowding, by which letters become unrecognizable when flanked by other letters – this effect increases with eccentricity (*37*).

The visual search experiments in our study involved searching for an oddball target (consisting of multiple letters) among multiple distractors. This would most certainly have involved detecting and making saccades to peripheral targets, although we did not monitor eye movements in our study. By contrast, the word recognition

tasks in our study involved subjects looking at words presented at the fovea. Our finding that visual search dissimilarity explains word recognition then indicates that shape representations are qualitatively similar in the fovea and periphery. Furthermore, the structure of the neural model suggests a possible mechanistic explanation for crowding. Neural responses might show greater sensitivity to spatial location at the fovea compared to the periphery, leading to more discriminable representations of multiple letters. Alternatively, neural responses to multiple letters might be more predictable from single letters at the fovea but not in the periphery. Both possibilities would predict reduced recognition with closely spaced flankers. Distinguishing these possibilities will require testing neural responses in higher visual areas to single letters and multi-letter strings of both familiar and unfamiliar scripts.

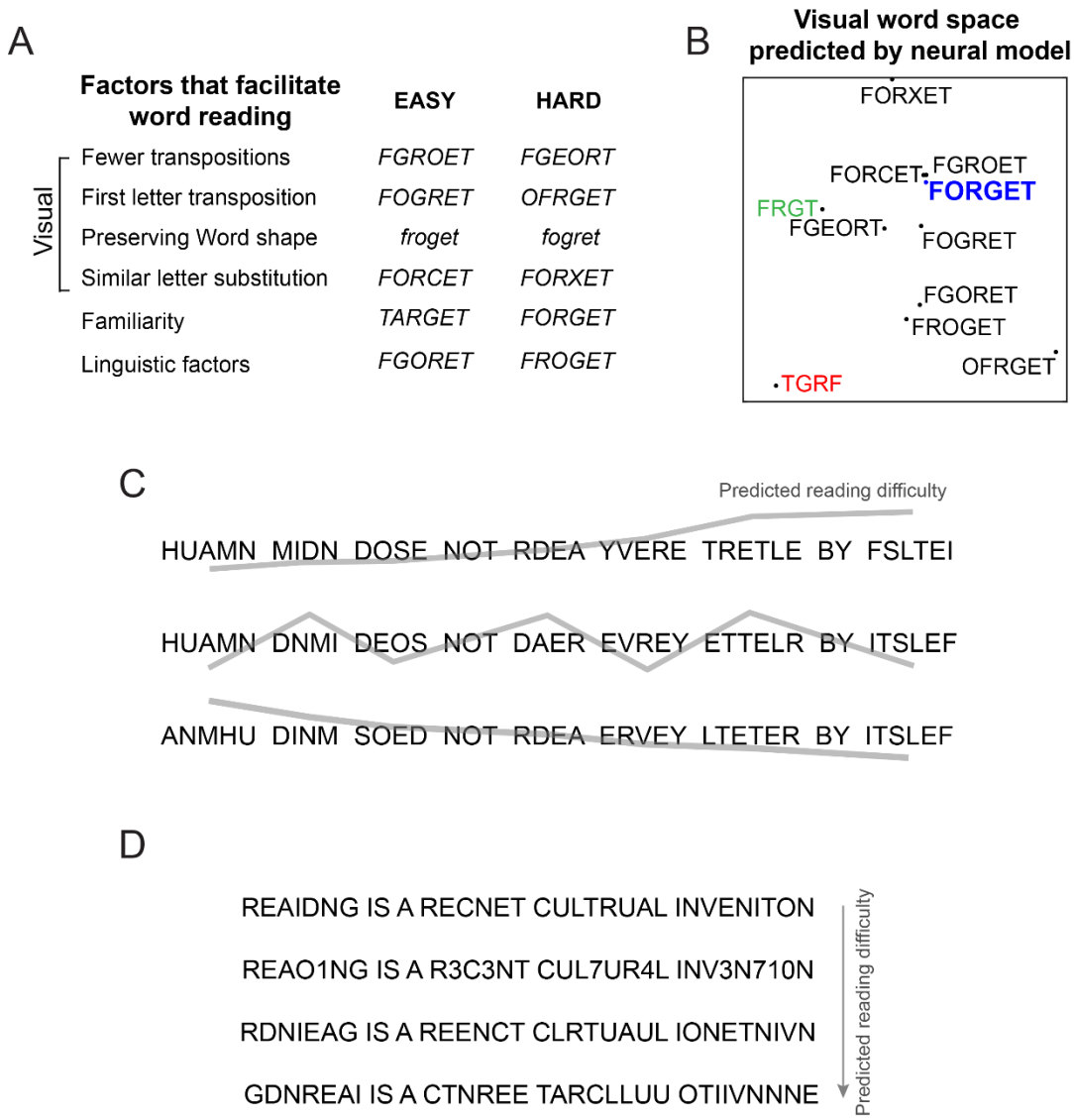


Figure 7. Predicting reading difficulty using the neural model

(A) Factors that facilitate word reading (same as Fig. 1B).

(B) Visual word space predicted by the neural model for a word (FORGET) and its jumbled versions from panel A. Neural model predictions were based on training the model on compound words (Experiment 3). The plot was obtained by performing multidimensional scaling on the pairwise dissimilarities between strings predicted by the neural model. It can be seen that classic features of orthographic processing are captured by the neural model, including priming effects such as FRGT (*green*) being more similar to FORGET than TGRF (*red*).

(C) The neural model can be used to sort jumbled words by their reading difficulty, allowing us to create any desired reading difficulty profile along a sentence. *Top row*: Sentence with increasing reading difficulty. *Middle row*: sentence with fluctuating reading difficulty. *Bottom row*: sentence with decreasing reading difficulty.

(D) The neural model yields a composite measure of reading difficulty that combines letter substitution and transposition effects. Sentences with digit substitutions (*second row*) can thus be placed along a continuum of reading difficulty relative to other sentences (*first, third and fourth rows*) with increasing degree of scrambling.

METHODS

All subjects had normal or corrected-to-normal vision and gave informed consent to an experimental protocol approved by the Institutional Human Ethics Committee of the Indian Institute of Science. All subjects were fluent English speaking students at the institute, where English is the medium of instruction.

Experiment 1 – Single letter searches

Procedure. A total of 16 subjects (8 males, 24.4 ± 2.5 years) participated in this experiment. Subjects were seated comfortably in front of a computer monitor placed ~60 cm away under the control of custom programs written in Psychtoolbox (*38*) and MATLAB.

Stimuli. Single letter images were created using the Arial font. There were 62 stimuli in all comprising 26 uppercase letters (A-Z), 26 lowercase letters (a-z), and 10 digits (0-9). Uppercase stimuli were scaled to have a longer dimension of 1°.

Task. Subjects were asked to perform an oddball search task without any constraints on eye movements. Each trial began with a fixation cross shown for 0.5 s followed by a 4x4 search array (measuring 40° by 25°). The search array always contained only one oddball target with 15 identical distractors. Subject were instructed to locate the oddball target as quickly and as accurately as possible, and respond with a key press ('Z' for left, 'M' for right). A red line divided the screen in two halves. The search display was turned off after the response or after 10 seconds, whichever was sooner. All stimuli were presented in white against a black background. Incorrect or missed trials were repeated after a random number of other trials. Subjects completed a total of 3,782 correct trials ($^{62}C_2$ letter pairs x 2 repetitions with either letter as target once).

For each search pair, the oddball target appeared equally often on the left and right sides so as to avoid creating any response bias. Only correct responses were considered for further analysis. The main experiment was preceded by 20 practice trials involving unrelated stimuli.

Data Analysis. Subjects were highly accurate on this task (mean ± std: 98 ± 1%). Outliers in the reaction times were removed using built-in routines in MATLAB (*isoutlier* function, MATLAB R2018a). This function removes any value greater than three scaled absolute deviations away from the median, and was applied to each search pair separately. This step removed 6.8% of the response time data.

Estimation of single letter tuning using multidimensional scaling

To estimate neural responses to single letters from the visual search data, we used a multidimensional scaling (MDS) analysis. We first calculated the average search time for each letter pair by averaging across subjects and trials. We then converted this search time (RT) into a distance measure by taking its reciprocal (1/RT). This is a meaningful measure because it represents the underlying rate of evidence accumulation in visual search (*39*), behaves like a mathematical distance metric (*14*) and combines linearly with a variety of factors (*39–41*). Next we took all pairwise distances between letters and performed MDS to embed letters into n dimensions, where we varied n from 1 to 15. This yielded n-dimensional coordinates corresponding to each letter, whose distances matched best with the observed distances. We then took the activation of each letter along a given dimension as the response of a single neuron. Throughout we performed MDS embedding into 10 dimensions, resulting in

single letter responses of 10 neurons. We obtained qualitatively similar results on varying this number of dimensions.

Estimation of data reliability

To obtain upper bounds on model performance, we reasoned that any model can predict the data as well as the consistency of the data itself. Thus, a model trained on one half of the subjects can only predict the other half as well as the split-half correlation r_{sh}. This process was repeated 100 times to obtain the mean and standard deviation of the split-half correlation. However when a model is trained on all the data, the upper bound will be larger than the split-half correlation. We obtained this upper bound, which represents the reliability of the entire dataset (r_{data}) by applying a Spearman-Brown correction on the split-half correlation, as given by $r_{data} = 2r_{sh}/(r_{sh}+1)$.

Experiment 2 – Bigram searches

A total of 8 subjects (5 male, aged 25.6 ± 2.9 years) took part in this experiment. We chose seven uppercase letters (A, D, H, I, M, N, T) and combined them in all possible ways to obtain 49 bigram stimuli. These letters were chosen to maximise the number of two-letter words e.g. HI, IT, IN, AN, AM, AT, AD, AH, and HA. Letters measured 3° along the longer dimension and were identical to Experiment 1. Subjects completed 2352 correct trials ($^{49}C_2$ search pairs x 2 repetitions). All other details were identical to Experiment 1. Letter/Bigram frequencies were obtained from an online database (http://norvig.com/mayzner.html).

Data Analysis. Subjects were highly accurate on this task (mean ± std: 97.6 ± 1.8%). Outliers in the reaction times were removed using built-in routines in MATLAB (*isoutlier* function, MATLAB R2018a). This step removed 8% of the response time data.

Estimating neural model parameters from observed dissimilarities

The total dissimilarity between two bigrams in the neural model is calculated by calculating the average dissimilarity across all neurons. For each neuron, the dissimilarity between bigrams AB & CD is given by:

$$d(AB, CD) = |r_{AB} - r_{CD}| = |(w_1 r_A + w_2 r_B) - (w_1 r_C + w_2 r_D)|$$

where $r_A, r_B, r_C \; and \; r_D$ are the responses of the neuron to individual letters A, B, C and D respectively (derived from single letter dissimilarities), and w_1, w_2 are the spatial summation weights for the first and second letters of the bigram. Note that w_1, w_2 are the only free parameters for each neuron.

To estimate the spatial weights of each neuron, we adjusted them so as to minimize the squared error between the observed and predicted dissimilarity. This adjustment was done using standard gradient descent methods starting from randomly initialized weights (*nlinfit* function, MATLAB R2018a). We followed a similar approach for experiments involving longer strings.

Experiment 3 – Compound word searches

A total of 8 subjects (4 female, aged 25 ± 2.5 years) participated. Twelve 3-letter words were chosen: ANY, FOR, TAR, KEY, SUN, TEA, ONE, MAT, GET, PAD, DAY, POT. Each word was scrambled to obtain twelve 3-letter nonwords containing

the same letters. The 12 words were combined to form 36 compound words (shown in Section S5), such that they appeared equally on the left and right half of the compound words. The compound words measured 6° along the longer dimension. Subjects completed 1260 correct trials ($^{36}C_2$ search pairs x 2 repetitions). Additionally, subjects also performed visual search on 3-letter words (n = 132, $^{12}C_2$ x 2 repetitions) and their jumbled versions (n = 132). Trials timed out after 15 seconds. All other details were identical to Experiment 1.

Data Analysis. Subjects were highly accurate on this task (mean ± std: 98 ± 1%). Outliers in the reaction times were removed using built-in routines in MATLAB (*isoutlier* function, MATLAB R2018a). This step removed 6.4% of the response time data.

Experiment 4 – Scrambled word task

Procedure. A total of 16 subjects (9 male, aged 24.8 ± 2.1 years) participated in the task. Other details were similar to Experiment 1.

Stimuli. We chose 300 words such that no two words were anagrams of each other. These comprised 75 four-letter words, 150 five-letter words and 75 six-letter words. Jumbled words were created by shuffling 2, 3, or 4 letters of each word. There were an equal proportion of 2, 3, and 4 letter transpositions. All stimuli were presented in uppercase against a black background.

Task. Each trial began with a fixation cross shown for 0.5 s followed by a scrambled word that appeared for 5 seconds (for the first 6 subjects) and 7 seconds (for the rest), or until the subject made a response by pressing the space bar on the keyboard. To ensure that subjects actually solved the scrambled word, they were asked to type the unscrambled word within 10 seconds of pressing the space bar. The response time

was taken as the time at which the subject pressed the space bar. To avoid any memory effects, the same set of jumbled words were shown to all subjects exactly once. We analysed response times only on trials in which the subject subsequently entered the correct word.

Data Analysis. Subjects were reasonably accurate on this task (average accuracy: 59.5 ± 8% across 300 words). Response times for wrongly typed words were discarded. Words correctly solved by more than 6 subjects (n = 238) were included for further analysis. Since trials were self-paced, we did not remove any outliers in the reaction times. Lexical properties were obtained from the English Lexicon Project (*42*).

Experiment 5 – Lexical decision task

Procedure. A total of 16 subjects (9 male, aged 24.8 ± 2.1 years) participated in this task as well as the scrambled word task.

Stimuli. The stimuli comprised 450 words + 450 nonwords. The nonwords were either random strings or made by modifying the 450 words in some way (Section S6).

Task. Each trial began a fixation cross shown for 0.75 s followed by a letter string for 0.2 s after which the screen went blank. The trial ended either with the subject's response or after at most 3 s. Subjects were instructed to press 'Z' for words and 'M' for nonwords as quickly and accurately as possible. All stimuli were presented at the centre of the screen and were white letters against a black background. Before starting the main task, subjects were given 20 practice trials using other words and nonwords not included in the main experiment.

Data Analysis. Some nonwords were removed from further analysis due to low accuracy (n = 8, average accuracy <20%). Subjects made accurate responses for both

801 words and nonwords (mean ± std of accuracy: 96 ± 2 % for words, 95 ± 3% for
802 nonwords). Outliers in the reaction times were removed using built-in routines in
803 MATLAB (*isoutlier* function, MATLAB R2018a).
804

REFERENCES

1. G. E. Rawlinson, thesis (1976).
2. J. Grainger, C. Whitney, Does the huamn mnid raed wrods as a wlohe? *Trends Cogn. Sci.* **8**, 58–9 (2004).
3. D. Norris, Models of visual word recognition. *Trends Cogn. Sci.* **17**, 517–524 (2013).
4. J. Grainger, Orthographic processing: A 'mid-level' vision of reading: The 44th Sir Frederic Bartlett Lecture. *Q. J. Exp. Psychol.* **71**, 335–359 (2018).
5. M. Perea, V. Panadero, Does viotin activate violin more than viocin? On the use of visual cues during visual-word recognition. *Exp. Psychol.* **61**, 23–9 (2014).
6. M. Perea, J. A. Duñabeitia, M. Carreiras, R34D1NG W0RD5 W1TH NUMB3R5. *J. Exp. Psychol. Hum. Percept. Perform.* **34**, 237–41 (2008).
7. K. Rayner, S. J. White, R. L. Johnson, S. P. Liversedge, Raeding wrods with jubmled lettres: There is a cost. *Psychol. Sci.* **17**, 192–193 (2006).
8. P. Gomez, R. Ratcliff, M. Perea, The overlap model: a model of letter position coding. *Psychol. Rev.* **115**, 577–600 (2008).
9. A. P. Sripati, C. R. Olson, Global Image Dissimilarity in Macaque Inferotemporal Cortex Predicts Human Visual Search Efficiency. *J. Neurosci.* **30**, 1258–1269 (2010).
10. K. A. Zhivago, S. P. Arun, Texture discriminability in monkey inferotemporal cortex predicts human texture perception. *J. Neurophysiol.* **112**, 2745–55 (2014).
11. H. Op de Beeck, J. Wagemans, R. Vogels, Inferotemporal neurons represent low-dimensional configurations of parameterized shapes. *Nat. Neurosci.* **4**, 1244–1252 (2001).
12. D. Zoccolan, D. D. Cox, J. J. DiCarlo, Multiple Object Response Normalization in Monkey Inferotemporal Cortex. *J. Neurosci.* **25**, 8150–8164 (2005).
13. G. M. Ghose, J. H. Maunsell, Spatial summation can explain the attentional modulation of neuronal responses to multiple stimuli in area v4. *J Neurosci*. **28**, 5115–5126 (2008).
14. S. P. Arun, Turning visual search time on its head. *Vision Res.* **74**, 86–92 (2012).
15. R. T. Pramod, S. P. Arun, Symmetric Objects Become Special in Perception Because of Generic Computations in Neurons. *Psychol. Sci.* **29**, 95–109 (2018).
16. D. G. Pelli, K. A. Tillman, Parts, wholes and context in reading: A triple dissociation. *PLoS One*. **2**, e680 (2007).
17. M. Scaltritti, S. Dufau, J. Grainger, Stimulus orientation and the first-letter advantage. *Acta Psychol. (Amst).* **183**, 37–42 (2018).
18. R. Ratcliff, G. McKoon, The Diffusion Decision Model: Theory and Data for Two-Choice Decision Tasks. *Neural Comput.* **20**, 873–922 (2008).
19. R. Ratcliff, P. Gomez, G. McKoon, A diffusion model account of the lexical decision task. *Psychol. Rev.* **111**, 159–82 (2004).
20. M. J. Yap, D. E. Sibley, D. A. Balota, R. Ratcliff, J. Rueckl, Responding to nonwords in the lexical decision task: Insights from the english Lexicon project. *J. Exp. Psychol. Learn. Mem. Cogn.* **41**, 597–613 (2015).
21. S. Dufau, J. Grainger, J. C. Ziegler, How to say "No" to a nonword: A leaky competing accumulator model of lexical decision. *J. Exp. Psychol. Learn. Mem. Cogn.* **38**, 1117–1128 (2012).

22. J. Grainger, S. Dufau, M. Montant, J. C. Ziegler, J. Fagot, Orthographic processing in baboons (Papio papio). *Science (80-.).* **336**, 245–8 (2012).
23. J. C. Ziegler *et al.*, Transposed-Letter Effects Reveal Orthographic Processing in Baboons. *Psychol. Sci.* **24**, 1609–1611 (2013).
24. N. A. Ratan Murty, S. P. Arun, Dynamics of 3D view invariance in monkey inferotemporal cortex. *J. Neurophysiol.* **113**, 2180–94 (2015).
25. A. P. Sripati, C. R. Olson, Responses to compound objects in monkey inferotemporal cortex: the whole is equal to the sum of the discrete parts. *J. Neurosci.* **30**, 7948–60 (2010).
26. P. Bao, D. Y. Tsao, Representation of multiple objects in macaque category-selective areas. *Nat. Commun.* **9**, 1774 (2018).
27. S. Dehaene, L. Cohen, M. Sigman, F. Vinckier, The neural code for written words: a proposal. *Trends Cogn. Sci.* **9**, 335–41 (2005).
28. S. Dehaene, L. Cohen, J. Morais, R. Kolinsky, Illiterate to literate: behavioural and cerebral changes induced by reading acquisition. *Nat. Rev. Neurosci.* **16**, 234–44 (2015).
29. M. Chanceaux, J. Grainger, Serial position effects in the identification of letters, digits, symbols, and shapes in peripheral vision. *Acta Psychol. (Amst).* **141**, 149–158 (2012).
30. R. E. B. Mruczek, D. L. Sheinberg, Distractor familiarity leads to more efficient visual search for complex stimuli. *Percept. Psychophys.* **67**, 1016–31 (2005).
31. A. Marcet, M. Perea, Is nevtral NEUTRAL? Visual similarity effects in the early phases of written-word recognition. *Psychon. Bull. Rev.* **24**, 1180–1185 (2017).
32. C. J. Davis, The spatial coding model of visual word identification. *Psychol. Rev.* **117**, 713–758 (2010).
33. D. Norris, S. Kinoshita, Reading through a noisy channel: why there's nothing special about the perception of orthography. *Psychol. Rev.* **119**, 517–45 (2012).
34. T. Yarkoni, D. Balota, M. Yap, Moving beyond Coltheart's N: a new measure of orthographic similarity. *Psychon. Bull. Rev.* **15**, 971–9 (2008).
35. M. Zorzi *et al.*, Extra-large letter spacing improves reading in dyslexia. *Proc. Natl. Acad. Sci. U. S. A.* **109**, 11455–9 (2012).
36. K. Rayner, Eye movements in reading and information processing: 20 years of research. *Psychol. Bull.* **124**, 372–422 (1998).
37. D. G. Pelli, K. a Tillman, The uncrowded window of object recognition. *Nat. Neurosci.* **11**, 1129–1135 (2008).
38. D. H. Brainard, The Psychophysics Toolbox. *Spat. Vis.* **10**, 433–436 (1997).
39. S. Sunder, S. P. Arun, Look before you seek: Preview adds a fixed benefit to all searches. *J. Vis.* **16**, 3 (2016).
40. R. T. Pramod, S. P. Arun, Object attributes combine additively in visual search. *J. Vis.* **16**, 8 (2016).
41. R. T. Pramod, S. P. Arun, Features in visual search combine linearly. *J. Vis.* **14**, 1–20 (2014).
42. D. A. Balota *et al.*, The English Lexicon Project. *Behav. Res. Methods*. **39**, 445–459 (2007).

ACKNOWLEDGEMENTS

Funding. This research study was funded by Intermediate & Senior Fellowships (Grant Numbers 500027/Z/09/Z and IA/S/17/1/503081) from the Wellcome Trust/DBT India Alliance to SPA.

Author contributions. AA, KVSH & SPA designed experiments; AA collected data; AA & SPA analysed and interpreted the data; SPA wrote the manuscript with inputs from AA & KVSH.

Competing interests. The authors declare no competing interests.

Data and code availability. Data and code necessary to reproduce the results are available from the authors upon reasonable request.

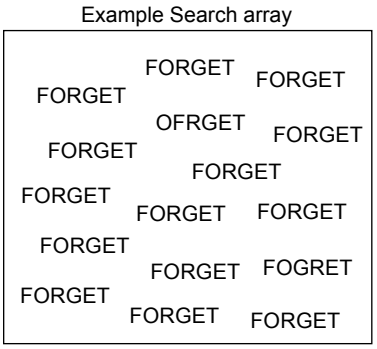
A

AOCCDRNIG TO A RSEEARCH AT CMABRIGDE UINERVTISY, IT DEOSN'T MTTAER IN WAHT OREDR THE LTTEERS IN A WROD ARE, THE OLNY IPRMOETNT TIHNG IS TAHT THE FRIST AND LSAT LTTEER BE AT THE RGHIT PCLAE. THE RSET CAN BE A TOATL MSES AND YOU CAN SITLL RAED IT WOUTHIT A PORBELM. TIHS IS BCUSEAE THE HUAMN MNID DEOS NOT RAED ERVEY LTETER BY ISTLEF, BUT THE WROD AS A WLOHE.

B

	Factors that facilitate word reading	EASY	HARD
Visual	Fewer transpositions	*FGROET*	*FGEORT*
Visual	First letter transposition	*FOGRET*	*OFRGET*
Visual	Preserving Word shape	*froget*	*fogret*
Visual	Similar letter substitution	*FORCET*	*FORXET*
	Familiarity	*TARGET*	*FORGET*
	Linguistic factors	*FGORET*	*FROGET*

C



D

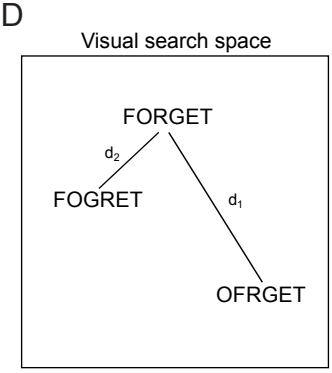


Figure 1

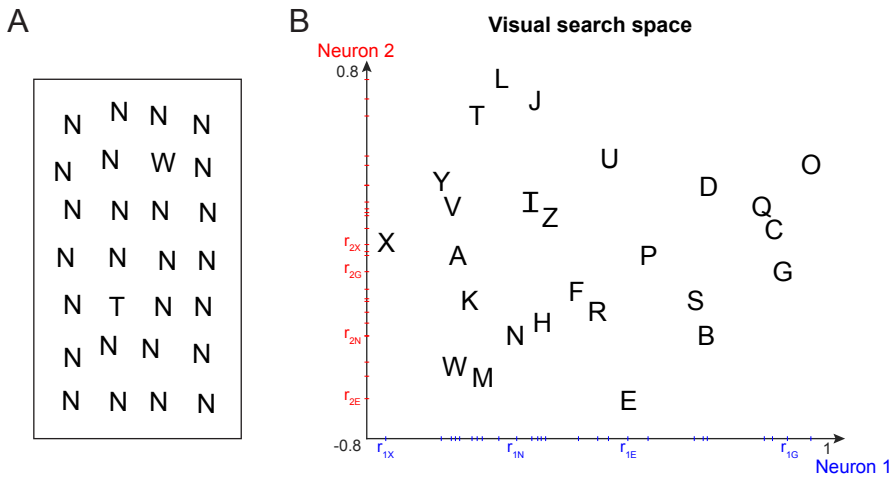


Figure 2

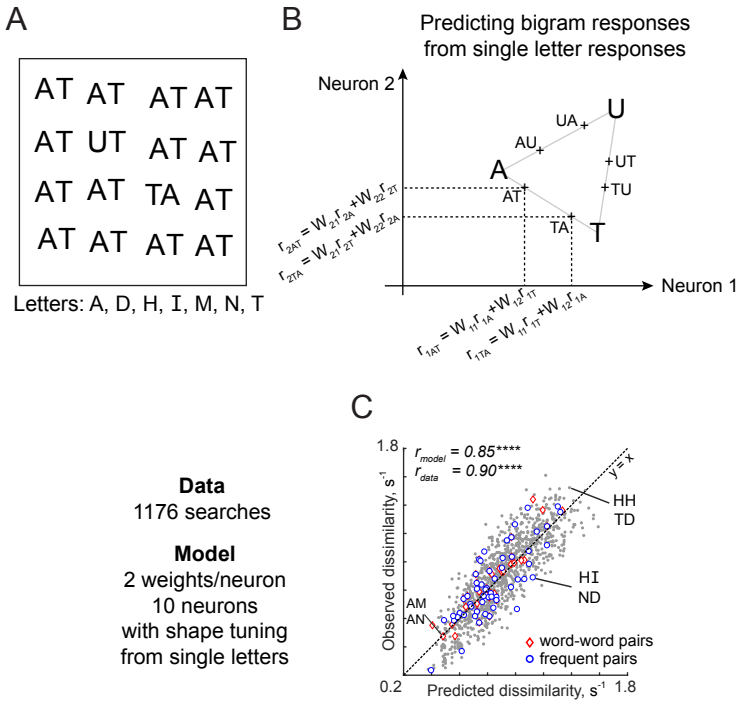


Figure 3

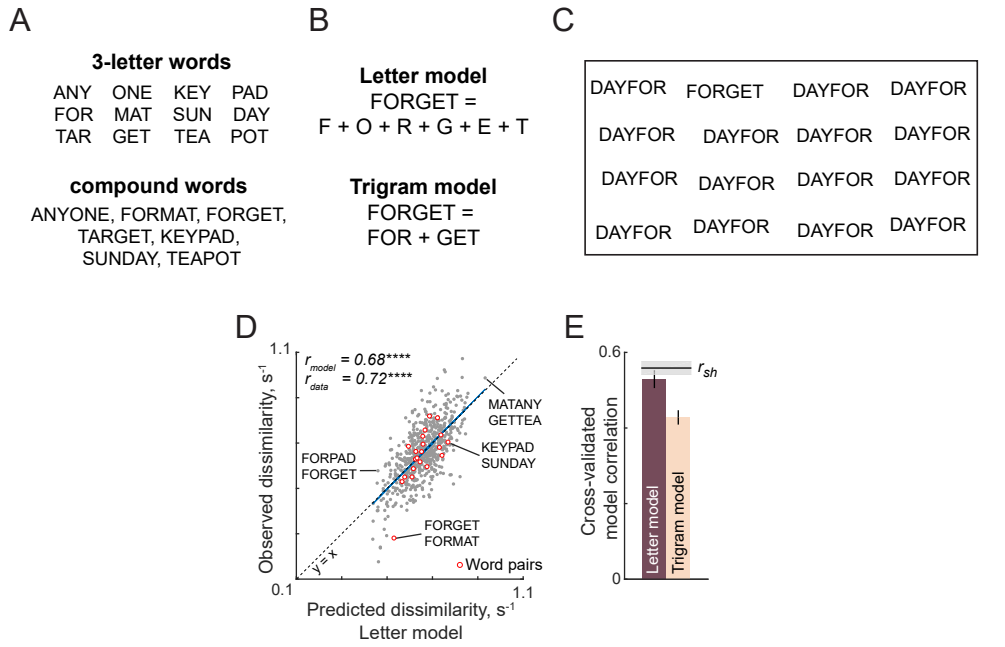


Figure 4

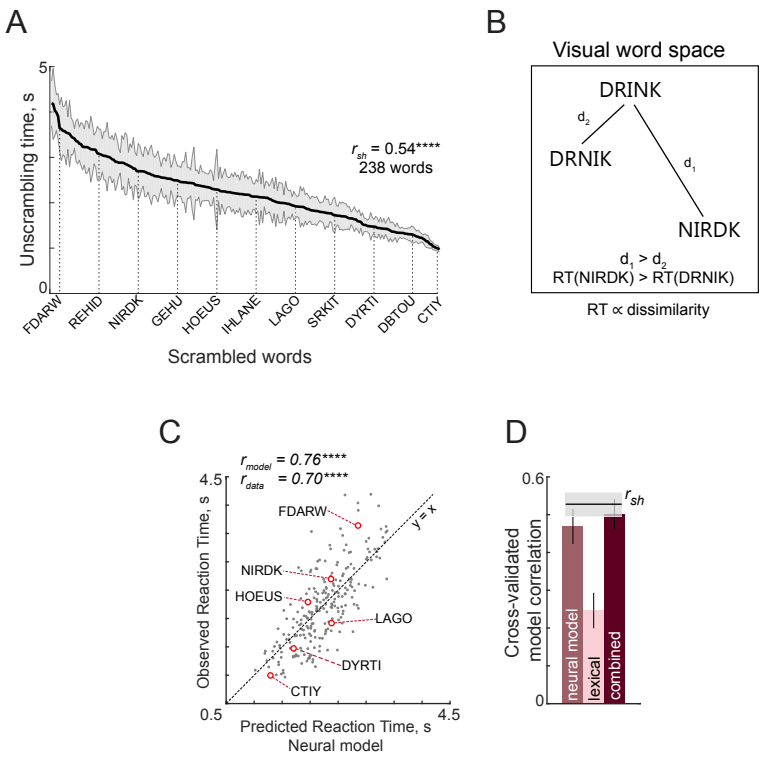


Figure 5

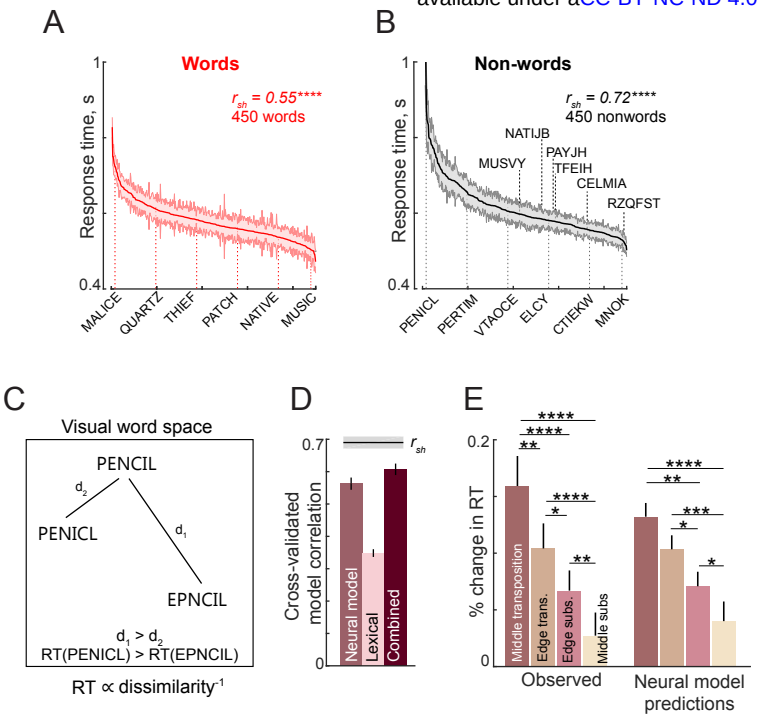
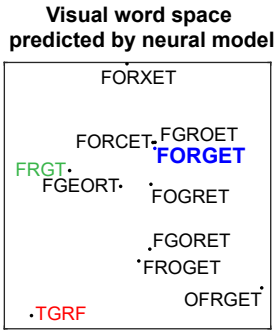


Figure 6

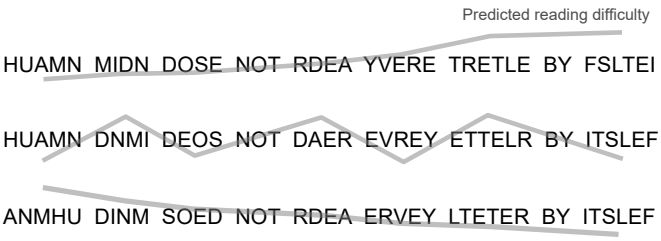
A

	Factors that facilitate word reading	EASY	HARD
Visual	Fewer transpositions	*FGROET*	*FGEORT*
	First letter transposition	*FOGRET*	*OFRGET*
	Preserving Word shape	*froget*	*fogret*
	Similar letter substitution	*FORCET*	*FORXET*
	Familiarity	*TARGET*	*FORGET*
	Linguistic factors	*FGORET*	*FROGET*

B



C



D

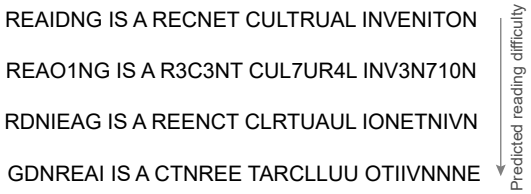


Figure 7

SUPPLEMENTARY MATERIAL

For

A compositional letter code explains how we read jumbled words

CONTENTS

SECTION S1. ADDITIONAL ANALYSIS FOR EXPERIMENT 1

The results in the main text were presented for uppercase English letters (Fig. 2), but in Experiment 1 we also collected visual search data for all pairs of English letters and numbers (n = 62 characters in all, comprising 26 uppercase + 26 lowercase + 10 numbers). We did so in order to predict the visual dissimilarity between letter strings containing both mixed case letters as well as numbers.

To visualize the dissimilarity relations between the 62 characters used, we performed multidimensional scaling. In the resulting plot (Fig. S1A), nearby characters represent hard searches. A number of interesting patterns can be seen: letters like C, G, Q, O are nearby which is expected given their shared curvatures. Letter pairs such as (M,W) and (6,9) are similar due to mirror confusion (1).

Next, we investigated the degree to which the observed pairwise dissimilarities are captured by the multidimensional embedding as a function of the number of dimensions. In the resulting plot (Fig. S1B), it can be seen that nearly 89% of the variance is captured by 10 dimensions as before, which reaches roughly the reliability of the dissimilarity data itself. For the analyses involving mixed case searches or fewer searches, we took a total of 6 neurons for the neural model, which explain 87.7% of the variance in the pairwise dissimilarities.

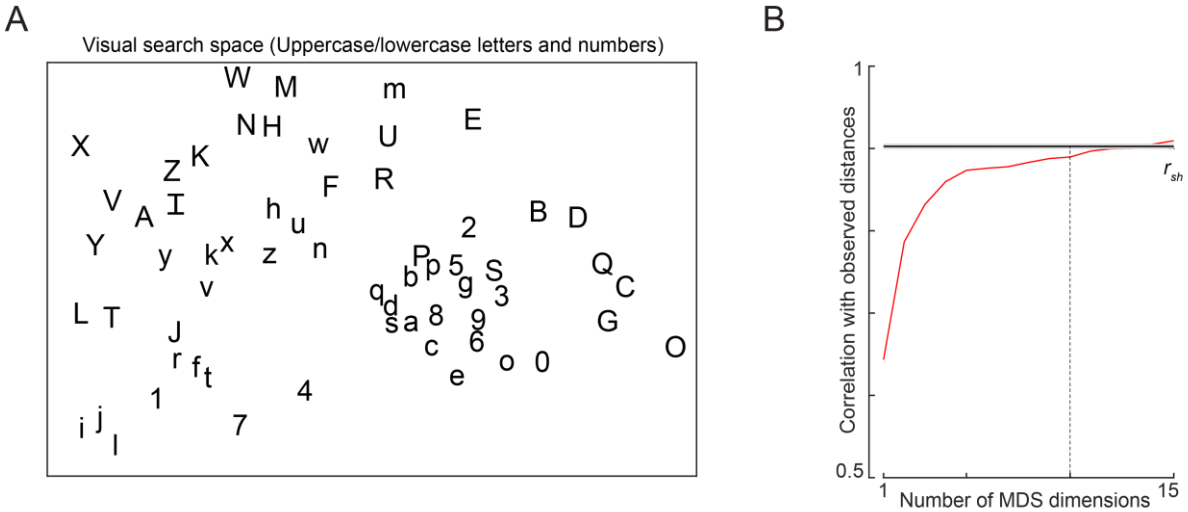


Figure S1. Visual search space for letters and digits

(A) Visual search space for letters (uppercase and lowercase) and digits obtained by multidimensional scaling of observed dissimilarities. Nearby letters represent hard searches. Distances in this 2D plot are highly correlated with the observed distances ($r = 0.79$, $p < 0.00005$).

(B) Correlation between observed distances and MDS embedding as a function of number of MDS dimensions. The horizontal line represents the split-half correlation with error bars representing s.d calculated across 100 random splits.

Can letter dissimilarity be predicted using low-level visual features?

To investigate whether single letter dissimilarity can be predicted using low-level visual features, we attempted to predict letter dissimilarities using two models. In the first model, which we call the pixel model, we calculated the dissimilarity between letters to be the absolute difference in pixel intensities between the images of the two letters. This pixel-based model showed a significant correlation ($r = 0.50$, $p < 0.00005$)

but was far from the reliability of the data itself (r_{sh} = 0.90; Fig. S1B). In the second model, we calculated the dissimilarity between two letters as the vector distance between the responses evoked by a population of simulated V1 neurons (2). This V1 model also showed a significant correlation (r = 0.44, p < 0.00005) but again far from the reliability of the data itself). We conclude that single letter dissimilarity can only be partially predicted by low-level visual features.

Is visual search dissimilarity related to subjective dissimilarity?

In this study, we have used visual search as a natural and objective measure for visual dissimilarity. However previous studies have measured letter dissimilarity either through confusions in letter recognition, or through subjective dissimilarity ratings (3, 4). We have previously shown that subjective dissimilarity for abstract silhouettes is strongly correlated with visual search dissimilarity (5). This may not hold for letters since subjects can activate letter representations that are modified through extensive familiarity. To investigate how visual search dissimilarity compares with subjective similarity ratings for letters, we compared search dissimilarities for uppercase letters against two sets of previously reported similarity data. First, we compared visual search dissimilarities with subjective dissimilarity ratings (4). This revealed a significant positive correlation (r = 0.69, p < 0.0005). Second, we compared visual search dissimilarities with letter confusion data (*3*). To convert letter confusion response times, which are a measure of similarity, into dissimilarities, we took their reciprocals, and then compared them with visual search dissimilarities. This revealed a significant positive, albeit weaker correlation (r = 0.34 p< 0.0005).

SECTION S2. EXPERIMENTS WITH LONGER STRINGS

In the main text, we showed that bigram dissimilarity in visual search can be explained using a simple neural model with single letter responses that match perception, and a compositional spatial summation rule that predicts responses to bigrams. Here we asked whether this approach would generalize to longer strings of letters.

To this end, we performed four additional experiments on longer strings. In Experiment 6, we created trigrams with a fixed middle letter and all possible combinations of flanking letters, to create multiple three-letter words. In Experiment 7, subjects performed searches involving 3, 4, 5 and 6-letter searches with uppercase, lowercase and mixed case strings. In Experiments 8 & 9, we attempted to optimize the search pairs used to estimate model parameters.

RESULTS

Cross-validated model fits across all experiments are shown in Figure S2. It can be seen that the neural model fit is consistently close to the split-half consistency of the data. Thus, visual discrimination of longer strings can be explained using a compositional neural code. Below we discuss some experiment-specific findings of interest.

Lowercase and mixed-case strings

Word shape is thought to play a role in reading lowercase letters, because of the upward deflection (e.g. l, d) and downward deflections (e.g. p, g) of letters which might confer a specific overall shape to a word. To conclusively establish this would require factoring out the contribution of individual letters to word discrimination, as with the neural model. We were therefore particularly interested in whether the neural model would predict the dissimilarity between lowercase and mixed-case strings where word shape might potentially play a role. As can be seen in Figure S2, cross-validated model predictions for lowercase letters were highly correlated with the observed data ($r = 0.59$, $p < 0.00005$). This correlation approached the upper bound given by the split-half reliability itself ($r_{sh} = 0.64$). Likewise, model predictions for mixed-case letters were also highly correlated with the observed data ($r = 0.59$, $p < 0.00005$; Fig. S2). However in this case model fits were well below the split-half consistency ($r_{sh} = 0.72$), suggesting that there is still some systematic unexplained variance in mixed-case strings. This gap in model fit could be simply due to the relatively few mixed-case searches used in this experiment (n = 100), or because of unaccounted factors like word shape. Nonetheless, the neural model explains a substantial fraction of variation in both lowercase and mixed case strings, suggesting that it can be used as a powerful baseline to elucidate the contribution of word shape to reading.

Unequal length strings

The neural model can be used to calculate responses to any string length, provided the spatial summation weights are known. Given the relatively few searches for unequal lengths in our data, we fit the neural model to unequal length strings using 6 neurons. Doing so still raised a fundamental issue: which subset of the 6 spatial summation weights for each neuron should be used to calculate the response to a 4-letter string? This requires aligning the 4-letter string to the 6-letter string in some manner.

To address this issue, we evaluated the neural model fit on four possible alignments between longer and shorter strings, and asked whether model predictions were better for any one alignment compared to others. We aligned the smaller length string to either the left, right, centre or edge of the longer string. Model performance for these different variations is shown in Table S1. It can be seen that the model fits are comparable across different choices. However, edge alignment is slightly but not significantly better than other choices. We therefore used edge alignment for all subsequent model predictions.

Alignment	Neural model correlation			
	6 vs 5	6 vs 4	5 vs 3	4 vs 3
Left: ABCDEF vs EFGHxx	0.54	0.66	0.58	0.57
Right: ABCDEF vs xxEFGH	0.51	0.66	0.57	0.58
Centre: ABCDEF vs xEFGHx	-	**0.68**	0.58	-
Edge: ABCDEF vs EFxxGH	**0.55**	0.63	**0.60**	**0.59**

Table S1: Model fits for various choices of string alignment. In each case we fit the neural model with unknown weights corresponding to the longer length. The alignment is indicated by the position of "x"s in the string. For instance, "Left" alignment means that a 6-letter string ABCDEF is matched to a 4-letter string EFGH by assuming that the response to EFGH is created using the first four weights of spatial summation. Likewise, right alignment means that EFGH is aligned to the right, and therefore its response is created using the last four weights in the 6-letter neural model. The best alignment is highlighted for each column in **bold**. None of the correlation coefficient differences were statistically significant ($p > 0.05$, Fisher's z-test).

METHODS

Experiment 6: Trigrams with fixed middle letter. A total of 8 subjects (5 males, aged 23.9 ± 1.8 years) participated in this experiment. Seven uppercase letters: A, E, I, P, S, T and Y were combined (around the stem R i.e. xRx) in all pairs to form a total of 49 stimuli. These letters were chosen to maximize the occurrence of 3-letter words and psuedowords in the stimulus set. The longer dimension of the stimuli was ~5°. Each subject completed searches corresponding to all possible pairs of stimuli (${}^{49}C_2$ = 1176) with two trials for each search. All other details were identical to Experiment 2.

Experiment 7: Random string searches. A total of 12 subjects (9 female, aged 24.8 ± 1.64 years) participated in this experiment. All 26 uppercase and lowercase letters were used to create 1800 stimuli, which were organized into 900 stimulus pairs with varying string length. These 900 pairs comprised 300 6-gram uppercase pairs, 100 6-gram lowercase pairs, 100 6-gram mixed-case pairs, 100 5-gram uppercase pairs, 50 4-gram uppercase pairs, 50 3-gram uppercase pairs and 200 pairs with uppercase strings of differing lengths (50 pairs each of 6- vs 5-grams, 6- vs 4-grams, 5- vs 4-grams, 5- vs 3-grams = 200 pairs total). For each string length, letters were randomly combined to form strings with a constraint that all 26 letters should appear at least once at each location. Each stimulus pair was shown in two searches (with either item as target, and either on the left or right side). The trial timed out at 15 seconds for all searches.

Experiment 8 – Optimized 6-letter searches. A total of 9 subjects (5 males, aged 24.1 ± 2.2 years) participated in this experiment. To maximize the importance of each

spatial location in a 6-letter uppercase string, stimuli were created such that there were at least 75 search pairs with the same letter at either of the corresponding locations. Further, to reliably estimate the model parameters, the randomly chosen letters were arranged to minimize the condition number of the linear regression matrix X (of the ISI model described below). In all there were 300 search pairs. The trial timed out after 15 seconds. All other details were similar to Experiment 2.

Experiment 9 – Optimized 4-letter searches. In all, 8 subjects (5 females, aged 23.5 ± 2.3 years) participated in this experiment. We chose 300 search pairs with 4-letters, according to the same criteria as in Experiment 8. All other details were the same as in Experiment 8.

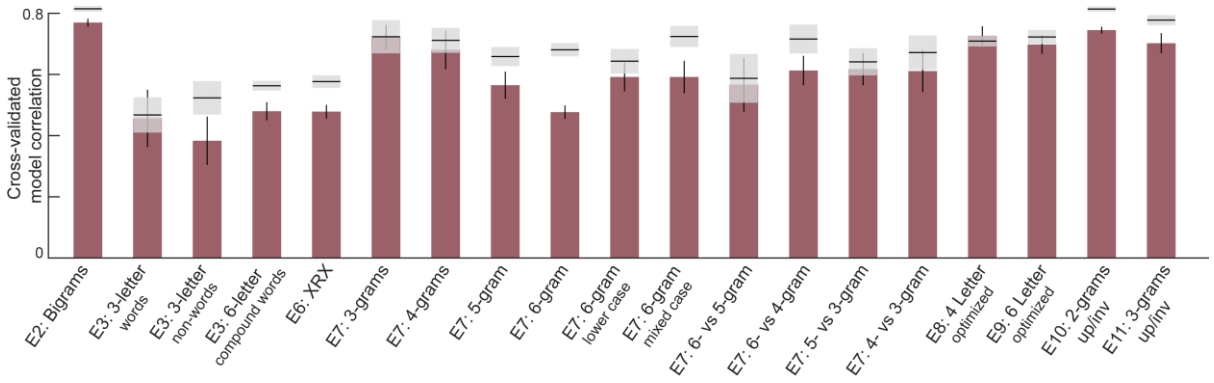


Figure S2. Neural model performance for varying length strings. For each experiment, we obtained a cross-validated measure of model performance using 6 neurons as follows: each time we divided the subjects randomly into two halves, and trained the neural model on one half of the subjects and tested it on the other half. This was repeated for 30 random splits. The correlation between the model predictions and the average dissimilarity from the held-out half of the data was taken to be the model fit. The correlation between the observed dissimilarity between the two random splits of subjects is then the upper bound on model performance (mean ± std shown as *gray shaded bars*).

SECTION S3. ESTIMATING LETTER DISSIMILARITIES FROM BIGRAMS

Part-sum model

The neural model described in the text has many desirable features but requires as input the responses to single letters, which were obtained from searches involving single isolated letters. However, it could be that bigram representations can be understood in terms of component letter responses that are different from the responses of letters seen in isolation. It could also be that letter responses are different at each location.

To address these issues, we developed an alternate model in which bigram dissimilarities can be written in terms of unknown single letter dissimilarities. These single letter dissimilarities can be estimated in the model. In this model, which we call the part-sum model, the dissimilarity between two bigrams AB & CD is written as the sum of all pairs of part dissimilarities in the two bigrams (Fig. S3A). Specifically:

$$d(AB,CD) = CL_{AC} + CR_{BD} + X_{AD} + X_{BC} + W_{AB} + W_{CD} + \text{constant}$$

where CL_{AC} is the dissimilarity between letters at Corresponding Left (CL) locations (A & C), CR_{BD} is the dissimilarity between letters at the Corresponding Right (CR) locations (B & D), X_{AD} & X_{BC} are the dissimilarities between letters across locations in the two bigrams (A & D, B & C), and W_{AB} & W_{CD} are the dissimilarities of letters within each bigram.

The part-sum model works because a given letter dissimilarity CL_{AC} will occur in the dissimilarity of many bigram pairs (e.g. in the pair AB-CD and in AE-CF) thereby allowing us to estimate its unique contribution. Since there are 7 parts, there are 7C2 = 21 possible part-pairs of each type (i.e. for CL, CR, X and W terms), resulting in 21 x 4 = 84 unknown part dissimilarities. Since a given bigram experiment contains all possible $^{49}C_2$ = 1176 bigram searches, there are many more observations than unknowns. The combined set of bigram dissimilarities can be written in the form of a matrix equation **y** = **Xb** where **y** is a 1176x1 vector of observed bigram dissimilarities, **X** is a 1176 x 85 matrix containing the number of times (0, 1 or 2) a given letter-pair of each type (CL, CR, X & W) contributes to the overall dissimilarity, and b is a 85 x 1 vector of unknown letter dissimilarities of each type (21 each of CL, CR, X & W and one constant term). The unknown letter dissimilarities of each type was estimated using standard linear regression (*regress* function, MATLAB).

The part sum model has several advantages over the neural model: (1) It is linear which means that its parameters can be uniquely estimated; (2) it is compositional in that the net dissimilarity between two bigrams is explained using the constituent parts without invoking more complex interactions; (3) it can account for potentially different part relations at each location in the two bigrams. We have previously shown that the part-sum model can explain the dissimilarities between a variety of objects (5).

The part sum model yielded excellent fits to the data ($r = 0.88$, $p < 0.00005$; Fig. S3B) that were close to the reliability of the data ($r_{data} = 0.90$). As before, we observed no systematic deviations between model fits for frequent bigrams compared to infrequent bigrams (Fig. S3B; average absolute residual error for the top 20 bigram pairs with highest mean bigram frequency: 0.09 ± 0.1 s^{-1}; for the bottom-20 bigram pairs: 0.11 ± 0.08 s^{-1}; $p = 0.42$, rank-sum test). To assess whether the part dissimilarities of each type (CL, CR, X and W) were related to each other, we plotted each of CR, X and W terms against the CL terms (Fig. S3C). The CR and X terms

were highly positively correlated (Fig. S3C), whereas the W terms were negative in sign and negatively correlated (Fig. S3C). The negative values of the W terms means that bigrams with dissimilar letters become less dissimilar, an effect akin to distractor heterogeneity in visual search (1, 6). We conclude that the CL, CR, X and W terms in the part-sum model are driven by a common part representation.

To visualize this underlying letter representation, we performed multidimensional scaling on the estimated part dissimilarities of the CL terms. In the resulting plot, nearby letters represent similar letters (Fig. S3D). It can be seen that I & T, M & N are similar as in the single-letter representation (Fig. S1A). These single letter dissimilarities estimated from bigrams using the part-sum model were highly correlated with the single-letter dissimilarities directly observed from visual search with isolated letters (Fig. S3D).

We conclude that bigram dissimilarities can be predicted from a common underlying letter representation that is identical to that of single isolated letters.

Equivalence between part-sum and neural model

Given that the part-sum model and neural model both give equivalent fits to the data, we investigated how they are related. Consider a single neuron whose response to a bigram AB is given by: $r_{AB} = \alpha r_A + r_B$, where r_A and r_B are its responses to A & B, and α is the spatial weight of A relative to B. Similarly its response to the bigram CD can be written as $r_{CD} = \alpha r_C + r_D$. Then the dissimilarity between AB and CD can be written as

$$
\begin{aligned}
& d(AB,CD)^2 \\
& = (r_{AB} - r_{CD})^2 = (\alpha r_A + r_B - \alpha r_C - r_D)^2 \\
& = \big(\alpha(r_A - r_C) + (r_B - r_D)\big)^2 \\
& = \alpha^2(r_A - r_C)^2 + (r_B - r_D)^2 + 2\alpha(r_A - r_C)(r_B - r_D) \\
& = \alpha^2(r_A - r_C)^2 + (r_B - r_D)^2 + 2\alpha(r_A r_B + r_C r_D - r_A r_D - r_B r_C) \\
& = \alpha^2(r_A - r_C)^2 + (r_B - r_D)^2 + \alpha[(r_A - r_D)^2 + (r_B - r_C)^2 - (r_A - r_B)^2 - (r_C - r_D)^2] \\
& = \alpha^2 d_{AC}^2 + d_{BD}^2 + \alpha(d_{AD}^2 + d_{BC}^2 - d_{AB}^2 - d_{CD}^2) \\
& = \alpha^2 d_{AC}^2 + d_{BD}^2 + \alpha(d_{AD}^2 + d_{BC}^2) \; - \; \alpha(d_{AB}^2 + d_{CD}^2)
\end{aligned}
$$

Thus, the squared dissimilarity between AB & CD can be written as a weighted sum of squared dissimilarities between parts at corresponding locations (A-C & B-D), parts at opposite locations (A-D & B-C) and between parts within each bigram (A-B & C-D), which is essentially the same as the part-sum model. The same argument extends to multiple neurons because the total bigram dissimilarity will be the sum of bigram dissimilarities across all neurons.

There are however two important differences. First, the part sum model is written in terms of a weighted sum of part dissimilarities, whereas the above equation refers to a weighted sum of squared dissimilarities. However, the squared sum of distances and a weighted sum of distances are highly correlated, so the essential relation will still hold. Second, the neural model predicts that the across-bigram terms (X_{AD}, X_{BC}) should be similar in magnitude but opposite in sign to the within-bigram terms (W_{AB}, W_{CD}). These weights are similar in magnitude but not exactly equal, as can be seen in Fig S3C. The part-sum model thus allows for greater flexibility in part interactions compared to the neural model.

Reducing part-sum model complexity

The observation that a common set of letter dissimilarities drive the part-sum model suggests that the part-sum model can be simplified. We therefore devised a reduced version of the part-sum model – called the Independent Spatial Interaction (ISI) model – in which the CL, CR, X and W terms are scaled versions of the single letter dissimilarities (Fig. S3E). Specifically, the dissimilarity between bigrams AB & CD is:

$$d(AB, CD) = \alpha_{10} d_{AC} + \alpha_{20} d_{BD} + \alpha_{11}(d_{AD} + d_{BC}) + \beta_{11}(d_{AB} + d_{CD}) + constant$$

where d_{AC} is the observed dissimilarity between the left letters A & C from visual search and α_{10} is an unknown scaling term, d_{BD} is the observed dissimilarity between the right letters B & D, and α_{20} is an unknown scaling term. Likewise, $\alpha_{11}$is an unknown scaling term for the net dissimilarity $(d_{AD} + d_{BC})$ between letters across locations, β_{11} is the unknown scaling term for the net dissimilarity $(d_{AB} + d_{CD})$ between letters within the two bigrams and *c* is a constant. Thus, the ISI model has only 5 free parameters: $\alpha_{10}, \alpha_{20}, \alpha_{11}, \beta_{11}\ and\ c$. These parameters can be estimated by solving the matrix equation **y** = **Xb** where **y** is a 1176x1 vector of observed bigram dissimilarities, **X** is a 1176 x 5 matrix containing the net single dissimilarity of each type (CL, CR, X & W) that contributes to the total dissimilarity, and **b** is a 5 x 1 vector of unknown weights corresponding to the contribution of each type of dissimilarity (plus a constant).

The performance of the ISI model is summarized in Fig. S3F. It can be seen that, despite having only 5 free parameters compared to 85 parameters of the part-sum model, the ISI model yields comparable fits to the data (Fig. S3F).

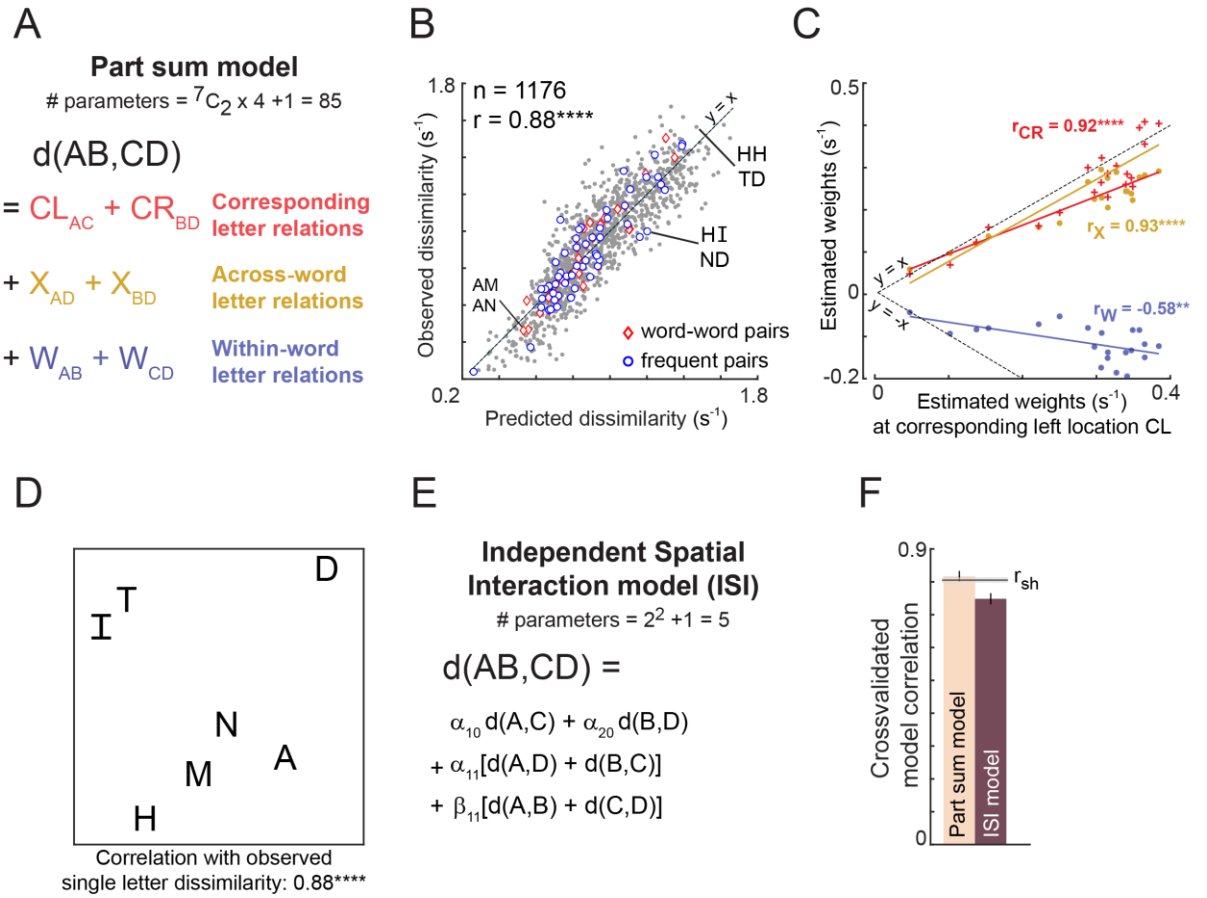


Figure S3. Predicting bigram dissimilarity using part-sum model

(A) Schematic of the part sum model. According to this model, the dissimilarity (1/RT) between bigrams 'AB' and 'CD' is written as a linear sum of dissimilarities of its corresponding part terms (AC and BD, shown in red), across part terms (AD and BC, shown in yellow), and within part terms (AB and CD, shown in blue).

(B) Correlation between the observed and predicted dissimilarities (1/seconds). Each point represents one search pair (n = 49C2 = 1176). Word-word pairs are highlighted using red diamonds, and frequent bigram pairs are highlighted using blue circles. Dotted lines represent unity slope line.

(C) Correlation between the estimated weights at corresponding location left with estimated weights at 1) corresponding location right (red), 2) across location (yellow), and 3) within location (blue). Each point represents one letter pair (n = 7C2 = 21). Dotted lines represent positive and negative unity slope line.

(D) Perceptual space of the single letter dissimilarities, that are the model coefficients of part terms at left corresponding location

(E) Schematic of the Independent Spatial Interaction model. In this model, we use the observed letter-pair dissimilarities and only estimate the weights of these letter-pair dissimilarities across different locations.

(F) Comparing part-sum and ISI model fits. Bar plots represents mean correlation coefficient between the observed and predicted dissimilarities. Error bars represent one standard deviation across 30 splits. Black horizontal line represents mean split-half correlation (r_{sh}) and the shaded error bar represents one standard deviation around the mean. (****, $p < 0.00005$, **, $p < 0.005$).

ISI model performance across all experiments

Next we asked whether the ISI model can be generalized to explain dissimilarities between longer strings. Consider two n-letter strings $u_1u_2u_3u_4 \ldots u_n$ and $v_1v_2v_3v_4 \ldots v_n$. The net dissimilarity between the two strings can be written as:

$$d(u_1u_2 \ldots u_n, v_1v_2 \ldots v_n) = \sum_{i=0}^{n}\sum_{k=0}^{n-i} \alpha_{ik}\big(d(u_i, v_{i+k}) + d(v_i, u_{i+k})\big) - \sum_{i=0}^{n}\sum_{k=1}^{n-i} \beta_{ik}\big(d(u_i, u_{i+k}) + d(v_i, v_{i+k})\big) + c$$

where α_{ik} are the unknown weights corresponding to pairs of letters across the two n-grams separated by "k" positions starting from 0, and β_{ik} are the unknown weights corresponding to pairs of letters separated by "k" positions within the two n-grams. Written in this manner, the total number of unknowns in the n-gram ISI model is n^2+1, which can be estimated using standard linear regression as before. For instance, for the 6-gram ISI model, there are $6^2+1 = 37$ free parameters.

In this manner, we fit the ISI model to all experiments. The resulting cross-validated model fits are shown together with the neural model in Figure S4. It can be seen that the ISI model performance is comparable to that of the neural model across all experiments.

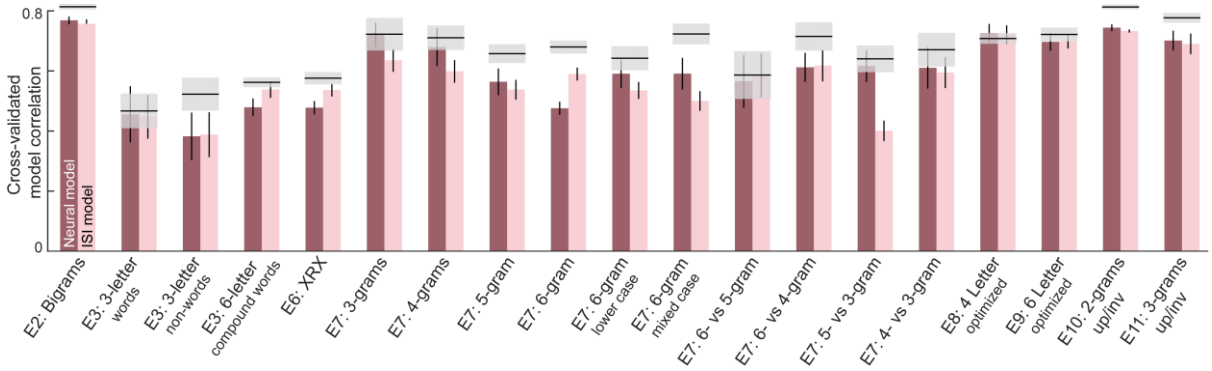


Figure S4. ISI & neural model performance across all experiments
For each experiment, we obtained a cross-validated measure of both neural and ISI model performance as follows: each time we divided the subjects randomly into two halves, and trained the neural model on one half of the subjects and tested it on the other half. This was repeated for 30 random splits. The correlation between the model predictions and the average dissimilarity from the held-out half of the data was taken to be the model fit. The correlation between the observed dissimilarity between the two random splits of subjects is then the upper bound on model performance (mean ± std shown as *gray shaded bars*).

Reducing the complexity of the ISI model

According to the ISI model, the net dissimilarity between two n-grams can be written as a weighted sum of dissimilarities between letter pairs that are varying distances apart. We wondered if the ISI model can be simplified further if there is a systematic pattern whereby these weight corresponding to a given letter pair varies systematically with letter position and distance between the letters.

To assess this possibility, we plotted model coefficients of the ISI model estimated from Experiment 7 along two dimensions. First, we asked if the contribution of letter pairs at corresponding locations in the two n-grams varies with letter position. For varying string lengths (3-, 4-, 5- and 6-letter strings) we observed a characteristic U-shaped function whereby the edge letters contribute more to the net dissimilarity compared to the middle letters (Fig. S5A). Second, we asked if model weights decrease systematically with inter-letter distance. This was indeed the case regardless of the starting letter in the pair (Fig. S5B). Finally, we note that across and within part terms are roughly equal in magnitude but opposite in sign (Fig. S3C).

The above pattern of weights in the ISI model suggest that we can make two simplifying assumptions. First, the weight of the starting letter is a U-shaped function when the inter-letter distance is zero (α_{i0}). Second, weights decrease exponentially thereafter with increasing inter-letter distance. Specifically:

$$\alpha_{i0} = ai^2 + bi + c \; for \; i = 1,2,\ldots n$$
$$\alpha_{ik} = \alpha_{i0} e^{-k/\tau} \; for \; k \geq 1$$
$$\beta_{ik} = -\alpha_{ik} \; for \; k \geq 1$$

where $a, b, c \; and \; \tau$ are the free parameters in this model. This simplified model, which we call the Spatial Interaction Decay (SID) model has only very few parameters and can be used to predict the dissimilarities between strings of arbitrary length. The model parameters are obtained using nonlinear gradient descent methods (*nlinfit* function, MATLAB).

To illustrate the performance of the SID model in comparison to the ISI model, we fit the model to 6-letter compound words (Experiment 3). To compare the two models, we plotted the ISI model terms directly estimated from the search data against the ISI model terms predicted from the SID model. This yielded a strong positive correlation (Fig. S5C). The SID model also yielded excellent fits to the data (Fig. S5D), and both models yielded comparable fits (Fig. S5E).

To evaluate this pattern across all experiments, we fit both SID and ISI models to all experiments. Here too we obtained qualitatively similar fits for the two models (Fig. S6). To confirm whether the SID model trained on one experiment can capture the variations in another, we trained the SID model on data from Experiment 9 and evaluated it on all other experiments. This too yielded largely similar but smaller predictions (Fig. S6). This decrease in model fit suggests that model parameters are somewhat dependent on the search pairs chosen.

We conclude that dissimilarities between arbitrary letter strings can be predicted using highly simplified models that operate on single letter dissimilarities and simple compositional rules.

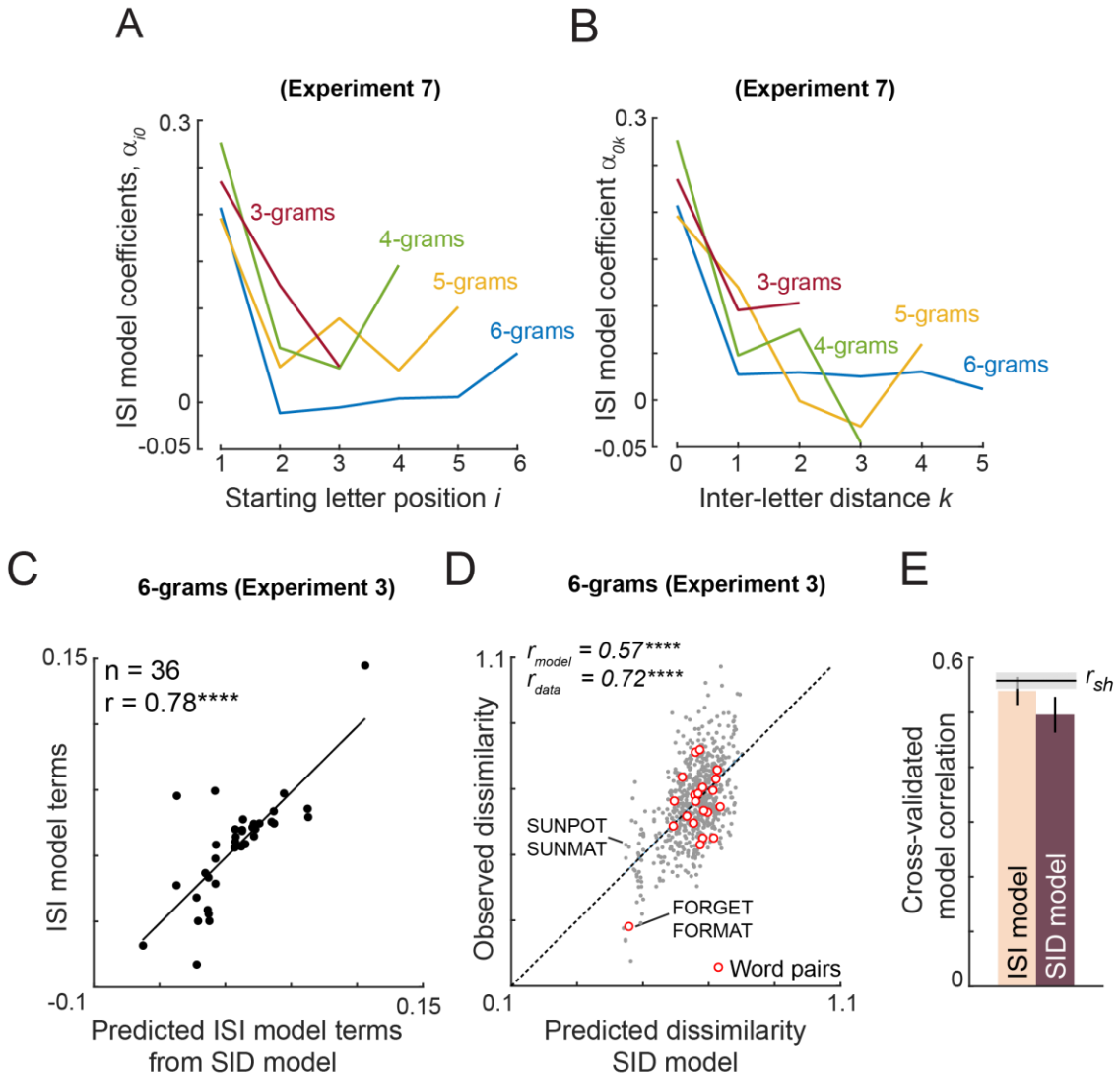


Figure S5. Reducing the ISI model

(A) ISI model coefficients α_{i0} as a function of starting letter position *i*, for Experiment 7, for varying string lengths.

(B) ISI model coefficients α_{1k} as a function of inter-letter distance *k* for Experiment 7, for varying string lengths.

(C) ISI model coefficients (both α_{ik} and β_{ik}) plotted against the predicted ISI model coefficients from the SID model. Both models are fitted to data from Experiment 3.

(D) Observed dissimilarity in Experiment 3 plotted against predicted dissimilarity from the SID model.

(E) Cross-validated model correlation for ISI & SID models.

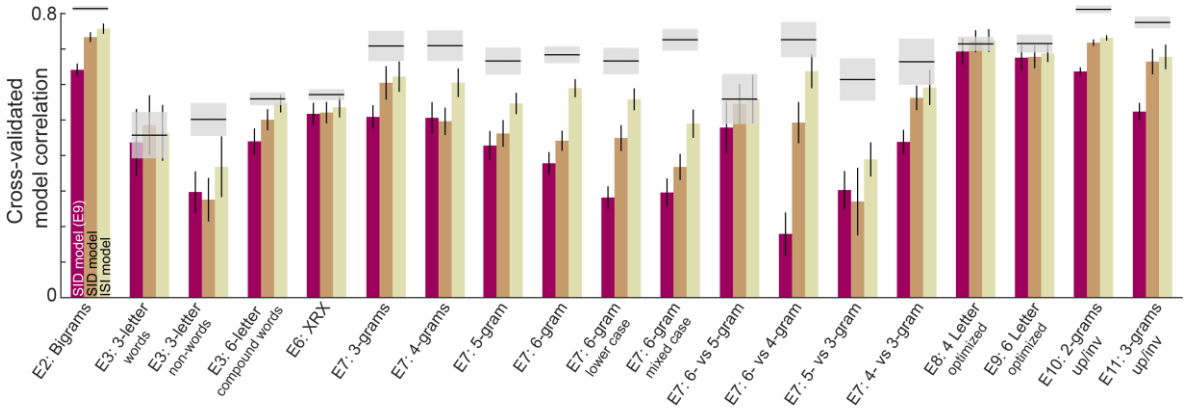


Figure S6. ISI and SID model fits across all experiments. Cross-validated model fits for the ISI and SID models across all experiments. In each case the SID and ISI models were fit on a randomly chosen half of the subjects and tested on the other half. The SID (E9) bars refer to the SID model trained on Experiment 9 and tested on data from a randomly chosen half of subjects in each experiment.

SECTION S4. UPRIGHT AND INVERTED BIGRAMS AND TRIGRAMS

It has been observed that readers are more sensitive to letter transpositions for letters of their familiar script. Since discrimination of letter transpositions in the neural model is a direct consequence of asymmetric spatial summation (main text, Fig. 3), we predicted that readers should show asymmetric spatial summation for familiar letters compared to unfamiliar letters. As a strong test of this prediction, we compared visual search performance on upright letters (which are highly familiar) with inverted letters (which are unfamiliar) across two experiments, one on bigrams and the other on trigrams.

The comparison of upright and inverted letter strings is also interesting for a second reason. If reading or familiarity with upright letters led to the formation of specialized detectors for longer strings, then we predict that the neural model (which assumes responses to be driven by single letters only) should yield worse fits for upright compared to inverted letters.

We tested the above two predictions in the following two experiments.

Experiment 10: Upright vs inverted bigrams

Methods. A total of 8 subjects (6 males, aged 24 ± 1.5 years) participated in this experiment. Six uppercase letters: A, L, N, R, S, and T were combined in all pairs to form a total of 36 stimuli. These uppercase letters were chosen because they appear very different when inverted (as opposed to letters like H that are unaffected by inversion), and were chosen to maximize the occurrence of frequent bigrams. The same stimuli were inverted to create another set of 36 stimuli. Stimuli subtended ~4° along the longer dimension. Subjects performed all possible searches among the upright letters ($^{36}C_2$ = 630 searches) with two repetitions and likewise for inverted letters. All trials were interleaved. All other details were exactly as in Experiment 2.

Results

We observed interesting differences in search difficulty depending on the nature of the bigrams. This pattern is illustrated in Fig. S7A-B. When the target and distractors consisted of repeated letters (e.g. TT among AA in Fig. S7A), search is equally easy when the array is upright or inverted. In contrast if the target and distractors are transposed versions of each other (e.g. TA among AT in Fig. S7B), search is easier in the upright array compared to when it is inverted.

To confirm that this effect is present across all such pairs, we compared observed RTs for these two types of searches between upright and inverted conditions (Fig. S7C). Response times for the AA-BB searches were comparable for upright and inverted conditions (mean ± sd of RT: 0.66 ± 0.09 s for upright, 0.67 ± 0.1 s for inverted). To assess the statistical significance of this difference, we performed an ANOVA with subject (8 levels), bigram (15 pairs) and orientation (upright vs inverted) as factors. We observed no significant difference in the response times between upright and inverted conditions for AA-BB searches ($p = 0.65$ for main effect of orientation; $p < 0.00005$ for subject and bigram factors, $p > 0.05$ for all interactions).

Next we compared transposed letter (AB-BA) searches. Here, subjects were clearly faster on the upright searches compared to inverted searches (mean ± sd of RT: 1.58 ± 0.25 s for upright, 3.12 ± 0.76 s for inverted). This difference was statistically significant ($p < 0.00005$ for main effect of orientation; $p < 0.0005$ for subject and $p <$

.05 for bigram factors, $p < 0.05$ for interactions between pairs and orientation. Other interaction effects were not significant).

To compare bigram dissimilarity between upright and inverted bigrams, we plotted one against the other. This revealed a highly significant correlation ($r = 0.80$, $p < 0.00005$; Fig. S7D). Here too it can be seen that the transposed letter searches are clearly faster when they are upright whereas the repeated letter searches show no such difference.

Thus, inversion slows down transposed letter searches but not repeated letter searches.

Explaining upright and inverted bigram dissimilarity using the neural model

We fit the neural model to both upright and inverted bigram searches using a total of 10 neurons with single letter responses derived from Experiment 1. The neural model yielded excellent fits on both upright and inverted bigrams. In both cases, the model fits approached the data consistency (Fig. S7E), implying that the model explained nearly all the explainable variance in the data.

To compare these model fits for upright vs inverted statistically, we performed a bootstrap analysis. Each time, we selected subjects with replacement and fit the neural model to the average dissimilarity computed for this random pool of subjects. Each time we calculated a normalized correlation measure that takes into account the difference in data reliability between upright and inverted trigram searches. This normalized correlation is simply the model correlation divided by the data consistency. To assess statistical significance, we calculated the fraction of times the normalized correlation in the upright samples was larger than the inverted samples. This analysis revealed significant difference in model performance between upright and inverted searches, but in the opposite direction (average model correlation: $r = 0.92$ for upright, 0.9 for inverted; fraction of upright < inverted normalized model correlation: $p = 0$). Thus, upright searches are more predictable than inverted searches using the neural model.

Next we asked whether the neural model can explain the intriguing observation that inversion affects transposed letter searches but not repeated letter searches. This is easy to explain in the neural model: The response to repeated letter bigrams such as AA is unaltered (Fig. 3B), and therefore the dissimilarity between AA and TT is unaffected by the asymmetry in spatial summation. By contrast, the dissimilarity between transposed letter pairs like AT & TA is directly driven by the asymmetry in spatial summation. We also note that the search TT among AA is much easier than the search for TA among AT, which is also explained by the neural model. This is also explained by the neural model by the fact that the response to repeated letters is the same as the response to individual letters, leaving their discrimination unaltered. By contrast transposed letters are much more similar since their neural responses are much closer (Fig. 3B).

To be sure that neural model predictions show the same pattern, we plotted the average response time predicted by the neural model for repeated letter (AA-BB) and transposed letter (AB-BA) searches. To assess the statistical significance, we performed a sign-rank test on the predicted RT. The neural model predictions were exactly as expected (Fig. S7F).

Next we analysed the model parameters in the neural model to ascertain whether the spatial summation in the neurons was indeed different for upright and inverted bigrams. To quantify the degree of asymmetry, we calculated for each neuron a spatial modulation index of the form $MI = abs(w1-w2)/(w1+w2)$ where w1 and w2 are

the estimated weights for each letter in the bigram. To avoid unnaturally large modulation indices, w1 and w2 values smaller than 0.01 were set to 0.01. The spatial modulation index for all 10 neurons for upright and inverted bigrams is shown in Fig. S7G. It can be seen that the modulation index is larger in most cases for the upright bigrams. This difference was statistically significant, as assessed using a sign-rank test on the spatial modulation indices (Fig. S7H).

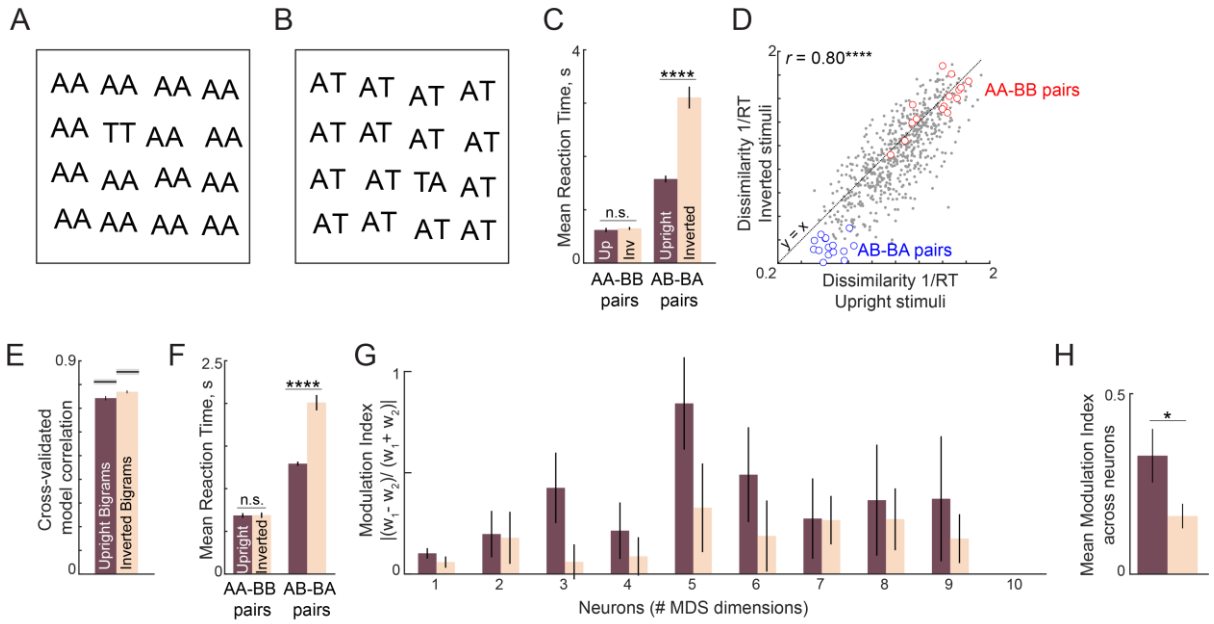


Figure S7. Neural model fits for upright and inverted bigrams

(A) Example oddball search array for a repeated letter target (TT) among identical repeated-letter distractors (AA). It can be seen that inverting this search array does not affect search difficulty.

(B) Example oddball search array for transposed letters (TA among AT). It can be seen by inverting this search array makes the search substantially more difficult.

(C) Average search times in the oddball search task for repeated-letter searches (AA-BB) and transposed letter (AB-BA) searches. Error bars represent s.e.m calculated across subjects. Asterisks represent statistical significance (**** is $p < 0.00005$), as obtained using an ANOVA on the response times with subject, bigram and orientation as factors (see text).

(D) Dissimilarity of inverted bigram pairs plotted against the dissimilarity of upright bigram pairs.

(E) Cross-validated model correlation of the neural model for upright bigrams and inverted bigrams. *Shaded gray bars* represent the upper bound achievable in each case given the consistency of the data, calculated using the split-half correlation r_{sh}.

(F) Predicted RT from the neural model for repeated letter pairs and transposed letter pairs. Asterisks denote statistical significance as obtained using a sign-rank test on the predicted RTs between upright and inverted conditions.

(G) Spatial modulation index for each neuron in the neural model for upright and inverted bigrams.

(H) Average spatial modulation index for upright and inverted bigrams. Asterisks represent statistical significance (* is $p < 0.05$) obtained using a sign-rank test on the spatial modulation index across the 10 neurons.

Comparing upright and inverted bigrams using part-sum model

The above results are based on fitting the neural model to upright and inverted bigrams but assuming a fixed set of single letter responses derived from uppercase letters. The fact that the neural model yielded excellent fits to both upright and inverted bigrams validates this assumption. Nonetheless, we wondered whether differences between upright and inverted bigram searches can be explained solely by different letter representations or by differences in letter interactions.

To investigate this possibility, we fit the part-sum model to upright and inverted bigram searches (Fig. S8A). The part-sum model also yielded equivalent fits to both upright and inverted searches (Fig. S8B). If model predictions were similar, we reasoned that the difference between upright and inverted searches must be explained by differences in model parameters. To this end, we compared the estimated letter dissimilarities of each type (CL, CR, X and W) in the upright and inverted searches (Fig. S8C). Model terms were comparable in magnitude for the CL terms, but were systematically weaker for both CR, X and W terms for inverted compared to upright searches (Fig. S8C). However in all cases, the recovered letter dissimilarities were correlated between upright and inverted conditions (correlation between upright and inverted model terms: r = 0.93, 0.91, 0.97 & 0.87 for CL, CR, X & W terms; all correlations $p < 0.00005$).

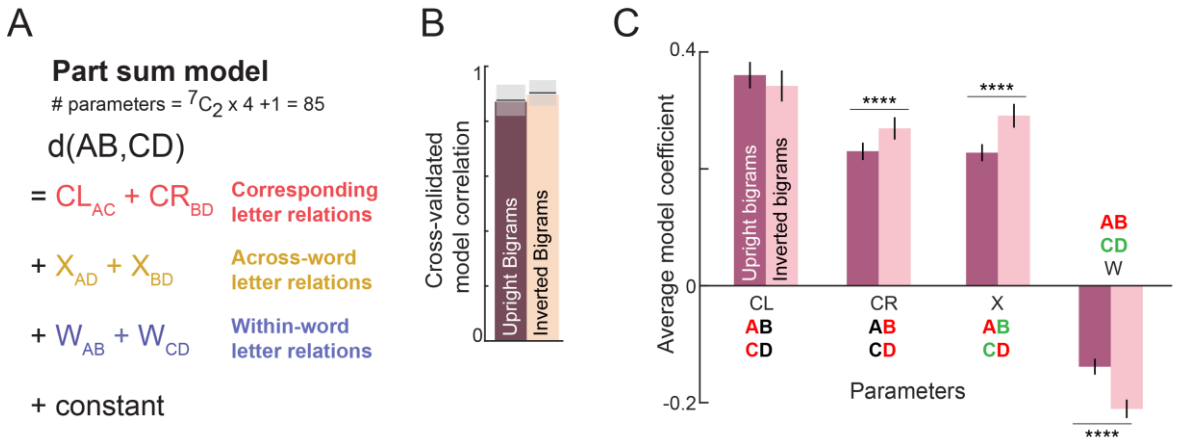


Figure S8. Part-sum model fits for upright and inverted bigrams

(A) Schematic of the part-sum model, in which the net dissimilarity between two bigrams is given as a linear sum of letter dissimilarities at corresponding locations (CL & CR), across-bigrams (X) and within-bigrams (W).

(B) Cross-validated model correlation of the part sum model for upright and inverted bigrams.

(C) Average model coefficients (mean ± sem) of each type for upright and inverted bigrams. Asterisks denote statistical significance (**** is $p < 0.00005$) obtained on a sign-rank test comparing 15 letter dissimilarities between upright and inverted conditions).

Experiment 11: Upright and inverted trigrams

Here, we asked whether the above results would extend to trigrams. We tested two predictions. First, we predicted greater spatial modulation for upright compared to inverted trigrams, on the premise that better discrimination of trigram transpositions should be driven by asymmetric spatial summation. Second, if repeated viewing of a trigram or word led to the formation of specialized trigram detectors, then the neural model (which is based only on knowledge of single letters) should produce larger errors compared to other trigrams. We tested this prediction by comparing model fits for searches involving frequent trigrams and words compared to other searches.

Methods. A total of 9 subjects (6 females, aged 24.5 ± 2.3 years) participated in the experiment. Six uppercase letters: A, G, N, R, T and Y were combined in all possible 3-letter combination to form a total of 216 stimuli. These letters were chosen to include as many three-letter words as possible. In all, 15 three-letter words could be created using these letters (ANT, ANY, ART, GAG, GAY, NAG, NAY, RAG, RAN, RAT, RAY, TAG, TAN, TAR, and TRY).

Since the total number of possible search pairs is large (${}^{216}C_2$ = 23,220 pairs), we chose 500 search pairs such that the regression matrix of the part-sum model had full rank i.e. all the model parameters can be estimated reliably using linear regression. These 500 searches consisted of 368 random search pairs, 105 (${}^{15}C_2$) word-word pairs, 15 (${}^{3!}C_2$) transposed pairs of nonword comprised of letters G,N, and R. Further, another set of 15 (${}^{3!}C_2$) transposed pairs were created using the word TAR. The search pairs formed using the words TAR, ART and RAT were presented only once (although they were counted as both word-word pairs and transposed pairs in the main analysis).

Subjects performed the same searches using upright and inverted trigrams. Stimuli subtended ~5° along the longer dimension. All subjects completed 2000 correct trials (500 searches x 2 orientations x 2 repetitions). All other details were identical to Experiment 2.

Results

An example oddball array in the trigram experiment is shown in Figure S9A. Note that it is no longer meaningful to compare repeated letter trigrams (AAA-BBB) with transposed trigrams (ABC-BCA) because the repeated letter pairs contain two unique letters whereas the transposed trigrams contain three unique letters. Subjects were highly consistent in both upright and inverted searches (split-half correlation between even and odd- subjects: r = 0.76 & 0.80, p < 0.00005). Upright and inverted dissimilarities were highly correlated (r = 0.80, p < 0.00005; Fig. S9B), although upright searches had higher dissimilarity compared to inverted searches.

Next we asked whether the neural model can predict dissimilarities between upright trigrams. As before, neural model predictions were highly correlated with the observed data (r = 0.79, p < 0.00005; Fig. S9C) and this model fit approached the data consistency itself (r_{data} = 0.88). Model fits were actually lower for transposed pairs compared to word-word pairs and other pairs (mean ± sd error: 0.1 ± 0.08 for word pairs; 0.07 ± 0.06 for transposed pairs; 0.11 ± 0.08 for other pairs; p = 0.02, rank-sum test). The neural model was also able to predict dissimilarities between various trigram transpositions (r = 0.69, p < 0.00005; Fig. S9C). Thus, trigram dissimilarities can be predicted by the neural model regardless of word status or trigram frequency.

We then compared model fits for upright and inverted bigrams. In both cases, the neural model predictions (r = 0.78 & 0.73 for upright and inverted) were close to

the consistency of the data (r_{data} = 0.85 & 0.78; Fig. S9D). To compare these model fits for upright vs inverted statistically, we performed a bootstrap analysis as before (Experiment 10). This analysis revealed no significant difference in model performance between upright and inverted searches (fraction of upright < inverted normalized model correlation: p = 0.07).

Finally we asked whether the spatial summation weights of the neural model were systematically different between upright and inverted trigrams. Since there are three spatial modulation weights for each neuron, we calculated the spatial modulation index for all possible pairs of weights (Fig. S9 E,F,G). The spatial modulation ratio was larger for upright compared to inverted trigrams in two of the three pairs, and this difference attained statistical significance for the first and third letters in the trigram (Fig. S9F). We conclude that the spatial modulation is stronger for upright compared to inverted trigrams.

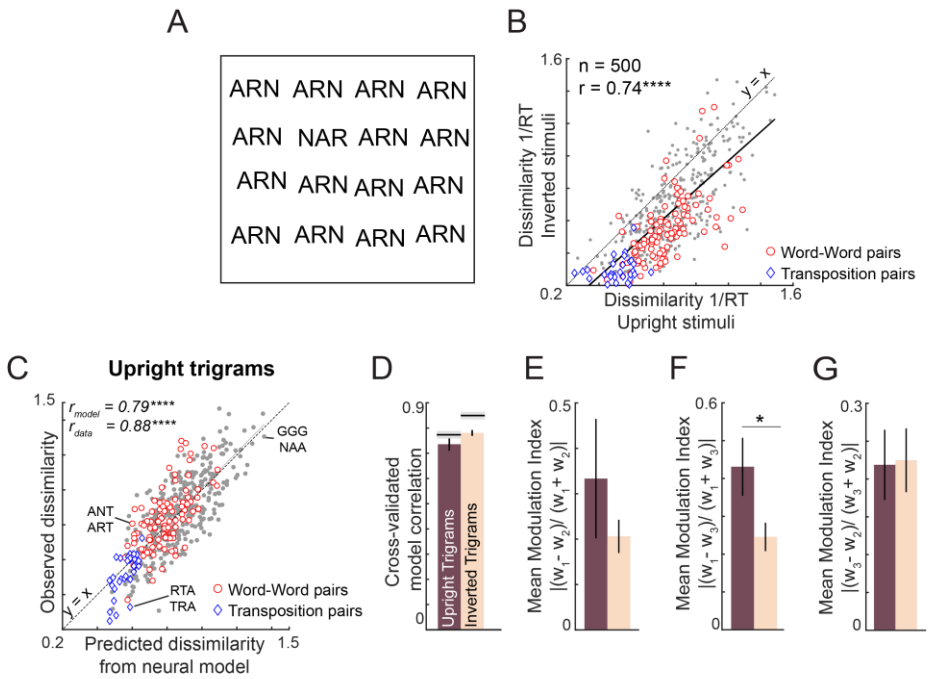


Figure S9. Neural model fits for upright and inverted trigrams

(A) Example trigram search array containing letter transpositions, with oddball target (NAR) among distractors (ARN). It can be seen that this search is substantially harder when inverted compared to upright.

(B) Dissimilarity for inverted trigram searches (1/RT) plotted against dissimilarity for upright trigram searches for word-word pairs (red circles, n = 105), transposed letter pairs (blue diamonds, n = 30).

(C) Observed dissimilarity for upright trigrams plotted against the predicted dissimilarity from the neural model with symbol conventions as in (B).

(D) Cross-validated neural model correlation for upright and inverted trigrams.

(E) Average spatial modulation index (across 10 neurons) for the first and second letters in the trigram.

(F) Same as (E) but for the first and third letters.

(G) Same as (E) but for the second and third letters.

Comparing upright and inverted bigrams using part-sum model

The above results are based on the assumption that the neural model is driven by a fixed set of single letter responses derived from uppercase letters. Although the neural model fits validate this assumption, we nonetheless tested this assumption by recovering the underlying letter dissimilarities using the part-sum model.

The part sum model applied to trigrams is depicted in Fig. S10A. In this model, the net dissimilarity between two trigrams can be written as a sum of single letter dissimilarities at every possible pair of locations. These locations are grouped as corresponding letters at left (C1), middle (C2) and right (C3) locations, letters across trigrams that are one letter apart starting from the left letter (XN1) or the middle letter (XN2), letters across trigrams that are two letters apart (XF), letters within each trigram that are one letter apart starting from the left letter (WN1) or middle letter (WN2), and letters within each trigram that are two letters apart (WF). Thus the full part-sum model has 9 groups of letter dissimilarities (C1, C2, C3, XN1, XN2, XF, WN1, WN2, WF) each having ${}^{6}C_{2}$ = 15 unknown single letter dissimilarities. Together with a constant term, this part-sum model has 9 x 15 + 1 = 136 free parameters. Since we have 500 searches each for upright and inverted trigrams, the part-sum model can be fit to this data to estimate these free parameters using standard linear regression.

Cross-validated model fits for the part-sum model are shown in Fig. S10B. It can be seen that the part-sum model explains nearly all the explainable variance in the data for both upright and inverted trigrams (Fig. S10B). This in turn means that differences between upright and inverted trigrams can be explained using differences in model parameters. This was indeed the case: on plotting the strength of model terms of each type it was clear that 7 of the 9 types of model terms (C1, C2, C3, XN2, XF, WN2, WF) were systematically larger for upright trigrams compared to inverted trigrams (Fig. S10C). Finally we confirmed that model terms for upright and inverted trigrams were highly correlated (correlation between upright and inverted model terms, averaged across 9 model term types: r = 0.65 ± 0.1, p < 0.05 in all cases).

We conclude that upright and inverted trigram searches can be explained using the part-sum model driven by a common single letter representation.

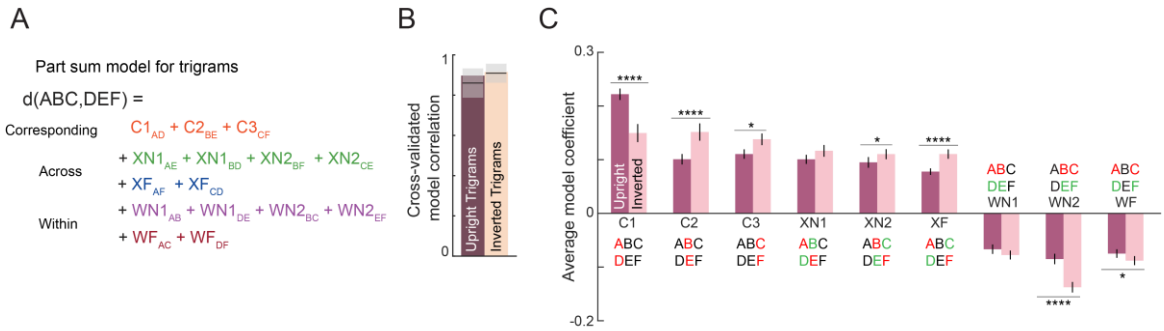


Figure S10. Part-sum model fits for upright and inverted trigrams

(A) Schematic of part-sum model for trigrams.

(B) Cross-validated model correlation of part-sum model for upright and inverted trigrams.

(C) Average model coefficient (averaged across ${}^{6}C_{2}$ = 15 terms) of each type for upright and inverted trigrams. Asterisks indicate statistical significance (* is p < 0.05, ** is p < 0.005, etc) calculated using a sign-rank test comparing the upright and inverted model terms.

SECTION S5. ADDITIONAL ANALYSIS FOR EXPERIMENT 3

Compound word stimulus set

The full set of compound words used in Experiment 3 are shown in Fig. S11. It can be seen that there are seven valid words, whereas the other compound words are pseudowords that carry no meaning.

	ANY	FOR	TAR	KEY	SUN	TEA
ONE	ANYONE	ONEFOR	ONETAR	KEYONE	ONESUN	TEAONE
MAT	MATANY	FORMAT	MATTAR	MATKEY	SUNMAT	TEAMAT
GET	GETANY	FORGET	TARGET	KEYGET	GETSUN	GETTEA
PAD	PADANY	FORPAD	TARPAD	KEYPAD	PADSUN	PADTEA
DAY	ANYDAY	DAYFOR	TARDAY	DAYKEY	SUNDAY	DAYTEA
POT	ANYPOT	POTFOR	POTTAR	POTKEY	SUNPOT	TEAPOT

Figure S11. Full stimulus set for Experiment 3.
The left and the right 3 letters words were combined to form a 6 letter string. The strings that formed compound words are highlighted in red.

Three-letter word and nonword dissimilarities

To investigate whether the neural model can predict dissimilarities between three-letter words and non-words, we fit a separate neural model with 6 neurons as before to the word and non-word dissimilarities. If frequent viewing of words led to the formation of specialized word detectors, the neural model would show worse model fits compared to nonwords. However, we observed no such pattern: the neural model fits were equivalent for words (r = 0.69, p < 0.00005; Fig. S12) and nonwords (r = 0.57, p < 0.00005; Fig. S12) – and these fits approached the respective data consistencies (r_{data} = 0.67 for words, 0.68 for nonwords). We conclude that three-letter string dissimilarities can be predicted by the neural model regardless of word status.

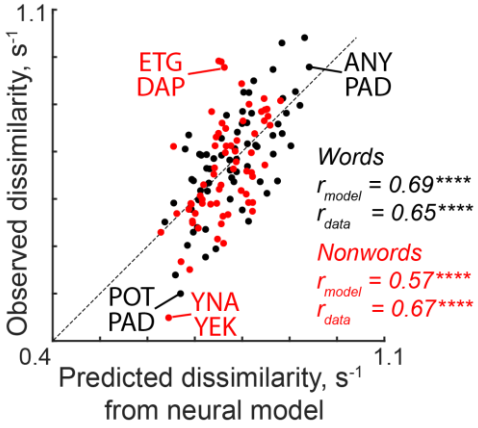


Figure S12. Neural model prediction of 3-letter word & nonword dissimilarities.
Observed dissimilarities for words (*black*) and nonwords (*red*) plotted against neural model predictions.

Spatial summation weights of each neuron

To investigate the spatial summation weights for each neuron, we plotted the estimated spatial summation weights separately (Fig. S13). It can be seen that spatial summation is heterogeneous across neurons, but the spatial summation of the first neuron follows the characteristic U-shaped curve observed in studies of reading.

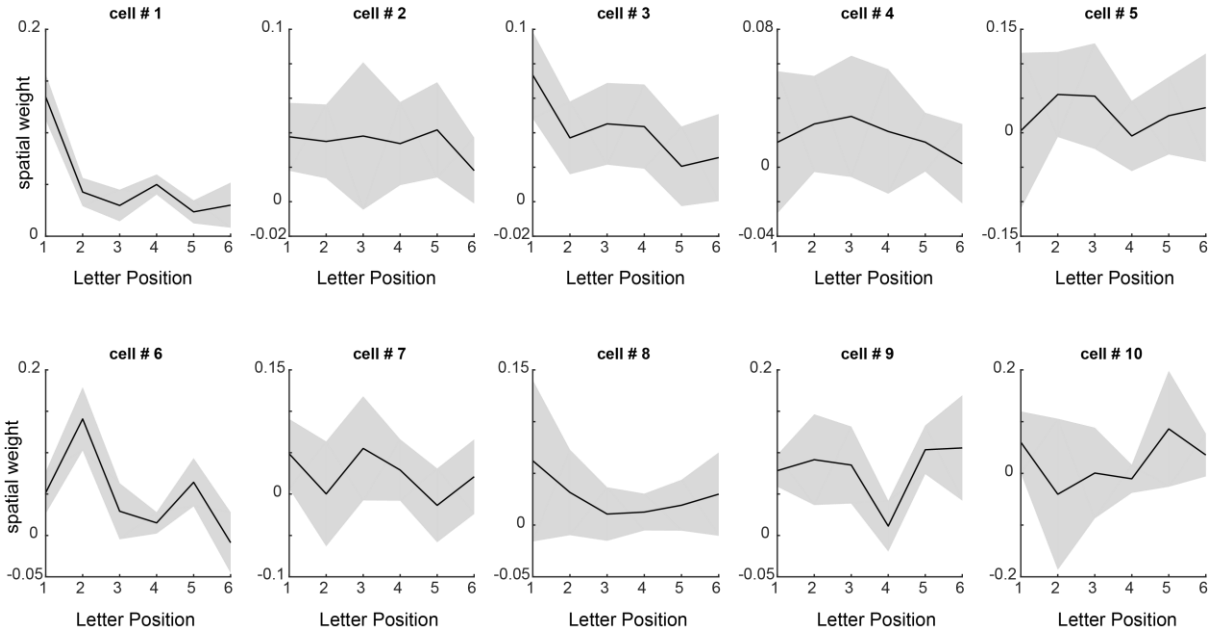


Figure S13. Spatial summation weights for each neuron. Estimated spatial summation weights (mean ± std across many random starting points of the neural model fits) for each neuron in the neural model.

SECTION S6. ADDITIONAL ANALYSES FOR EXPERIMENT 5

Nonword design

Nonwords in the lexical task were chosen according to the table below.

	Variations of word ABCDE	4 letters	5 letters	6 letters	Total
1)	*Edge transpositions*: BACDE or ABCED	15	15	20	50
2)	*Middle transposition*: ACBDE or ABDCE	15	15	20	50
3)	*2 step edge transposition*: CBADE or ABEDC	0	20	30	50
4)	*2 step middle transposition*: ADCBE	0	20	30	50
5)	*Random transposition*: CDABE, ACDBE, etc.	25	35	40	100
6)	*Edge Substitution*: MZCDE or ABCMZ	15	15	20	50
7)	*Middle Substitution*: ABMZE	15	15	20	50
8)	*Random substitution and permutation*: MACZE, AMDEZ, etc.	15	15	20	50
	Total	100	150	200	450

Table S2: Non-word stimuli in lexical decision task (Experiment 5).

Prediction of word response times using lexical factors

We asked whether response times for words can be predicted using lexical factors. To this end, we calculated a number of lexical factors for each word: its overall frequency, the number of orthographic neighbors, the average frequency of all bigrams in the word, and average frequency of all letters in the word. We then asked whether response times for words can be predicted using each of these factors, or a linear combination of these factors. The results are shown in Fig. S14 for 4, 5 and 6-letter words. In all cases, the overall word frequency was the single largest predictor of response times (Fig. S14 A,B,C). The performance of the combined model rarely exceeded the response of the word frequency model (fraction of bootstrap splits in which the combined model was worse: $p = 0.31$, 0.23 and 0.56 for 4, 5 and 6 letters). We conclude therefore that word frequency was the main driver of word response times.

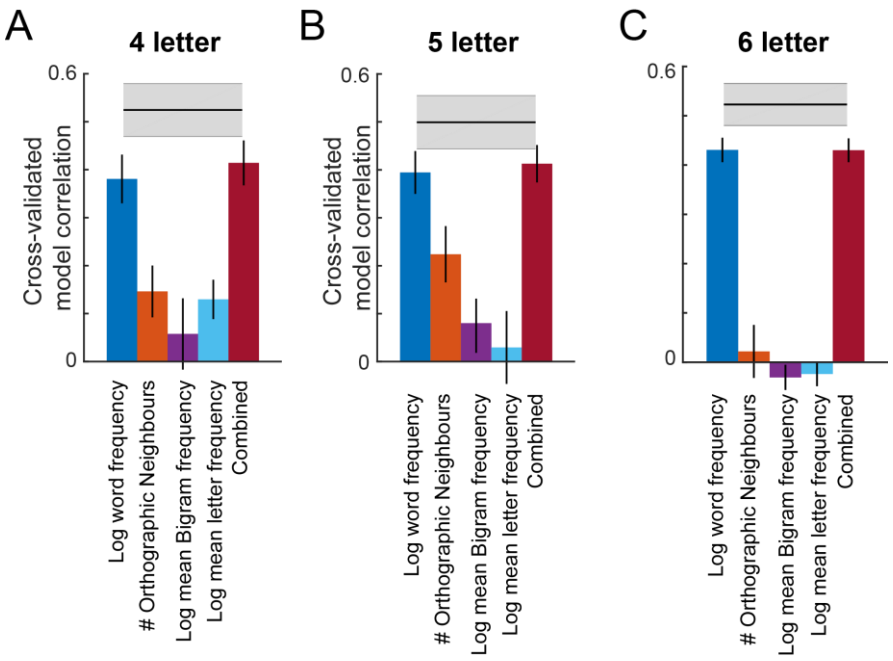


Figure S14. Prediction of word response times in the lexical task.

(A) Model correlation between observed and predicted word response times on 4-letter words for various models: word frequency alone (*blue*), number of orthographic neighbours (*orange*), mean bigram frequency (*purple*), mean letter frequency (*cyan*) and combined model containing all these factors (*red*). Shaded error bars indicate mean ± sd of the correlation across multiple splits of the observed data.

(B) Same as (A) but for 5-letter words.

(C) Same as (A) but for 6-letter words.

SUPPLEMENTARY REFERENCES

1. Vighneshvel T, Arun SP (2013) Does linear separability really matter? Complex visual search is explained by simple search. *J Vis* 13(11):1–24.
2. Ratan Murty NA, Arun SP (2015) Dynamics of 3D view invariance in monkey inferotemporal cortex. *J Neurophysiol* 113(7):2180–94.
3. Mueller ST, Weidemann CT (2012) Alphabetic letter identification: Effects of perceivability, similarity, and bias. *Acta Psychol (Amst)* 139(1):19–37.
4. Simpson IC, Mousikou P, Montoya JM, Defior S (2013) A letter visual-similarity matrix for Latin-based alphabets. *Behav Res Methods* 45(2):431–9.
5. Pramod RT, Arun SP (2016) Object attributes combine additively in visual search. *J Vis* 16(5):8.
6. Duncan.J, Humphrey GW, Duncan J, Humphreys GW (1989) Visual search and stimulus similarity. *Psychol Rev* 96(3):433–458.