Holographic String Encoding

Thomas Hannagan, Emmanuel Dupoux, Anne Christophe

Laboratoire de Sciences Cognitives et Psycholinguistique, EHESS/CNRS/DEC-ENS

Abstract

In this article, we apply a special case of holographic representations to letter position coding. We translate different well-known schemes into this format, which uses distributed representations and supports constituent structure. We show that in addition to these brain-like characteristics, performances on a standard benchmark of behavioral effects are improved in the holographic format relative to the standard localist one. This notably occurs because of emerging properties in holographic codes, like transposition and edge effects, for which we give formal demonstrations. Finally, we outline the limits of the approach as well as its possible future extensions.

Keywords: Letter position coding; Distributed representations; Holographic memories

1. Introduction

Research on letter position encoding has been through a remarkable expansion in recent years. At the origin of this are numerous findings concerning the flexibility of letter string representations, as well as the growing realization of their importance in the computational study of lexical access. Nevertheless, how the brain encodes visual words is still a subject of vivid debate. Obviously, a stable code must be used if we are to retrieve words successfully, and order must be encoded because we can distinguish between anagrams like ''relating'' and ''triangle.'' But researchers disagree on exactly how this string encoding step[1] is achieved.

There is currently a great variety of proposals in the literature, but it is possible to classify them according to the scheme and code they use. Indeed, current proposals fall into two classes of schemes: relative and absolute, and each of them can be implemented using two kinds of codes: localist or distributed.

We will see that all schemes but one have been implemented using localist codes. In this article, we explore a special kind of distributed coding, based on holographic representations. We show that well-known schemes behave differently depending on the code they use, holographic ones performing generally better on a benchmark of behavioral effects. Finally, we exploit the compositional power of holographic representations to study a new scheme in the spirit of the local combination detector model (LCD; Dehaene, Cohen, Sigman, & Vinckier, 2005), a recent and biologically constrained proposal for string encoding.

1.1. Absolute versus relative schemes

A string encoding *scheme* is an abstract specification of the basic entities and relationships by which order will be achieved in the string representation.

1.1.1. Absolute position schemes

Absolute schemes hold that for any given entity in a string, its position is marked by some coordinates in a common reference frame. The basic entity in all current absolute schemes is the letter, and the most used origin for the reference frame is the first letter. Examples of this approach include slot coding (Coltheart, Rastle, Perry, Langdon, & Ziegler, 2001; McClelland & Rumelhart, 1981), double-anchoring (Jacobs, Rey, Ziegler, & Grainger, 1998), the overlap model (Gomez, Ratcliff, and Perea, 2008), or Spatial Coding (Davis, in press).

A scheme should be flexible if it is to bear any resemblance to human performances, and absolute schemes can be more or less flexible depending on the reference frame. The less flexible, but still the most widespread, is slot coding, in which position coordinates are perfectly determined and originate at the beginning of the word. This amounts to feeding letters in a positional array, implying that as few as a single letter shift will displace all elements and minimize similarity (the so-called alignment problem; Davis, 1999). Building on slot coding, some variants like double-anchoring introduce more flexibility by using two origins (first and last letters) instead of one (Fischer-Baum, McCloskey, & Rapp, 2010; Jacobs et al., 1998). Many absolute schemes also choose to feature uncertainty in position assignment. This approach is best illustrated in the overlap model (Gomez et al., 2008), where the position of a letter in the frame is not marked by a constant coordinate, but rather by a probability density function centered on this coordinate. Finally, the most flexible absolute scheme is Spatial Coding (Davis, in press), where letters are assigned monotonic position values as they progress in the string. The flexibility comes partly from exploiting some uncertainty in position assignment, but mostly from the similarity measure between two strings, which assesses how consistently matching letters are shifted in both strings.

1.1.2. Relative position (RP) schemes

The second approach conceives of letter order in relative terms. Here, the information concerning the position of single letters is utterly irrelevant, and there is no common coordinate frame. Rather, what is important is the relative order in which two or more (possibly distant) letters appear in the string. That is, a string code is best thought of as a set of activated *n*-gram entities. Among the tenants of this approach figure Wickelcoding (Seidenberg

& McClelland, 1989; Wickelgren, 1969), Blirnet (Mozer, 1991), and open-bigram schemes (Grainger & Van Heuven, 2003; Whitney & Berndt, 1999).

Relative schemes vary as to the size and contiguity properties an entity should possess. Wickelcoding and Blirnet both use letter trigrams, but whereas the first imposes complete contiguity upon their constituent letters, the second allows for a one-letter gap to occur. Open-bigram schemes restrict the entity to the bigram, but enlarge the authorized distance between letters: up to two intervening letters are allowed in constrained open-bigrams (COB; Grainger & Van Heuven, 2003) and Seriol (Whitney & Berndt, 1999), while no restriction is imposed in unconstrained open-bigrams (UOB). More recently, an overlapping open-bigram scheme was proposed in which contiguity is reinstated upon entities, but position uncertainty takes place at lower levels; consequently, activation of noncontiguous bigrams is still achieved, albeit in a different way (Grainger, Granier, Farioli, Van Assche, & Van Heuven, 2006).

Note that we have used the terms *absolute* and *relative* slightly differently from that in the literature. The term ''absolute'' in the context of letter position coding has been applied to any scheme using a fixed coordinate frame, with a single origin and no uncertainty. The term ''relative'' was then usually defined by opposition, and hence includes schemes with more than one origin. Accordingly, double-anchoring schemes were first dubbed relative on the grounds that position is encoded with respect to two anchors, and Spatial Coding has been referred to as relative because match calculations are based on the relative alignment between letter positions in both strings, rather than on the number of letters that match at any absolute position. But we feel this terminology could profit from a more traditional use of the absolute/relative dichotomy. In the classical meaning, any scheme that encodes order through a coordinate system, using the distance of entities to one or more common origin(s), should be referred to as absolute, whereas any scheme that purports to encode order directly by the relationship between entities and bypassing any origin, should be referred to as relative. This terminology is orthogonal to flexibility; indeed, relative schemes can be quite rigid (e.g., Wickelcoding) and absolute schemes can be very flexible (e.g., Spatial Coding).

1.1.3. A recent hierarchical proposal: Local combination detector

Although it has not yet been implemented, a recent string encoding proposal privileging biological plausibility and constrained by brain imaging data has recently appeared (Dehaene et al., 2004, 2005; Vinckier et al., 2007).

LCD holds that hierarchical feature extraction from the stimulus results in invariance for size and font in the left fusiform gyrus, and more precisely in the posterior part of the so-called visual word form area (Cohen & Dehaene, 2004; Cohen et al., 2000). From there, string encoding is best described as involving three levels of units with increasing receptive fields, culminating in the anterior part of the region. The model distinguishes between letter, bigram, and quadrigram detectors, which would be segregated into distinct cortical stripes—although all types of units could also be present at each level in different proportions (Vinckier et al., 2007). Based on evidence from behavioral and brain studies (Dehaene et al., 2004; Peressotti & Grainger, 1999), the model assumes an absolute scheme at the

letter level, but the highest level in the hierarchy uses open-quadrigrams and thus a scheme built from the LCD model would ultimately be a relative one.

1.2. Localist and distributed codes

Once a scheme has been decided upon, its actual implementation requires to define a *code*. Codes are primarily concerned with units, and how they relate to the entities that have been specified in the scheme. One should distinguish between two different kinds of relationships between units and entities: one-to-one relationships, which produce localist codes,[2] and many-to-many relationships, which produce distributed codes (Plaut & McClelland, 2010; Rumelhart & McClelland, 1986).

1.2.1. Localist codes

A localist code describes a one-to-one mapping between entities and units. For example, consider encoding the word ''CABS'': localist Wickelcoding could use a binary vector where each component stands for a single letter triplet. The code would then assign ones only to these four units corresponding to CA, CAB, ABS, and BS and zeros to all others. This code is localist in the strongest sense: A single bit would map to exactly one element and hence flipping a bit from 1 to 0 would entirely and exclusively destroy the information about the corresponding triplet.

The localist approach has a long history in cognitive science. As each unit represents one entity, localist codes have appeared more compatible with so-called classical cognitive frameworks in which rules operate on symbols (Chomsky, 2002; Fodor, 1983; Pinker, 1999; Turing, 1950). Localist coding has also been used in many successful computational models coming from various areas of cognition, and especially in visual word recognition (Coltheart et al., 2001; Grainger & Jacobs, 1996; McClelland & Rumelhart, 1981; Norris, 2006). At the neural level, the approach concords with the so-called grandmother cell hypothesis, which states that single cells can be dedicated to a single arbitrarily complex object irrespective of its size, orientation, or color (Gross, 2002). As noted by Gross, the original grandmother cell hypothesis advocated a weaker form of localist representations, in which many neurons (the number 18,000 has been advanced) would be tuned to the same entity, and indeed some localist proponents have embraced this many-to-one *redundant coding* (Bowers, 2009).

1.2.2. Distributed codes

Alternatively, one can use a code in which many units participate in representing any given entity, and many entities use each unit. That is, no unit can be singled out as uniquely representing one stimulus—or any class, or any given feature of it—and instead information about any given entity is spread over the activity of many units. At one extreme, this many-to-many mapping produces a dense distributed code, each entity activating in average half of the units. At the other extreme, we find sparsely distributed codes, in which an entity uses only a few units and each unit codes for a few entities. For many years, the only distributed string code in the literature implemented a Wickelcoding scheme, and had been proposed in

the context of the Parallel Distributed Processing (PDP) model (Seidenberg & McClelland, 1989). It consisted of 400 binary units, each responding to 1,000 randomly chosen letter triplets. This so-called coarse coding approach is distributed because considering only a unit cannot tell us which one of the 1,000 triplets it is activated for, yet taken together the 400 units do define a virtually unique correspondence.[3] Recently, however, Dandurand, Grainger, and Dufau (2010) trained a backpropagation network on a location invariant word recognition task, and it has been shown that this network implements a densely distributed version of the overlap scheme (Hannagan, Dandurand, & Grainger, in press).

Distributed codes are slightly more recent than localist ones but also have a strong pedigree. Their study was precipitated by the discovery of the backpropagation learning algorithm (Rumelhart, Hinton, & Williams, 1986), which usually produces distributed codes whose sparseness depends on the network structure and the task at hand (Plaut & McClelland, 2010). These codes are also intimately tied with high-stake theoretical issues in cognitive science, contrasting a vision of human cognition as emerging from associations between (often distributed) representations, rather than as systems of rules and symbols (McClelland, Patterson, Pinker, & Ullman, 2002). However, this distinction was blurred in the 1990s by the introduction of such techniques as Tensor Product representations (Smolensky, 1990) and later by Holographic representations, which instantiate what has been described as an Integrated Connectionist Symbolic framework (Smolensky, Legendre, & Miyata, 1992).

1.3. The localist versus distributed debate

As this article illustrates, the choice of a representational format has profound impacts at the computational level: Schemes do not make the same predictions depending on the code they use. In consequence, one should assess which of the distributed or localist approach, if any, is currently favored by the evidence.

Nowadays, the commitment to distributed representations is widespread among both neuroscientists and computational modelers, in part because of the success of the PDP framework (Rumelhart & McClelland, 1986), and also because they have been reported in a vast number of brain regions and in a variety of forms—see Abbott and Sejnowski (1999) and Felleman and Van Essen (1991) for good surveys. As regards string encoding, it is also worth noting here that the LCD proposal, which may currently have the strongest claims to biological plausibility, postulates distributed representations.

Recently, however, Bowers (2009) argued for a localist vision in which units are redundant (i.e., many dedicated localist units code for any given entity), activation spreads between them (i.e., each dedicated unit can possibly also show some low level of activity for a related stimuli), and units are limited to coding for individual words, faces, and objects (Bowers, 2009). Bowers points to human neurophysiological recordings showing rare cells with highly selective responses to a given face, and negligible or flat responses to all others.

It has been pointed out that reports of highly tuned cells can hardly count as evidence for localist coding since one can neither test exhaustively all other cells for any given stimulus, nor the presumed localist cell exhaustively for all possible stimuli (Plaut & McClelland, 2010). Nevertheless, Bayesian estimates of sparseness can and have been derived for

medial temporal lobe cells (Waydo, Kraskov, Quiroga, Fried, & Koch, 2006), assuming that on the order of 10^5 items are stored in this area—where coding is generally believed to be sparser than elsewhere in the brain—and these estimates turned out to be unequivocally in favor of sparse distributed coding. Crucially, the following claim from Bowers is incorrect:

> Waydo et al.'s (2006) calculations are just as consistent with the conclusion that there are roughly 50–150 redundant Jennifer Aniston cells in the hippocampus. (p. 245)

Waydo et al.'s (2006) study was agnostic with respect to redundant coding and explicitly concludes that rather than 50–150, on the order of several million medial temporal (MT) neurons should respond to any given complex stimulus (Jennifer Aniston included), and each cell is likely to respond to a hundred of different stimuli. In fact such ''multiplex'' cells, to use Plaut and McClelland's (2010) proposed terminology, have been directly observed by Waydo et al. (2006) (personal communication).

As for computational efficiency, distributed codes have well-known computational qualities: They are robust to noise, have a large representational power, and support graded similarities. In distributed codes, randomly removing n units does not lead to the total deletion of n representations, but rather to the gradual loss of information in all representations. This would seem especially useful considering the different sources of noise—environmental or neural—in spite of which the brain has to operate (Knoblauch & Palm, 2005). Redundant localist coding would indeed provide a similar advantage, were it not already disproved by the above mentioned estimates. Distributed coding also provides a significant gain in representational power. For instance, n binary units can encode at most n local representations (without any redundancy), but this number is a lower bound when using distributed representations—and can reach 2^n with dense coding. Indeed, because firing a neuron takes energy, there is likely to be a trade-off between representational efficiency and metabolic cost. Brain regions supporting highly regular associations would be more likely to settle on dense representations, whereas those supporting highly irregular associations would settle on sparser ones, although not to the localist extreme. Moreover, while distributed representations are susceptible to result in ''catastrophic interference'' of new memories on old ones, it is known that the independently motivated introduction of constraints, such as interleaved learning or assumptions, such as pseudo-rehearsal can circumvent the issue (Ellis & Lambon Ralph, 2000; McClelland, McNaughton, & O'Reilly, 1995).

Finally, in localist coding, the question arises as to which entities or class of entities a unit should stand for, this especially considering that class distinctions can be entirely context dependent (Plaut & McClelland, 2010). This is intimately tied to the problem of defining similarities between representations, a problem which is solved in distributed representations where graded similarities can be defined as the overlap between patterns—for instance, by computing the euclidean distance between activation patterns.

In light of the available evidence, we think that the localist approach is not currently supported by physiology, that it runs into the fundamental issue of how to determine what a unit codes for, and that it severely limits representational power especially if one acknowledges that redundant coding is required to overcome noise issues—which is not compatible with

current estimates. On the other hand, distributed representations have been abundantly found in the brain, and they allow for a much larger, graded, and robust spectrum of representations to be coded by any given population of unit.

1.4. Constituent structure and the dispersion problem

As we have seen, there are behavioral and biological reasons to explore the possibility that letter combinations are involved in the string encoding process, as proposed for instance in LCD, open-bigrams, or Wickelcoding. In these schemes by definition a given letter combination is active only if its constituents are present in the string in the correct order.

There is, however, a consequence of this way to represent letter order: the possible loss of information pertaining to constituent structure. Under a localist open-bigram implementation, for instance, bigram 12 is equally similar to bigrams 1d or 21, with which it shares common constituents,[4] than to bigram dd, with which it does not—see also Davis and Bowers (2006), for the same remark on localist Wickelcoding. This is what mathematicians refer to as a topologically discrete space: the (normalized) distance between two localist representations is always one (or zero iff identical). As *n*-gram units increase in size, this lack of sensitivity to constituent structure can become problematic for the string encoding approach, implying for instance that a localist open-quadrigram code would assign no similarity whatsoever between *any* four letter strings, in flat contradiction with experimental results (Humphreys, Evett, & Quinlan, 1990).

While this issue does not arise in absolute codes because similarities are computed at the level of letters, it is potentially serious for relative schemes. The issue seems to stem in part from the use of localist representations, but it is unclear how using distributed representations would actually improve the situation. Indeed, as described earlier, one example of a distributed string code can be found in the original PDP model and was applied to the Wickelcoding scheme. The code has good robustness to noise and provides graded similarity measures for strings of more than three letters, but it is of little use in the problem described above. This is because as triplets are randomly assigned to units, there is still no relationship between units coding for triplets with similar letter constituents. This phenomenon is related to the *dispersion problem* observed in PDP (i.e., the fact that orthographic-to-phonological regularities learned in some context do not carry over to different contexts) and it has been identified as a major obstacle to generalization in the PDP model (Plaut, McClelland, Seidenberg, & Patterson, 1996).

What seems to be required is a way to represent letter-combination entities so as to respect similarities between combinations that share constituents, and we now describe how this can be achieved using holographic representations.

1.5. Binary spatter code

Binary spatter codes (BSC; Kanerva, 1997, 1998) are a particular case of holographic reduced representations (Plate, 1995) and provide a simple way to implement a distributed and compositional code.

Formally, BSCs are randomly distributed vectors of large dimensions (typically in the order of 10^3 to 10^4), where each element is bipolar, identically and independently distributed in a Bernoulli distribution. Vectors can be combined together using two particular operators: X-or and Majority rule, often, respectively referred to as the Binding and Chunking operators.[5] The composition of BSC vectors using these operators results in a new vector of the same dimension and distribution.

Operators are defined element-wise (see Appendix A for formal definitions): X-oring two bipolar variables gives 1 if they are different and −1 otherwise, while the majority rule applied to k bipolar variables gives 1 if they sum up above zero and −1 otherwise. Both operators preserve the format (size and distribution) of vectors, so that the resulting vectors can themselves be further combined, thus allowing for arbitrarily complex combinatorial structures to be built in a constant sized code.

As an example, consider how to implement the sequence of events [A, B, C] with an absolute scheme and vectors of dimension 10. We start by generating three event vectors at random:

$$A := [1, 0, 1, 1, 1, 0, 0, 1, 1, 0]$$
$$B := [1, 0, 0, 0, 0, 1, 0, 1, 0, 1]$$
$$C := [0, 1, 0, 1, 0, 1, 1, 1, 0, 1]$$

Likewise, we also generate at random the three temporal vectors required by an absolute position scheme:

$$T1 := [1, 0, 1, 1, 0, 1, 1, 0, 1, 1]$$
$$T2 := [0, 1, 0, 1, 0, 0, 0, 0, 0, 1]$$
$$T3 := [1, 1, 0, 1, 1, 0, 1, 1, 1, 1]$$

We express the idea that an event occurred at a given time by X-oring these vectors together, giving the three bindings:

$$A \otimes T1 = [0, 0, 0, 0, 1, 1, 1, 1, 0, 1]$$
$$B \otimes T2 = [1, 1, 0, 1, 0, 1, 0, 1, 0, 0]$$
$$C \otimes T3 = [1, 0, 0, 0, 1, 1, 0, 0, 1, 0]$$

These vectors are again densely and randomly distributed ($p(0) = p(1) = 1/2$). Calculating the expected similarities[6] between one binding and its constituents, we see that it must be zero: X-or destroys similarity.

Finally, we signify that these three events constitute one sequence by applying the majority rule:

$$S = \text{Maj}(A \otimes T1, B \otimes T2, C \otimes T3)$$
$$\equiv A \otimes T1 \oplus B \otimes T2 \oplus C \otimes T3 = [1, 0, 0, 0, 1, 1, 0, 1, 0, 0]$$

The sequence vector *S* that results from the Maj also has the same distribution as its argument vectors. As we calculate its expected similarity with its arguments, we see that contrary to the X-or it is non-zero, but here, 1/2.[7] The fact that Maj allows for part/whole similarities also implies that two structures which share some constituents will be more similar than chance.

Note that BSCs can be made arbitrarily sparse (Kanerva, 1995), although in this article, we use the dense version for simplicity. It has also been shown that randomly connected sigma–pi neurons can achieve holographic reduced representations (Plate, 2000). In addition, holographic techniques, such as BSCs have been recently applied to a number of language related problems, including grammatical parsing (Whitney, 2004), learning and systematization (Neumann, 2002), lexical acquisition (Jones & Mewhort, 2007), language acquisition (Levy & Kirby, 2006), or phonology (Harris, 2002).

1.6. Comparing codes: Method, objections, and replies

The masked priming paradigm (Forster & Davis, 1984) is most commonly used to assess the relative merits of candidate schemes. Simply stated, in this paradigm, a mask is presented for 500 ms, followed by a prime for a very brief duration (e.g., 50 ms), and then directly by the target string for 500 ms. Using this paradigm, subjects are generally unaware of the prime—it is said to be subliminal. However, when target strings are preceded by subliminal primes that are close in visual form, subjects are significantly faster and more accurate in their performances. In most studies the primes of interest are non-words, and there is no manipulation of frequency [F] or neighborhood [N]). Varying the orthographic similarity with the target (e.g *hoss*-TOSS vs. *nard*-TOSS), one can observe different amounts of facilitation that can then be compared to similarity scores derived from candidate schemes.

The way similarity scores are calculated depends upon the scheme. Most of the time, the similarity between prime and target strings is obtained simply by dividing the number of shared units by the number of units in the target. This is the similarity defined for instance in Grainger and Van Heuven's localist open-bigrams, and when absolute positions are taken into account, in slot coding. More sophisticated similarities can introduce weights for different units (Seriol model), signal-to-weight differences (Spatial Coding), or sums of integrals over Gaussian products (overlap model).

Some researchers have recently raised objections to this way of assessing codes. Lupker and Davis (2009) challenge the notion that masked priming effects can constrain codes, since the latter abstract away from N and F that despite all precautions will play a role in the former, and a new paradigm is proposed that is deemed less sensitive to these factors. Although this ''sandwich priming'' appears promising, a dismissal of the standard approach would certainly be premature. All researchers agree that the similarity/masked priming comparison should be used with caution—see, for example, Gomez et al. (2008)—which would exclude, for instance, conditions that explicitly manipulate primes for N or F. But the standard view is that in general masked priming facilitation should highly correlate with the magnitude of similarities, which in all models are thought to capture bottom-up activation

from sublexical units (see, for instance, Davis & Bowers, 2006; Gomez et al., 2008; Grainger et al., 2006; Van Assche & Grainger, 2006; Whitney, 2008). This view is also supported by unpublished investigations showing that similarities in the Spatial Coding scheme correlate 0.63 with simulated priming in the Spatial Coding model, and 0.82 when conditions that manipulate F and N are excluded (Hannagan, 2010).

Although such correlations are bound to differ across models because these make different assumptions about N and F mechanisms, the differences put models' behaviors into perspective, allowing one to distinguish bottom-up contributions from other contributions in priming simulations. Finally, in some circumstances, similarities alone can be sufficient to rule codes out. Such a situation occurred, for instance, in the case of standard localist slot coding, where a string code suffers equal disruptions from double substitutions or from transpositions. No lexical level from any model could save this localist code from being inconsistent with human data, because these show strong priming in the latter case but weak or no priming in the former (Perea & Lupker, 2003a).

In summary, assessing codes on masked priming constraints has been standard practice in the field, can rule out codes on face value, arguably provides accurate expectations for priming simulations carried out on models, and in any case provides a useful way to understand their inner workings. Consequently and following others, in this article, we will use similarity scores to predict these priming effects that would be expected from the codes alone in any lexical access model, and we will assess codes exclusively on masked priming data.

1.7. Four family of constraints from masked priming

We now describe several properties we believe word codes should display. These properties are by no means exhaustive but were chosen because they have been consistently reported in the literature.

1.7.1. Stability

First and foremost the code should be stable, if words are ever to be accessed reliably. A sufficient condition of stability for string S is that the similarity between S and any other string should be strictly less than one, but it should be one between S and itself. However, this cannot be a necessary condition, as in a noisy environment, the system must be prepared to accept some variation to occur even in the identity condition. Thus, we propose that a necessary and sufficient stability condition for string S is that in the absence of noise, no transformation of S can produce a code closer to S than a given similarity criterion (we use 0.95 throughout this article). We will test for stability of five letter strings by computing the codes for minimal transformations, fair instances of which being substitutions, transposition, insertion, repetition, or deletion (see Table 1 for examples).

1.7.2. Edge effects

Another desirable although controversial property concerns the importance of outer letters relatively to inner ones. In a letter identification paradigm, subjects are significantly

Table 1
Description of the constraints used in this article, and their respective conditions

Constraint Family	Condition	Prime	Target	Criterion
Stability	(1)	12345	12345	> 0.95
	(2)	1245	12345	< (1)
	(3)	123345	12345	< (1)
	(4)	123d45	12345	< (1)
	(5)	12dd5	12345	< (1)
	(6)	1d345	12345	< (1)
	(7)	12d456	123456	< (1)
	(8)	12d4d6	123456	< (7)
Edge effect	(9)	d2345	12345	< (10)
	(10)	12d45	12345	< (1)
	(11)	1234d	12345	< (10)
Transposed letter effect	(12)	12435	12345	> (5)
	(13)	21436587	12345678	= Min
	(14)	125436	123456	< (7) and > (8)
	(15)	13d45	12345	< (6)
Relative position effect	(16)	12345	1234567	> Min
	(17)	34567	1234567	> Min
	(18)	13457	1234567	> Min
	(19)	123256	1232456	> Min
	(20)	123456	1232456	= (19)

Note. Min:minimum similarity value among all conditions.

better at reporting the first and last letters than any other (Stevens & Grainger, 2003). In masked priming lexical decision studies, primes being one or two letters different from the target produce weaker (or no) facilitation when outer letters are manipulated than when inner letters are—see Jordan, Thomas, Patching, and Scott-Brown (2003) for a good summary of studies on the subject. The facilitation produced is often equal for initial or final letter substitutions (Grainger & Jacobs, 1993), but edge effects have sometimes proved elusive (Grainger et al., 2006; Guerrera & Forster, 2008), or confined to the initial letter (Whitney, 2001). In this article, we will say that a code shows edge effects if single substitutions occurring at outer positions are more destructive in terms of similarity than those occurring at inner ones (i.e., sim(d2345, 12345) < sim(12d45) and sim(1234d, 12345) < sim(12d45)[7]).

1.7.3. Transposition priming

Transposed letter (TL) priming refers to the robust finding that transposing two contiguous letters is less destructive than replacing them altogether (e.g ''jugde'' primes ''JUDGE'' significantly more than ''jupte'' does; Perea & Lupker, 2003a). Recently, TL has been further investigated in a number of directions and the influence of contiguity, status, length or number of transposed letters is being assessed (Lupker, Perea, & Davis, 2008; Perea & Acha, 2008; Perea, Duñabeitia, & Carreiras, 2008). In this article, we choose to focus on four results: the basic single contiguous TL constraint for five letter strings

(sim(12435, 12345) > sim(12dd5, 12345), hereafter local TL), a global TL constraint from a recent study (Guerrera & Forster, 2008) where transposing all contiguous letter couples from an eight-letter string produced no facilitation (sim(21436587, 12345678) = Min,[8] a distant TL constraint where a single noncontiguous transposition in six-letter words was found to give more facilitation than the corresponding double substitution, but less than a single substitution (sim(12d456) < sim(125436, 123456) < sim(12d4d6, 123456)) (Perea & Lupker, 2004), and finally what we will call the compound TL constraint, stating that a transposition coupled with a substitution (13d45, neighbors-once-removed, N1R hereafter) produces less priming than a single median substitution (Davis & Bowers, 2006). The global TL constraint currently defies predictions from all schemes but Seriol[9] by producing no priming at all,[10] whereas the compound and distant TL constraints seem to challenge all schemes, but Spatial Coding.

1.7.4. RP priming

Relative position priming covers the fact that strings obtained by disturbing the absolute position of the target letters, while sparing their relative positions, still make efficient primes (e.g., 12345, 34567, and 13457 all prime the word 1234567; Grainger et al., 2006). This goes against expectations based on absolute letter coding schemes like slot coding, as well as schemes that incorporate some local context like Wickelcoding. Although RP priming has recently been extended to superset primes (Van Assche & Grainger, 2006), in this article we will focus on subset conditions but will consider conditions with and without repeated letters. A code for a seven-letter word will be said to satisfy the distinct RP priming constraint if it exhibits strictly positive similarities with the strings described above (12345, 34567, and 13457). The repeated RP constraint covers the finding that given a target with one repeated letter, the same amount of priming is produced whether a deletion involves the repeated letter or not (sim(123456, 1232456) = sim(123256, 1232456)). This result is at odds with predictions coming from the open-bigram scheme (Schoonbaert & Grainger, 2004).

Table 1 recapitulates these constraints.

1.8. General procedure

In the next four sections, we will assess a number of schemes on this set of criteria. The same procedure is used throughout all simulations: Each scheme is implemented using holographic coding on the one hand, and localist coding on the other, for comparison purposes. Similarities in the localist case are computed in the standard way described earlier. The holographic procedure is carried out as follows. For each trial, we randomly generate orthogonal holographic vectors of dimension 1,000. From these, we build the various codes for all conditions, compute similarity scores based on hamming distance between vectors, and average them across 100 trials. The use of high dimensional vectors implies very little variance for these similarity scores, typically 0.01 for dimension 1,000. Consequently, we will only report mean similarities, and we will consider that two conditions yield equal facilitation if their difference in similarities lies within 0.01.
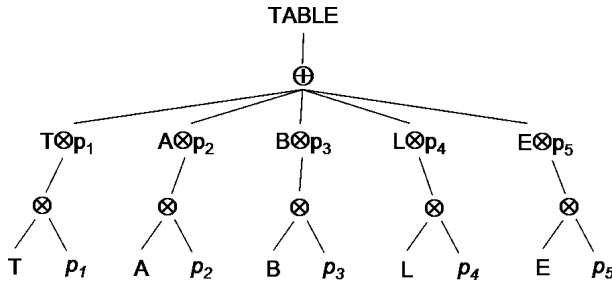
Fig. 1. Holographic slot coding: arborescence of the code. Each node represents a vector of constant size, while ⊗ and ⊕ connections respectively stand for the X-or and Maj operators. The *i*th element of the global code is one if and only if a majority of *i*th elements from letter/position bindings are 1.

2. Slot coding

We begin by translating the slot coding scheme in the language of holographic representations. This is done by generating position and letter vectors. Each letter vector is bound to a position vector representing the position within the string, using the X-or operator. These bindings are then chunked together using the majority rule, resulting in the holographic structure:

$$\text{TABLE} = T{\otimes}p_1 \oplus A{\otimes}p_2 \oplus B{\otimes}p_3 \oplus L{\otimes}p_4 \oplus E{\otimes}p_5,$$

where $\otimes$ is the binding operator X-or, $\oplus$ is the majority rule operator, and p_i are independent holographic vectors. Fig. 1 gives the arborescence of the holographic slot coding scheme.

2.1. Results

Table 2 presents similarity results for both versions of the slot coding scheme. The first line shows the localist code, and the second the holographic code. As is well known, when the string size is unchanged the least destructive transformation in localist slot coding is single substitution (conditions 6, 7, 9, 10, 11), and it produces a code distant enough from the original to satisfy the stability constraint. Moreover, a single substitution has the same impact wherever it occurs in the string, thus precluding edge effects (conditions 9–11). Generally speaking, the code is a quite stringent one, sanctioning even the slightest transformation with very low similarity scores (conditions 2–5). A single transposition in slot coding is necessarily as destructive as a double substitution, thus preventing local or distant transposition effects (resp. conditions 12 vs. 5, and 14 vs. 8). However, localist slot coding correctly captures the absence of similarity resulting from global transposition (condition 10), and rightly makes N1R more destructive than single substitution neighbors (conditions 15 vs. 6). Subset conditions produce very different similarities depending on where deletions are performed: The earlier it occurs in the string, the more destructive it is. Similarities decrease steadily for deletion positions 6, 5, 4, and 2, hitting a floor at position 1 where no similarity to the original string remains (resp. conditions 16, 19, 20, 18, and 17), and as a consequence both the distinct and the repeated RP constraints are violated.

Table 2
Slot coding scheme: similarities derived in the localist and holographic case

Condition	Stability (1–8)								Edge Effects (9–11)		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Prime	12345	1245	123345	123d45	12dd5	1d345	12d456	12d4d6	d2345	12d45	1234d
Target	12345	12345	12345	12345	12345	12345	123456	123456	12345	12345	12345
Localist slot	**1.00**	**0.40**	**0.60**	**0.60**	**0.60**	**0.80**	**0.83**	**0.67**	**0.80**	**0.80**	**0.80**
Holographic slot	**1.00**	**0.28**	**0.36**	**0.36**	**0.43**	**0.76**	**0.83**	**0.67**	**0.62**	**0.62**	**0.62**

Condition	Transposition (12–15)				Relative Position (16–20)				
	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
Prime	12435	21436587	125436	13d45	12345	34567	13457	123267	123567
Target	12345	12345678	123456	12345	1234567	1234567	1234567	1232567	1232567
Localist slot	**0.60**	**0.00**	**0.67**	**0.60**	**0.71**	**0.00**	**0.14**	**0.57**	**0.43**
Holographic slot	**0.62**	**0.00**	**0.84**	**0.67**	**0.68**	**0.00**	**0.12**	**0.70**	**0.65**

Notes. Average similarity scores (bold) calculated for different prime–target conditions. For each condition, the main constraint it intends to test is outlined. Note that technically the stability constraint extends to all conditions.

As for holographic slot coding, Table 2 first shows that it is stable—the identity condition returns one (condition 1) while all other transformations for five-letter strings remain well below the instability threshold.

A remarkable feature of holographic slot coding is that it produces transposition effects: Transposing two letters (condition 12, sim 0.62) is less destructive than replacing them altogether (condition 5, sim 0.43). This is surprising because as explained earlier, binding two vectors using X-or yields a new vector that bears no similarity to either of them (i.e., $\text{sim}(A, A \otimes B) = \text{sim}(B, A \otimes B) = 0$). Thus, a given letter vector T, X-ored with either the first or the second position vector, will give two unrelated vectors (i.e., $\text{sim}(T \otimes p_1, T \otimes p_2) = 0$). Consequently, at first sight, one might not expect any TL effects to occur.

However, in Appendix B, we demonstrate that this TL effect is an inevitable consequence of the combinatorics involved with BSC's operators when order is encoded using positional cues. Indeed, in terms of hamming distance to the original string, we show that double substitutions must be more destructive than transpositions by a factor of 3/2. The reason why one observes a TL effect in holographic slot coding, despite letter and position vectors being orthogonal, can be intuited in dimension one, where each representation is a simple bipolar variable. Consider two letters A and B, respectively, at position p_1 and p_2 in a string of arbitrary length. All other things being equal, transposing letters A and B will only change the code for the string if $A \cdot p_1 + B \cdot p_2$ is different from $A \cdot p_2 + B \cdot p_1$. It is easy to compute the probability of this happening, and to see that it is not equal to 1/2 (chance), but to 1/4. Thus, transposing two letters is less destructive than replacing them altogether.

Furthermore, this TL effect can be shown empirically to decrease with the number of letters involved in a transposition, and to vanish when all letter couples are transposed (condition 13), in good agreement with Guerrera and Forster (2008). Hence, it seems that this code displays the adequate flexibility to solve a difficult conundrum: accounting for local transposition effects without predicting any facilitation for global transpositions. As to this day and to our knowledge, holographic slot coding stands alone in this precise matter. The code also satisfies the compound TL constraint (condition 15 lower than 6) as well as part of the distant TL constraint: If it mistakenly equates the impacts of distant transposition and single substitution (condition 14 vs. 7), holographic slot coding nevertheless rightly makes both of them less disruptive than distant double substitutions (conditions 8).

Be it localist or holographic, however, the absolute slot coding scheme has two limitations: It cannot reproduce RP effects and edge effects, the final letter being no more important than inner ones. Although it is possible to introduce a weight gradient in the Maj operator (see Appendix A), this weighted holographic slot coding version cannot fix all the weaknesses of the slot coding scheme. In particular, it would suffer from the same alignment problem as the localist version because all absolute positions are disturbed by an initial letter deletion. The only way in which relative position priming could be restored to some extent would be to use double anchors, a move that might be supported by a recent study (Fischer-Baum et al., 2010), or alternatively to introduce a small correlation between position vectors.[11] Arguably, it might even be possible to combine both moves in

a holographic overlapping and double-anchored scheme, which could presumably inherit the qualities of each.[12] Either way, we see that to account for edge effects and relative position priming, a positional code needs to introduce several additional hypotheses. Thus, for the sake of parsimony, we should first make sure that all simple alternatives have been ruled out.

2.1.1. Summary

In summary, localist slot coding can satisfy three constraints: stability, compound, and global TL, but it fails on edge effects, local and distant TL, and RP constraints. Holographic slot coding satisfies the same constraints plus local TL, which obtains by virtue of the binding and chunking operators. This shows that contrary to the common belief, local TL is not sufficient to rule out the slot coding scheme, and instead, the real obstacle appears to be in RP constraints.

3. Open-bigrams

One simple alternative to slot coding is the open-bigram scheme, which we now proceed to study, first in its constrained version and then in the unconstrained one. Again, in both cases, we will compare the localist and holographic implementations.

3.1. Constrained open-bigrams

The COB code can be translated in holographic representations as follows:

$$\begin{aligned} \text{TABLE} = \; &TA \oplus TB \oplus TL \\ &\oplus AB \oplus AL \oplus AE \\ &\oplus BL \oplus BE \\ &\oplus LE \end{aligned}$$

where the code for bigram $L_i L_j$ is:

$$L_i L_j = L_i \otimes l \oplus L_j \otimes r$$

and l, r are two orthogonal vectors, respectively, coding for left and right positions within a bigram. The arborescence of the holographic COB code is depicted in Fig. 2.

3.1.1. Results

Table 3 shows that whether localist or holographic, the COB scheme is stable. It is also more flexible than slot coding, as apparent from the fact that similarity patterns have generally increased. Despite this general increase, similarity rankings remain more or less unchanged with respect to slot coding. The main difference is that single substitutions in five-letter strings (conditions 6, 9, 10, 11) are now more disruptive than repetitions (condition 3), additions (condition 4), or transpositions (condition 12). However, the most destruc-
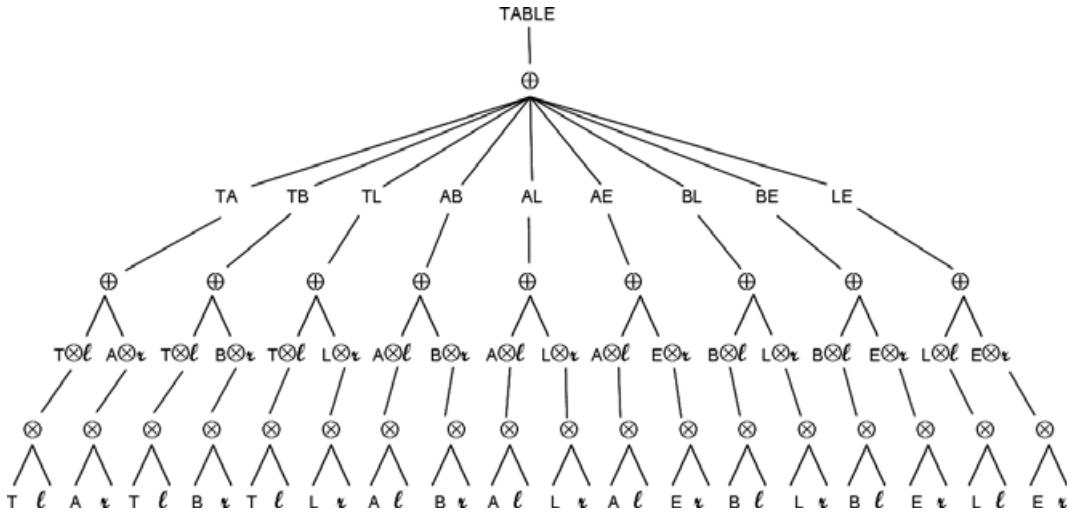
Fig. 2. Holographic COB (constrained open-bigrams): arborescence of the code.

tive transformation for five-letter strings is still double substitution (condition 5), while the least destructive is still local transposition (condition 9).

Switching to a bigram scheme does not improve the number of constraints satisfied but changes their variety, allowing one to account for some aspects of TL and RP priming at the same time. Indeed, deleting the initial letters from a string (resp. transposing two letters) now results in high similarities because the relative position of letters is maintained (resp. not completely lost) and thus several bigrams remain from the original. However, this comes at the cost of global, distant, and compound TL (conditions 13–15), for which both codes now make wrong predictions. Global TL produces high similarities because many letters are left in the same relative order and thus generate the same bigrams, distant TL is less disruptive than single substitution because it affects four bigrams rather than five, and compound TL is violated as the N1R condition is exactly as disruptive as single substitution. In this last case, this is because an N1R string (13d45) maintains the same relative order to the target as a single substitution neighbor (1d345), producing exactly the same set of bigrams except for one, bigram ''3d'' in the first case and ''d3'' in the second. On both accounts, this leads to identical similarities with the original string (12345).

There are also differences between the two implementations of COB. A first difference is that the holographic, but not the localist code, satisfies the repeated RP constraint (i.e., conditions 19 and 20 should produce equal similarities). In the localist case, each bigram counts only once in the representation and thus redundant bigrams play no role in similarity calculations. The undue difference between repeated condition (19) and nonrepeated condition (20) then occurs because the latter shares ten bigrams with the target, whereas the former only shares nine.[13] In the holographic case, on the other hand, redundant bigrams all contribute through the majority rule to impact on the final representation. Hence, in conditions

Table 3
Constrained open-bigrams (COB): similarities derived for holographic and localist codes across selected conditions

	Stability (1–8)								Edge Effects (9–11)		
Condition	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Prime	12345	1245	123345	123d45	12dd5	1d345	12d456	12d4d6	d2345	12d45	1234d
Target	12345	12345	12345	12345	12345	12345	123456	123456	12345	12345	12345
Localist COB	**1.00**	**0.56**	**0.78**	**0.78**	**0.22**	**0.55**	**0.58**	**0.33**	**0.67**	**0.55**	**0.67**
Holographic COB	**1.00**	**0.59**	**0.70**	**0.67**	**0.42**	**0.78**	**0.78**	**0.79**	**0.65**	**0.64**	**0.65**

	Transposition (12–15)				Relative Position (16–20)				
Condition	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
Prime	12435	21436587	125436	13d45	12345	34567	13457	123267	123567
Target	12345	12345678	123456	12345	1234567	1234567	1234567	1232567	1232567
Localist COB	**0.89**	**0.67**	**0.66**	**0.56**	**0.60**	**0.60**	**0.47**	**0.82**	**0.91**
Holographic COB	**0.76**	**0.72**	**0.84**	**0.78**	**0.59**	**0.59**	**0.62**	**0.83**	**0.82**

Note. Boldface type indicates similarities.

(19) and (20), the strings are really made of 12 bigrams each, and as it turns out, they share exactly 10 of them with the target, resulting in equal similarities.

Another difference is that localist COB displays reverse edge effects, whereas there are little or no such effects in the holographic code (conditions 9–11). It is only natural that the former displays reverse edge effects. Indeed, Fig. 2 shows that each edge letter is only involved in three bigrams, against four bigrams for inner letters. Thus, outer substitutions disturb fewer bigram units than inner substitutions, and they result in codes that are more similar to the original. This implies that reverse edge effects will be produced by just every COB scheme, whatever the length of the word or the authorized letter gap. To avoid this reverse effect, one must resort to unconstrained bigrams or edge units.

The same logic should apply to the holographic code, and we should thus ask why it yields very small reverse edge effects, if any. The key difference is that as already noticed, the localist version assigns a similarity of zero between any two distinct bigram units. On the contrary, the holographic version provides graded similarities between distinct bigrams, for the same reasons previously exposed for slot coding. Thus, replacing bigram ''TA'' by, for example, bigram ''XA'' does not completely destroy similarities in a holographic scheme, and this might provide one cue as to why reverse edge effects are counterbalanced. We will shortly give a more thorough explanation of edge effects in the context of unconstrained open-bigrams.

3.1.2. Summary

Contrary to slot coding, the COB scheme can accomodate TL constraints as well as RP constraints. Localist COB satisfies three constraints: stability, local TL, and distinct RP. Again, the holographic version satisfies precisely the same constraints plus another: repeated RP, which obtains essentially because repeated bigrams contribute more to the holographic word code. This solves the difficult conundrum for bigram schemes introduced in Schoonbaert and Grainger (2004), showing that repeated RP does not challenge the COB scheme itself, but only its localist implementation.

3.2. Unconstrained open-bigrams

The unconstrained open-bigrams code can be translated in holographic representations as follows:

$$\text{TABLE} = TA \oplus TB \oplus TL \oplus TE$$
$$\oplus AB \oplus AL \oplus AE$$
$$\oplus BL \oplus BE$$
$$\oplus LE$$

Its arborescence, exactly identical to COB except for additional bigrams (bigram TE in our case), is given in Fig. 3.
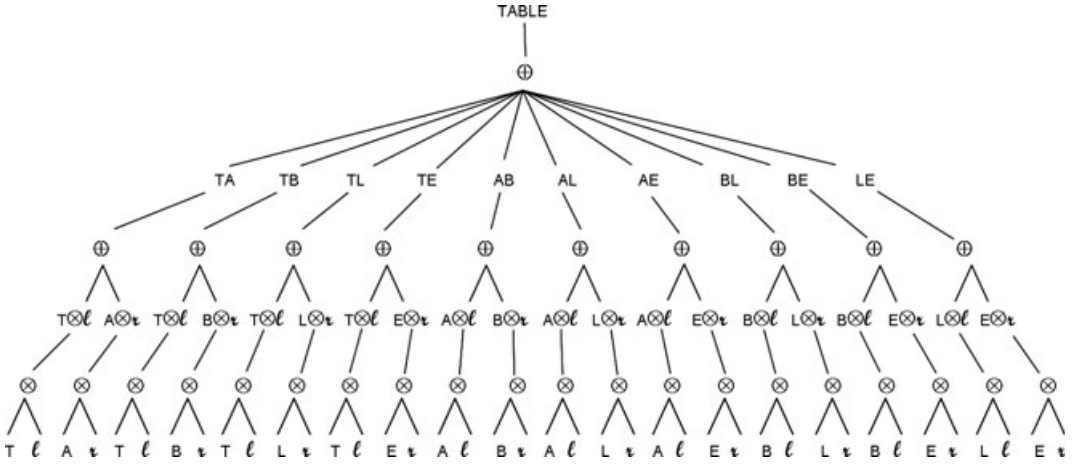
Fig. 3. Holographic UOB (unconstrained open-bigrams): arborescence of the code.

3.2.1. Results

The UOB scheme behaves like the constrained one in a number of ways. For the same reasons as above, both the localist and the holographic UOB versions satisfy the local TL and distinct RP constraints, but they fail to reproduce distant, global, and compound transposition effects. Nevertheless, Table 4 indicates that the differences due to the code are most salient for the UOB scheme: in fact, according to our benchmark presented in Table 6, localist and holographic UOB, respectively, rank as last and first in terms of explanatory power.

Localist UOB violates the stability constraint: Insertions or repetitions produce equivalent codes to the original, as defined by the similarity measure. This problem does not appear in the holographic implementation, where the less disruptive transformation is still single transposition (condition 12), and the similarity remains well below our stability criterion (0.82 < 0.95). This is because in holographic codes, the similarity measure is not target oriented but proceeds from the symmetrical hamming distance.

A second difference is that holographic UOB satisfies the repeated RP constraint. As in the case of COB, localist UOB is hindered by the exclusion of redundant bigrams from similarity calculations. Another difference concerns edge effects. As expected from the previous section, the reverse edge effects observed with localist COB are absent with UOB. Indeed, single substitutions, be they inner or outer, now impact on exactly the same number of bigrams: $L-1$, where L is the string length. However, the holographic implementation now displays strong edge effects.

It is possible to gain some insights into why edge effects appear in the UOB code by returning to the arborescence in Fig. 3. Let us place ourselves in the one-dimensional case, and look at the four bigrams in which the inner letter B occurs. Across these bigrams, letter B appears twice on the left, and twice on the right. Thus, if left and right position variables have different values, the ''vote'' of letter B in the majority rule simply cancels itself out. However, this cannot happen if we now consider the four bigrams in which an outer letter

Table 4
Unconstrained open-bigrams (UOB): similarities derived for holographic and localist codes across selected conditions

Condition	Stability (1–8)								Edge Effects (9–11)		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Prime	12345	1245	123345	123d45	12dd5	1d345	12d456	12d4d6	d2345	12d45	1234d
Target	12345	12345	12345	12345	12345	12345	123456	123456	12345	12345	12345
Localist UOB	**1.00**	**0.60**	**1.00**	**1.00**	**0.30**	**0.60**	**0.66**	**0.40**	**0.60**	**0.60**	**0.60**
Holographic UOB	**1.00**	**0.62**	**0.79**	**0.73**	**0.45**	**0.79**	**0.85**	**0.75**	**0.49**	**0.62**	**0.53**

Condition	Transposition (12–15)				Relative Position (16–20)				
	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
Prime	12435	21436587	125436	13d45	12345	34567	13457	123267	123567
Target	12345	12345678	123456	12345	1234567	1234567	1234567	1232567	1232567
Localist UOB	**0.90**	**0.86**	**0.80**	**0.60**	**0.48**	**0.48**	**0.48**	**0.71**	**0.88**
Holographic UOB	**0.82**	**0.79**	**0.92**	**0.79**	**0.52**	**0.51**	**0.61**	**0.89**	**0.89**

Note. Boldface type indicates similarities.

occurs, because of course outer letters always appear at the same position in any bigram. Thus, the ''vote'' of an outer letter naturally carries much more weight than the vote of an inner letter. As a consequence, inner substitutions are less disruptive than outer ones. Note that this ''voting'' combinatorics induced by the operators is also present in holographic COB, although it does not result in edge effects because it must fight the previously discussed reverse tendency resulting from the constraint on bigrams.

3.2.2. Edge effects in holographic UOB

From this, we can draw three predictions. First, the same mechanism should hold regardless of word length, so that edge effects should not be restricted to this example. Second, edge effects should not arise only from the contrast with this particular inner position. In fact, the similarity pattern created by a single substitution should not be constant across inner positions, but increase as one approaches the edges, because the number of cases in which the substituted letter's vote cancels out diminishes as one gets closer to the edges. Third and for the same reason, the similarity obtained with inner left substitutions (resp. for inner right substitutions) should be proportional to the binomial number C_r^l (resp. C_l^r), where l and r, respectively, stand for the number of appearance of the substituted letter in left and right positions.

To test these claims, we computed similarities for single substitutions across all positions in strings of length 5 and 7, and compared them with the corresponding human data for five-letter words reported in Grainger and Jacobs (1993).

Fig. 4 first shows that simulations can match the human data obtained for five-letter strings. Although Grainger and Jacobs (1993) found no significant difference between outer conditions or between inner ones, and the medial condition was not carried out, there is a visible tendency toward an inverse U-shaped form which is also pickep up by holographic UOB similarities. What we can say thus is that in this code, despite the lack of any
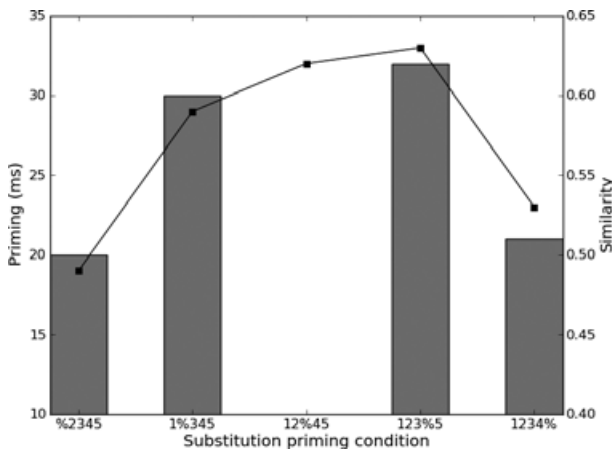

Fig. 4. Single-substitution priming across position for five-letter words. Bars: experimental results from Grainger and Jacobs (1993) (central condition unavailable). Curve: similarities from holographic UOB.

information on absolute position or any weight gradient, the correct priming pattern appears in virtue of the prominence of letters in left or right positions across bigrams.

This saliency pattern also impacts on distinct RP priming. Having equal letter weights, the localist UOB code naturally shows constant similarities for initial, medial, or final deletions. On the contrary, in the holographic implementation, we find that initial or final deletions should be more disruptive than medial ones. This is not in agreement with behavioral data from Grainger et al. (2006), which show a tendency for less facilitation with medial primes.

3.2.3. Summary

Differences between implementations are most salient within the UOB scheme. While localist UOB fails on all constraints but local TL and distinct RP, holographic UOB satisfies in addition stability, edge effects, and repeated RP. Stability and repeated RP are recovered because all bigrams contribute to determine the final holographic code. Edge effects emerge naturally from the chunking operator, which tends to cancel out contributions from central letters in a holographic code. This simple explanation could easily have been falsified by behavioral data, which on the contrary support the holographic UOB account (Grainger & Jacobs, 1993).

3.3. LCD

We finally proceed to implement a holographic code in the spirit of the LCD proposal (Dehaene et al., 2005). The arborescence of the code is illustrated in Fig. 5.
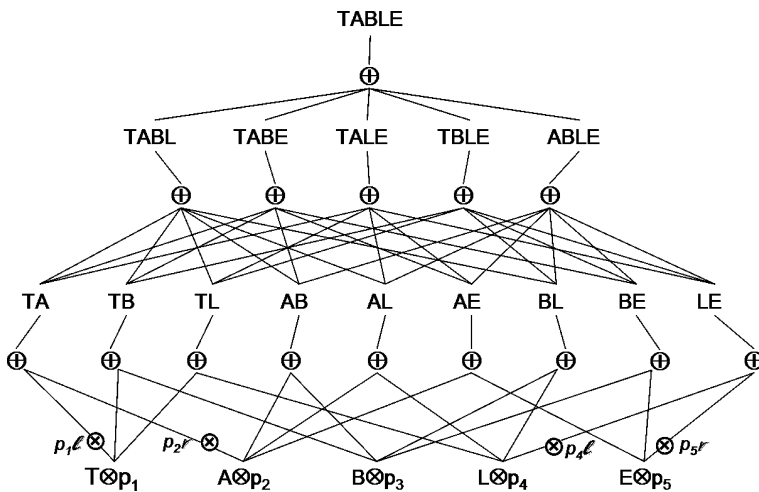

Fig. 5. Arborescence of holographic LCD (local combination detector). A three-level hierarchy progressively achieves recognition of larger letter combinations. Simple X-or operations (only shown for the first and last bigrams) perform the transition between absolute position (letter level) and relative position (bigram level) coding.

The letter and bigram levels of the hierarchy are, respectively, identical to the holographic slot and COB codes. However, in the previous sections, we have studied them in isolation, whereas the two schemes are now connected levels in the LCD architecture. Apart from the difference in the authorized gap between constituent letters (one for the original LCD against two here), our implementation departs from LCD in several important ways. LCD assumes that some position specificity remains at the bigram level. This has the consequence that duplicated banks of bigram units must exist at this level, each receptive to a particular part of the string. On the contrary, our code follows the initial COB approach, which assumes only a single bank of units, relevant bigrams being activated regardless of the absolute positions of letters in the string. Also, in LCD the letter to bigram transition is thought to be achieved by banks of bigram units with overlapping receptive fields, whereas in the proposed code, this transition between position specificity and invariance is achieved using simple X-or operations taking place between the letter and bigram levels.

This transition can be illustrated by building bigram unit $L_iL_j := L_i {\otimes} l \oplus L_j {\otimes} r$ from letters L_i at position p_i, and L_j at position p_j, $i > j$. This requires two arbitrary left and right vectors l and r and the following operations:

$$
\begin{aligned}
L_iL_j &= (L_i{\otimes}p_i) \otimes (l{\otimes}p_i) \oplus (L_j{\otimes}p_j) \otimes (r{\otimes}p_j) \\
&= (L_i{\otimes}l) \otimes (p_i{\otimes}p_i) \oplus (L_j{\otimes}r) \otimes (p_j{\otimes}p_j) \text{ because X-or is associative} \\
&= L_i{\otimes}l \oplus L_j{\otimes}r \text{ because X-or is its own inverse}.
\end{aligned}
$$

At the quadrigram level, we use units allowing for a one-letter gap. Again, this differs from LCD, which makes no such assumption but holds that only frequent quadrigrams are activated. As this code does not incorporate any notion of frequency, another selection mechanism must be assumed and open-quadrigrams of gap 1 seem to achieve the correct compromise between number and diversity of units. Alternatively, this level can also be seen as an extension of Mozer's (1987) Blirnet to four-letter units.

Eventually the LCD code is built in the following way:

$$\text{TABLE} = \text{TABL} \oplus \text{TABE} \oplus \text{TALE} \oplus \text{TBLE} \oplus \text{ABLE}$$

where each quadrigram unit is activated by consistent units from the bigram level, for instance:

$$\text{TABE} = TA \oplus TB \oplus AB \oplus AE \oplus BE$$

and where, for example, the code for bigram *TA* is:

$$TA = T{\otimes}l \oplus A{\otimes}r$$

At any level, a unit's code is obtained in the usual way, by chunking the code of consistent subunits from the previous level. Note the absence of positional vectors in the final

code, resulting from the hypothesized unbinding that takes place between letter and bigram levels. In effect, all absolute position information is lost at this point and replaced by relative information, although Fig. 5 features an absolute letter level and unbinding operations to highlight the relationship to LCD. Also, bigram TE, although consistent, does not participate in quadrigram TABE. Indeed, this bigram is not activated because the constraint on the authorized letter gap would be violated.

3.3.1. Results

Table 5 reports similarity scores obtained using holographic LCD, as well as its localist open-quadrigram counterpart. Whatever the code, this version of LCD is stable and can account for distinct RP, although in both cases outer conditions show higher similarities than inner ones. This contrasts with what we observed for holographic UOB, where edge effects induced the opposite tendency for relative priming conditions. This is because in holographic LCD, large units and reduced gap conspire to produce a code that is more sensitive to contiguity.

There is a lack of flexibility in the localist quadrigram code that is particularly visible for double substitutions, where the code produces no similarity at all but where the holographic version assigns generous similarities. In fact, as a consequence, both codes fail to reproduce distant TL: While the holographic code mistakenly predicts more facilitation for a distant transposition than for a single substitution, the localist code predicts no priming whatsoever with double substitution or distant transposition primes.

Neither the localist nor the holographic version of LCD satisfies the repeated RP constraint, assigning different similarities in conditions (19) and (20). Moreover, no version can account for compound TL. This can be easily seen in the localist case, because N1R and single substitution primes have exactly one quadrigram in common with the target. In the holographic case, things are more complicated because quadrigrams can have graded similarities, but the fact that the quadrigrams that differ in the two conditions are only one transposition away—and thus have the same distance to the corresponding quadrigram in the target—appears sufficient to make the codes themselves equally distant to the target.

As for the differences between implementations, it is worth noting that the rigidity of the localist code does not weigh entirely against it, as it appears to be in just the right proportion to capture both local and global TL effects, whereas the holographic code can only reproduce local TL. It might come as a surprise that two codes that differ in every respect, holographic slot coding and localist open-quadrigrams, are the only one able to accommodate local and global TL effects at the same time. But intuitively we see that they actually achieve similar compromises in terms of flexibility. Slot coding is a priori a strictly compartmented scheme, but implementing it holographically has the effect of suppressing positional frontiers to some extent. Open-quadrigrams inherit the rigidity of localist representations and use large units, but this rigidity is relaxed by using a relative position scheme that tolerates noncontiguity. However, a problem for localist LCD would be that as noted earlier, it predicts no similarity between four-letter strings, which is falsified experimentally (Humphreys et al., 1990).

Table 5
Unconstrained local combination detector (LCD): similarities derived for holographic and localist codes across selected conditions

	Stability (1–8)								Edge Effects (9–11)		
Condition	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Prime	12345	1245	123345	123d45	12dd5	1d345	12d456	12d4d6	d2345	12d45	1234d
Target	12345	12345	12345	12345	12345	12345	123456	123456	12345	12345	12345
Localist LCD	**1.00**	**0.20**	**0.40**	**0.40**	**0.00**	**0.20**	**0.22**	**0.00**	**0.20**	**0.20**	**0.20**
Holographic LCD	**1.00**	**0.53**	**0.45**	**0.51**	**0.37**	**0.78**	**0.80**	**0.70**	**0.56**	**0.63**	**0.56**

	Transposition (12–15)				Relative Position (16–20)				
Condition	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
Prime	12435	21436587	125436	13d45	12345	34567	13457	123267	123567
Target	12345	12345678	123456	12345	1234567	1234567	1234567	1232567	1232567
Localist LCD	**0.40**	**0.00**	**0.00**	**0.20**	**0.38**	**0.38**	**0.15**	**0.23**	**0.31**
Holographic LCD	**0.79**	**0.62**	**0.87**	**0.78**	**0.58**	**0.58**	**0.49**	**0.82**	**0.78**

Note. Boldface type indicates similarities.

Considering Fig. 5, we see that a localist open-quadrigram code cannot show edge effects in the case of five-letter strings, where each letter participates in exactly four quadrigrams, but will produce reverse edge effects in general, since inner letters would participate in more quadrigrams than outer ones. On the contrary, the holographic implementation does result in edge effects. The reason why edge effects are produced in holographic LCD can again be understood by considering the links to the quadrigram level. The arborescence in Fig. 5 shows that for each quadrigram unit, its set of five activating bigrams is balanced. This is to say that across these bigrams, letters can only appear at most twice at the same position, thus preventing systematical majorities to occur. However, edge quadrigrams are the exception to the rule. Embodying a sequence of contiguous letters, these units are activated by not five, but six bigrams. This disturbs the balance, and the initial letter (resp. final) is more represented in quadrigram 1234 (resp. 2345) than any other letter at any position in remaining bigrams. As a result, transformations involving edge letters are more destructive than others. It should be noted that edge effects in holographic LCD are also dependent on word length and seem to reverse for longer strings. Nevertheless, this code provides a clear demonstration that edge effects in holographic codes are not limited to unconstrained open-bigrams, but can also appear with constrained units of higher grain, even though in this case they appear to be inconstant.

3.3.2. Summary

In summary, our study of an LCD-like scheme suggests that just as open-bigrams, it can reproduce both TL and RP effects. Localist LCD is stable, satisfies distinct RP, and even accommodates local and global TL at the same time. Despite these interesting properties, the code appears to be falsified by priming results on four-letter words, a problem which is avoided in the holographic implementation. Holographic LCD has generally the same profile, but trades global TL against edge effects, illustrating that holographic edge effects can occur for other relative schemes than open-bigrams.

4. Discussion

Table 6 recapitulates how the various codes behave with respect to our benchmark of constraints.

It is clear from Table 6 that not all constraints are equally easy to satisfy. Distant TL in particular resists all the schemes considered above, whatever the implementation. This is because most of the codes wrongly imply that distant TL is less disruptive than single substitution. A bit less elusive is the compound TL constraint, which remains beyond the reach of all codes but localist or holographic slot coding. A scheme that has also been known to satisfy compound TL is Spatial Coding (Davis & Bowers, 2006), and we will shortly consider it in more detail. Much more reachable are global TL and edge constraints, as they are each satisfied by at least two different codes. The easiest constraints are the stability and local TL, met by seven codes of eight. There is also a general trend in Table 6 for TL constraints to be more often satisfied by holo-

Table 6
Summary of the constraints satisfied by the codes

	Stability	Edge Effects	Local TL	Distant TL	Compound TL	Global TL	Distinct RP	Repeated RP
Localist								
Slot coding	✔	–	–	–	✔	✔	–	–
COB	✔	–	✔	–	–	–	✔	–
UOB	–	–	✔	–	–	–	✔	–
LCD	✔	–	✔	–	–	✔	✔	–
Holographic								
Slot coding	✔	–	✔	–	✔	✔	–	–
COB	✔	–	✔	–	–	–	✔	✔
UOB	✔	✔	✔	–	–	–	✔	✔
LCD	✔	✔	✔	–	–	–	✔	–
Other schemes								
Spatial Coding	–	✔	✔	✔	✔	✔	✔	✔
Seriol	✔	–	✔	✔	✔	–	✔	–

Note. COB, constrained open-bigrams; UOB, unconstrained open-bigram; LCD, local combination detector; RP, relative position; TL, transposed letter.

graphic slot coding, and RP constraints by holographic relative position codes (COB, UOB, and LCD).

A second conclusion we can draw from Table 6 is that representational format has a profound impact on the number and nature of the constraints satisfied by a scheme: Holographic representations are globally better at explaining the available data. Indeed, with the notable exception of localist LCD on the global transposition effect, holographic codes always account at least for the same effects as localist ones. In addition, this implementation often satisfies more constraints: It recovers TL effects from the slot coding scheme, produces edge effects when used with open-ngrams, stabilizes UOB and makes it able to account for repeated RP.

Edge effects arise in codes using ngram units as a consequence of the holographic chunking operator, the majority rule. Importantly, we are not claiming that edge effects must exclusively be explained in this way. Rather, we report how a contribution to edge effects can arise from holographic codes. Several other factors may be at work to produce edge effects, including crowding effects and top-down facilitation (Tydgat & Grainger, 2009). As to the mechanisms by which TL effects appear in the slot coding scheme, we have informally described them earlier and they are laid down formally in Appendix B. It is important to note that this mechanism is also active for other schemes, for instance, by making unit AB and BA more similar than chance in open-bigrams. Generally speaking, both properties take root in the same basic quality of holographic coding, which is that compound units are similar to one another. This makes for more flexible codes, introduces rich majority combinatorics and breaks positional frontiers.

Repeated RP effects are produced in holographic COB and UOB again as a result of the holographic Maj operator. While localist versions are hindered by their exclusion of

redundant bigrams in similarity scores, the Maj operator in effect assigns different weights to bigrams in proportion to their redundancy. Taking all bigrams into account thus appears sufficient to satisfy the repeated RP constraint.

We also introduced a code in the spirit of the LCD model, although admittedly with considerable liberty. The localist version appears to be too rigid to account for single substitution priming results in words of various lengths (Humphreys et al., 1990; Perea & Lupker, 2004). Being distributed and hierarchical, the holographic version is much more in line wit the LCD proposal, and satisfies the same constraints except for global TL, that it ''exchanges'' for edge effects.

Despite these achievements, Table 6 shows that the most efficient holographic code with respect to our benchmark is holographic UOB, which satisfies one more constraint without invoking a quadrigram level. As acknowledged by Vinckier et al., the quadrigram level in LCD remains hypothetical and in fact even the evidence from Vinckier et al. (2007) is debatable because of some discrepancies between stimuli legality across conditions.[14] Secondly, as we have seen, edge effects in UOB are much more robust than in LCD. As for now, holographic UOB seems to be the richest code in phenomena and the most parsimonious in hypotheses.

4.1. Comparison with other schemes

In this article, we have been essentially concerned with the differences arising when a given scheme is implemented with distributed or localist representations. However, in this comparison process, we have left aside some well-known schemes, either because they were not easily translated into the holographic format (e.g., Spatial Coding, whose assumptions on how to compute similarities are at odds with hamming distance), or because they were closely related to schemes that had already been considered (Seriol is related to COB, and Overlap to slot coding). Although direct comparisons with previous codes would be hindered by differences in degrees of freedom—Spatial Coding and Seriol have more degrees of freedom because they both use activation gradients and edge marker parameters—it is still of interest to assess how the latest versions of these schemes behave on our benchmark.

The bottom panel in Table 6 shows the performances of Spatial Coding and Seriol (see Appendix C for detailed values). It is clear that Spatial Coding has the upper hand, satisfying all our constraints, but stability (see Appendix C, repeated letter condition 3). However, edge effects occur in Spatial Coding by virtue of an ''end letter marker'' parameter that increases the importance of edge letters in similarity calculations. More impressive is the coverage of transposition constraints: Spatial Coding manages to satisfy single, distant, global, and compound TL constraints at the same time. This is particularly impressive considering the failure of previous codes to account for distant TL—most of the time because they wrongly deem single substitution to be more disruptive—and provides a good occasion to gain some insights into Spatial Coding's inner workings. Recall that in Spatial Coding a letter only contributes in similarity scores if it is shared by both strings, and the contribution is then based on the misalignment in absolute letter position across strings—the difference between gradient values. Hence, single substitution only results in one missing contribution but maximal

contributions from all other letters, whereas in distant TL there is no missing contribution but two letters undercontribute since they are misaligned. In general, all gradients such that twice a TL contribution remains inferior to a full contribution will satisfy this part of the TL constraint. This holds for the gradient used in Davis (in press), but scaling it by half would break the TL constraint—giving similarities for distant TL and single substitution of 0.9 and 0.88, respectively . Similar sensitivities arise for other conditions, prompting one to ask how the activation gradient is learned in Spatial Coding. The question of learning is touched upon in Davis (in press), where it is assumed that in the full visual word recognition model, the gradient would be in phase with connection weight values learned by self-organizing mechanisms. What Spatial Coding seems to be telling us is that whereas an absolute scheme can indeed account for all TL constraints, these are by no means inherent to the scheme itself but specific to the gradient, which in turn will depend on the learning environment. Thus, at the very least, the Spatial Coding account raises doubts as to the universality of TL constraints.

Although Seriol performs less well than Spatial Coding, it is on a par with the best code previously studied (holographic UOB). Seriol is a bigram-based scheme that also uses an activation gradient, but where values do not serve as position markers and are only determined by the contiguity of letter constituents (zero-gap = 1, one-gap = 0.7, two-gap = 0.5). Seriol also introduces edge bigrams, and thus the absence of edge effects might come as a surprise. This is because a substitution at the edge leaves two one-gap bigrams and one two-gap bigram unchanged (contributing $2 \times 0.7 \times 0.7 + 1 \times 0.5 \times 0.5 = 1.48$ to the final similarity), whereas a substitution in medial position leaves one one-gap bigram and two two-gap bigrams unchanged (contributing $0.98 = 2 \times 0.5 \times 0.5 + 1 \times 0.7 \times 0.7$ to the final similarity). This holds with or without edge bigrams, as their contribution is factored out across conditions. Hence, to obtain edge effects, the gradient should assign higher values to two-gap than to one-gap bigrams, a move that is not easily interpreted. However, like Spatial Coding and unlike all other codes, Seriol manages to account for distant TL. In the case of Seriol though, this arises because of similar trade-offs taking place between one-gap and two-gap contributions, which further involve contributions from cross-terms because TL turns one-gap bigrams into two-gap bigrams. Apart from gradient values, this kind of account is sensitive to the authorized letter gap between bigrams, as for instance allowing for three-gap bigrams with 0.5 activation values would break the distant TL constraint in Seriol (giving similarities for distant TL and single substitution of 0.7 and 0.68, respectively).

In conclusion, Spatial Coding and Seriol perform very well on our benchmark of constraints, although the former has the upper hand. Interestingly, in both cases, the introduction of a gradient proves sufficient to account for the TL constraint that resisted all previous codes. Nevertheless, using gradients also implies dealing with more degrees of freedom and we have argued that to avoid the arbitrary fitting that can ensue, the question of how this gradient is learned must be addressed.

4.2. Below and above holographic string encoding

In all the codes we have studied, letter vectors were assumed to be orthogonal. However, it is clear that some letters (like ''C'' and ''O'') share more visual features than others.

Introducing these visual features proves straightforward in a holographic implementation, because one only needs to generate vectors for as many visual features and 2D-positions as required. Letters can then be created by X-oring visual features to adequate 2D-position vectors and chunking these together.

Using these codes in a connectionist network would be the next step in specifying a lexical access model. As distributed string encoding scheme, holographic codes can directly serve as input for PDP models (Plaut et al., 1996; Seidenberg & McClelland, 1989), or unsupervised network such as self-organizing maps or hopfield networks. Such an extension has been studied in Hannagan (2009) and shows that the joint use of holographic representations and attractor networks is not only possible but also fruitful. Indeed, we found that holographic string codes were sufficiently distant from one another to allow for efficient retrieval and storage under noisy conditions, and they supported an interpretation of frequency effects as emerging from noise during learning.

It might also be of interest to apply holographic coding to sequences of symbols or digits, considering the evidence that their encoding recruits at least partially overlapping mechanisms (Carreiras, Dunabeitia, & Perea, 2007; Tydgat & Grainger, 2009). Indeed, there is nothing specific to letter strings in the holographic codes we have presented, and our similarity measures should also hold for strings of digits and symbols. Thus, according to the holographic panel in Table 6, the fact that edge effects are still observed for digits but not for symbols (Tydgat & Grainger, 2009) would suggest that these do not use the same scheme. We speculate that the same holographic coding system might resort to an absolute scheme such as slot coding for symbols, because symbol combinations do not appear regularly in the visual environment, but might use a relative scheme such as UOB in the case of letters and digits, because they appear more frequently in short combinations.

4.3. Limits of holographic string encoding

One limitation of this holographic coding approach is the ill-defined Maj operator: because the representations we use are binary, ties can arise for even number of arguments that need to be broken arbitrarily. This parity issue is inherent to bipolar holographic codes—it is the price we pay for simplicity—and it should be noted that it disappears in the continuous case with real valued vectors. Other systems using bipolar codes, such as Hopfield networks, are also afflicted with the same problem but being dynamic they can resort to breaking the tie in favor of the value at $t - 1$. The same solution would work for dynamic systems based on binary holographic coding. Finally, in the static case the parity issue can be circumvented in situations where different arguments of the Maj ought to be given different strength, by generalizing the Maj operator to a weighted Maj (see Appendix A).

Although we have argued that holographic codes endorse more biological plausibility because they are distributed and support constituent structure, we have not discussed how the brain would actually compute them. Hence, this study cannot substitute to a computational model of string encoding, which needs to be specified in future work—possibly using randomly connected sigma–pi neurons as suggested elsewhere (Plate, 2000). Nevertheless,

Hannagan et al. (in press) recently showed that holographic slot coding with correlated location vectors captures very sharply the computations performed by a location invariant feed-forward network trained by backpropagation. This correspondence holds between holographic letter/location bindings and input-to-hidden weight vectors, and stems from broken patterns of symmetries in network connection weights. That holographic codes can emerge from a well-known learning algorithm and with a very simple network structure certainly brings us closer to understanding how they could be implemented neurally.

One might also ask how these results would carry over to sparser holographic codes. Indeed, in this article, we have used a dense code for simplicity ($p(1) = p(-1) = 1/2$), but as noted earlier, the code can be made as sparse as required (Kanerva, 1995). As in the process the X-or operator is not affected and the Maj operator only lowers its threshold, we expect that the simulations presented here will not vary drastically with the density parameter. In particular, emerging TL and edge effects would be anticipated to diminish, but not to disappear, as they are driven by the nature of both holographic operators and not their thresholds.

Finally, a limitation of the string encoding approach in general, and thus of the holographic codes we studied here, is the silent treatment of phonology. Indeed, the extent to which phonological representations interact with visual ones during string encoding is not at all considered (Ziegler, 1995). We would like to emphasize, however, that phonological information can easily be introduced in static holographic code, either indirectly as a unit selection mechanism, or directly by creating phonological vectors and introducing them adequately in the codes.

5. Conclusion

In this article, we have proposed a string encoding approach that uses distributed representations and takes into account constituent structure. We used the BSC, a special case of holographic reduced representations designed to capture and manipulate combinatorial structures.

We have assessed these codes against their localist counterparts using several different schemes and a number of criteria coming from masked priming experiments. Our first conclusion is that schemes do not behave in the same way whether they are implemented using localist or holographic codes. This implies that a given scheme (e.g., slot coding or open-bigrams) cannot be ruled out simply because one implementation of it fails to satisfy a constraint (resp. single TL and repeated RP), as this only falsifies one possible implementation (localist). Our results show that to falsify a string encoding scheme properly, its distributed implementations should also be considered.

Our second conclusion is that with one exception, using holographic representations never jeopardizes the qualities observed in the localist case, and on the contrary holographic coding can show a number of desirable emerging properties. Particularly surprising is the apparition of edge effects in UOB and LCD, and of transposition effects in slot coding. We have explained these phenomena in detail, formally in the case of transposition, and empiri-

cally in the case of edge effects. Among holographic codes, we found that UOB could account for a large number of effects (stability, edge effects, relative position priming with or without repetitions, local transposition) while at the same time being more parsimonious than LCD in hypotheses.

More work is required to understand the properties and limits of these codes, but they are laid in a mathematical framework, which encourages formal analysis. Future research could investigate the performance of holographic codes that use gradients, correlated position vectors, phonological information, or frequency as a unit selection mechanism. Word codes can also be made more sophisticated by building on arbitrarily detailed visual feature codes, and the holographic approach as a whole can be extended to digit or symbol processing.

Notes

1. In this article, we use *string encoding* and *letter position coding* as interchangeable terms. Although the latter is the authorized one, our preference goes to the former because it is less misleading: As we will see, schemes do not always encode order at the letter level.
2. Although we will later discuss redundant localist coding, which defines a special many-to-one relationship.
3. The probability that two different strings actually activate exactly the same set is very low with 400 units, and decreases exponentially with this number.
4. We use the standard letter position notation, which replaces letters by their absolute position, and where the letter ''d'' in the prime means a different letter is used relatively to the target.
5. The chunking operator is not always constant across all binary spatter codes. Here, we use a deterministic version of the original majority rule.
6. In this article, we use a similarity measure based on the normalized Hamming distance h between two vectors A and B, and defined as $sim(A,B) = 1 - 2|h(A,B)|$.
7. This similarity decreases as the number of argument vectors grows.
8. This condition should produce the least similarity among all conditions tested.
9. Current specification, yet unpublished.
10. This result has been disputed by Lupker and Davis (2009) on the grounds that global TL priming can be obtained using an alternative ''sandwich'' priming paradigm.
11. When each vector correlates with the previous one, the resulting code becomes the holographic pendant of the overlap model.
12. However, it has been argued that the saliency gradient obtained in Gomez et al. (2008) is incompatible with such a modification.
13. There was a mistake in Schoonbaert and Grainger (2004), where 10 bigrams were believed to be shared in this condition.
14. We thank one anonymous reviewer for pointing this out to us.

Acknowledgments

References

Abbott, L., & Sejnowski, T. (1999). *Neural codes and distributed representations*Cambridge, MA: MIT Press.

Bowers, J. S. (2009). On the biological plausibility of grandmother cells: Implications for neural network theories in psychology and neuroscience. *Psychological Review*, *116*(1), 220–251.

Carreiras, M., Dunabeitia, J. A., & Perea, M. (2007). Reading words, number and symbol. *Trends in Cognitive Science*, *11*(11), 454–455.

Chomsky, N. (2002). *Syntactic structures* (2nd ed.). New York: Walter de Gruyter.

Cohen, L., & Dehaene, S. (2004). Specialization within the ventral stream: The case for the visual word form area. *NeuroImage*, *22*, 466–476.

Cohen, L., Dehaene, S., Naccache, L., Lehéricy, S., Dehaene-Lambertz, G., Hénaff, M., & Michel, F. (2000). The visual word-form area: Spatial and temporal characterization of an initial stage of reading in normal subjects and posterior split-brain patients. *Brain*, *123*, 291–307.

Coltheart, M., Rastle, K., Perry, C., Langdon, R., & Ziegler, J. (2001). Drc: A dual route cascaded model of visual word recognition and reading aloud. *Psychological Review*, *108*, 204–256.

Dandurand, F., Grainger, J., & Dufau, S. (2010). Learning location invariant orthographic representations for printed words. *Connection Science*, *22*(1), 25–42.

Davis, C. J. (1999). The self-organising lexical acquisition and recognition (solar) model of visual word recognition. Doctoral dissertation, University of New South Wales, Sydney, New South Wales, Australia. Dissertation Abstracts International, vol. *62*, p. 594.

Davis, C. J. (in press). The spatial coding model of visual word identification. *Psychological Review*, *117*, 713–758.

Davis, C. J., & Bowers, J. S. (2006). Contrasting five theories of letter position coding. *Journal of Experimental Psychology: Human Perception and Performance*, *32*(2), 535–557.

Dehaene, S., Cohen, L., Sigman, M., & Vinckier, F. (2005). The neural code for written words: a proposal. *Trends in Cognitive Science*, *9*, 335–341.

Dehaene, S., Jobert, A., Naccache, L., Ciuciu, P., Poline, J. B., LeBihan, D., et al. (2004). Letter binding and invariant recognition of masked words: Behavioral and neuroimaging evidence. *Psychological Science*, *15*(5), 307–313.

Ellis, A. W., & Lambon Ralph, M. A. (2000). Age of acquisition effects in adult lexical processing reflect loss of plasticity in maturing systems: Insights from connectionist networks. *Journal of Experimental Psychology: Learning, Memory & Cognition*, *26*, 1103–1123.

Felleman, D., & Van Essen, D. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, *1*(1), 1–47.

Fischer-Baum, S., McCloskey, M., & Rapp, B. (2010). Representation of letter position in spelling: Evidence from acquired dysgraphia. *Cognition*, *115*(3), 466–490.

Fodor, J. A. (1983). *The modularity of mind*. Cambridge, MA: The MIT Press.

Forster, K. I., & Davis, C. (1984). Repetition priming and frequency attenuation in lexical access. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*, 680–698.

Gomez, P., Ratcliff, R., & Perea, M. (2008). The overlap model: A model of letter position coding. *Psychological Review*, *115*(3), 577–600.

Grainger, J., Granier, J., Farioli, F., Van Assche, E., & Van Heuven, W. (2006). Letter position information and printed word perception: The relative-position priming constraint. *Journal of Experimental Psychology: Human Perception and Performance*, *32*, 865–884.

Grainger, J., & Jacobs, A. M. (1993). Masked partial-word priming in visual word recognition: Effects of positional letter frequency. *Journal of Experimental Psychology: Human Perception and Performance*, *19*(5), 951–964.

Grainger, J., & Jacobs, A. M. (1996). Orthographic processing in visual word recognition: A multiple read-out model. *Psychological Review*, *103*, 518–565.

Grainger, J., & Van Heuven, W. J. B. (2003). Modeling letter position coding in printed word perception. In P. Bonin (Ed.), *Mental lexicon: ''Some words to talk about words''* (pp. 1–23), New York: Nova Science.

Gross, C. G. (2002). Genealogy of the ''grandmother cell''. *Neuroscientist*, *8*, 512–518.

Guerrera, C., & Forster, K. I. (2008). Masked form priming with extreme transposition. Cracking the Orthographic Code. *A special issue of Language and Cognitive Processes*, *23*, 117–142.

Hannagan, T. (2009). Visual word recognition: Holographic representations and attractor networks. Unpublished doctoral dissertation (p. 6). Paris: Université Paris.

Hannagan, T. (2010). Match calculations and priming simulations: The case of spatial coding. Available at: http://www.lscp.net/persons/hannagan/Material/MatchVsSim.pdf.

Hannagan, T., Dandurand, F., & Grainger, J. (in press). Broken symmetries in a location invariant word recognition network. *Neural Computation*.

Harris, H. (2002). Holographic reduced representations for oscillator recall: A model of phonological production. In W. D. Gray & C. D. Schunn (Eds.), *The Proceedings of the 24th Annual Meeting of the Cognitive Science Society* (pp. 423–428). Mahwah, NJ: Lawrence Erlbaum Associates.

Humphreys, G. W., Evett, L. J., & Quinlan, P. T. (1990). Orthographic processing in visual word identification. *Cognitive Psychology*, *22*, 517–560.

Jacobs, A. M., Rey, A., Ziegler, J. C., & Grainger, J. (1998). Mrom-p: An interactive activation, multiple readout model of orthographic and phonological processes in visual word recognition. In J. Grainger & A. M. Jacobs (Eds.), *Localist connectionist approaches to human cognition* (pp. 147–188). Mahwah, NJ: Erlbaum.

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and information in a composite holographic lexicon. *Psychological Review*, *114*, 1–37.

Jordan, T., Thomas, S., Patching, G., & Scott-Brown, K. (2003). Assessing the importance of letter pairs in initial, exterior and interior position in reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *29*, 883–893.

Kanerva, P. (1995). A family of binary spatter codes. In F. Fogelman-Soulié and P. Gallineri (Eds.), *ICANN '95: Proceedings of the International Conference on Artificial Neural Networks* (vol. 1, pp. 517–522). Paris, France: EC2 & Cie.

Kanerva, P. (1997). Fully distributed representation. In *Proceedings of the 1997 Real World Computing Symposium (Tokyo, Japan)*, pp. 358–365.

Kanerva, P. (1998). Pattern completion with distributed representation. In *IEEE International Conference on Neural Networks (IJCNN'98), II*, pp. 1416–1421.

Knoblauch, A., & Palm, G. (2005). What is signal and what is noise in the brain? *Biosystems, 79*(1–3), 83–90.

Levy, S., & Kirby, S. (2006). Evolving distributed representations for language with self-organizing maps. In P. Vogt (Eds.), *Symbol Grounding and Beyond: Proceedings of the Third International Workshop on the Emergence and Evolution of Linguistic Communication* (pp. 57–71). Berlin: Springer.

Lupker, S. J., & Davis, C. J. (2009). Sandwich priming: A method for overcoming the limitations of masked priming by reducing lexical competitor effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *35*, 618–639.

Lupker, S. J., Perea, M., & Davis, C. J. (2008). Transposed letter priming effects: Consonants, vowels and letter frequency. *Language and Cognitive Processes*, *23*, 93–116.

McClelland, J., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights on the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*, 419–457.

McClelland, J., Patterson, K., Pinker, S., & Ullman, M. (2002). The past tense debate: Papers and replies by S. Pinker & M. Ullman and by J. McClelland & K. Patterson. *Trends in Cognitive Sciences*, *6*, 456–474.

McClelland, J., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. an account of basic findings. *Psychological Review*, *88*, 375–407.

Mozer, M. C. (1987). Early parallel processing in reading: A connectionist approach. 83–104.

Mozer, M. C. (1991). *The perception of multiple objects: A connectionist approach*. Cambridge, MA: MIT Press.

Neumann, J. (2002). Learning the systematic transformation of holographic reduced representations. *Cognitive Systems Research*, *3*(2), 227–235.

Norris, D. (2006). The Bayesian reader: Explaining word recognition as an optimal Bayesian decision process. *Psychological Review*, *113*(2), 327–357.

Perea, M., & Acha, J. (2008). The effects of length and transposed-letter similarity in lexical decision: Evidence with beginning, intermediate, and adult readers. *British Journal of Psychology*, *99*, 245–264.

Perea, M., Duñabeitia, J., & Carreiras, M. (2008). Transposed-letter priming effects for close vs. distant transpositions. *Experimental Psychology*, *55*, 397–406.

Peressotti, F., & Grainger, J. (1999). The role of letter identity and letter position in orthographic priming. *Perception and Psychophysics*, *61*, 691–706.

Perea, M., & Lupker, S. J. (2003). Does jugde activate court? Transposed-letter similarity effects in masked associative priming. *Memory and Cognition*, *31*, 829–841.

Perea, M., & Lupker, S. J. (2004). Can caniso activate casino? Transposed-letter similarity effects with nonadjacent letter positions. *Journal of Memory and Language*, *51*, 231–246.

Pinker, S. (1999). *Words and rules: The ingredients of language*. New York: Basic Books.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6* (3), 623–641.

Plate, T. A. (2000). Randomly connected sigma-pi neurons can form associative memories. *Network: Computation in neural systems*, *11*(4), 321–332.

Plaut, D. C., & McClelland, J. L. (2010). Locating object knowledge in the brain: A critique of Bowers' (2009) attempt to revive the grandmother cell hypothesis. *Psychological Review*, *117*, 284–290.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, *103*, 56–115.

Rumelhart, D. E., Hinton, G., & Williams, R. J. (1986). Learning internal representations by error propagation In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 318–362). Cambridge, MA: MIT Press.

Rumelhart, D. E., McClelland, J. L., & the P. R. Group. (Eds.). (1986). *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1. Foundations*. Cambridge, MA: MIT Press.

Schoonbaert, S., & Grainger, J. (2004). Letter position coding in printed word perception: Effects of repeated and transposed letters. *Language and Cognitive Processes*, *19*(3), 333–367.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, *96*, 523–568.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159–216.

Smolensky, P., Legendre, G., & Miyata, Y. (1992). Principles for an integrated connectionist/symbolic theory of higher cognition (Tech. Rep. No. CU-CS-600-92). Department of Computer Science and Institute of Cognitive Science, University of Colorado, Boulder.

Stevens, M., & Grainger, J. (2003). Letter visibility and the viewing position effect in visual word recognition. *Perception and Psychophysics*, *65*, 133–151.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, *49*, 433–460.

Tydgat, I., & Grainger, J. (2009). Serial position effects in the identification of letters, digits, and symbols. *Journal of Experimental Psychology: Human Perception and Performance*, *35*(2), 480–498.

Van Assche, E., & Grainger, J. (2006). A study of relative-position priming with superset primes. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *32*(2), 399–415.

Vinckier, F., Dehaene, S., Jobert, A., Dubus, J., Sigman, M., & Cohen, L. (2007). Hierarchical coding of letter strings in the ventral stream: Dissecting the inner organization of the visual word-form system. *Neuron*, *55*, 143–156.

Waydo, S., Kraskov, A., Quiroga, R. Q., Fried, I., & Koch, C. (2006). Sparse representation in the human medial temporal lobe. *Journal of Neuroscience*, *26*, 10232–10234.

Whitney, C. (2001). How the brain encodes the order of letters in a printed word: The seriol model and selective literature review. *Psychonomic Bulletin and Review*, *8*, 221–243.

Whitney, C. (2004). *Investigations into the neural basis of structured representations*. Unpublished doctoral dissertation, University of Maryland.

Whitney, C. (2008). Comparison of the seriol and solar theories of letter position encoding. *Brain and Language*, *107*(2), 170–178.

Whitney, C., & Berndt, R. S. (1999). A new model of letter string encoding: Simulating right neglect dyslexia. In J. A. Reggia, E. Ruppin, and D. Glanzman (Eds.), *Progress in brain research* (vol. 121, pp. 143–163). Amsterdam: Elsevier.

Wickelgren, W. A. (1969). Auditory or articulatory coding in verbal short-term memory. *Psychological Review*, *76*, 232–235.

Ziegler, J. C. (1995). Phonological information provides early sources of constraint in the processing of letter strings. *Journal of Memory and Language*, *34*(5), 567–593.

Appendix A: Operators: Formal definition

X-or

Let A_1, A_2 be bipolar variables. The binding operator X-or is a dyadic operator from $\{-1,1\}^2$ to $\{-1,1\}$ and is defined as:

$$\text{X-or}(A_1, A_2) := A_1 \otimes A_2 = \begin{cases} -1, & \text{if } A_1 \cdot A_2 = 1 \\ 1, & \text{if } A_1 \cdot A_2 = -1 \end{cases}$$

Majority rule

Here we use a deterministic version of the original Maj operator introduced in Kanerva (1997):

Let $(A_p)_{p=1,\ldots,k}$ be k bipolar variables. The chunking operator Maj is defined as:

$$\forall i, \text{Maj}(A_1^i, \ldots, A_k^i) := \bigoplus_{p=1}^k A_p^i = \begin{cases} -1, & \text{if } \sum_{p=1}^k A_p^i < 0 \\ 1, & \text{if } \sum_{p=1}^k A_p^i > 0 \\ T, & \text{if } \sum_{p=1}^k A_p^i = 0 \end{cases}$$

where $T = A_1^{(i+1)\text{mod}(n)} \cdot A_k^{(i+1)\text{mod}(n)}$

The weighted Maj version used in weighted holographic slot coding is given by:

Let $(A_p)_{p=1,\ldots,k}$ be k bipolar variables and $(\alpha_p)_{p=1,\ldots,k} \in \Re$.

$$\text{WMaj}((A_1, \ldots, A_k), (\alpha_1, \ldots, \alpha_k)) := \bigoplus_{p=1}^{k} \alpha_p A_p = \begin{cases} -1, & \text{if } \sum_{p=1}^{k} \alpha_p A_p < 0 \\ 1, & \text{if } \sum_{p=1}^{k} \alpha_p A_p > 0 \\ T, & \text{if } \sum_{p=1}^{k} \alpha_p A_p = 0 \end{cases}$$

where T is an independent random bipolar variable following a Bernoulli distribution of parameter 1/2.

Appendix B: Transposition effects in BSCs

In this appendix, we show why transposition effects arise in absolute coding schemes using BSCs. We restrict our proof to structures of k odd arguments, similar mechanisms being at work in the even case.

$\forall k \in \mathbb{N}, k \text{ odd}, \forall j = 1, \ldots, k, \text{let}$

$$S = A_1 \otimes X_1 \oplus A_2 \otimes X_2 \oplus \cdots \oplus A_k \otimes X_k$$

and

$$T = A_2 \otimes X_1 \oplus A_1 \otimes X_2 \oplus \cdots \oplus A_k \otimes X_k$$

be two BSC vectors related by transposition of A_1 with A_2. We first compute the expected Hamming distance H between S and T:

$$E[H(S, T)] = P\left[A_1 X_1 + A_2 X_2 < A_2 X_1 + A_1 X_2, \sum_{i=3}^{k} A_i X_i = 1\right]$$

$$+ P\left[A_1 X_1 + A_2 X_2 > A_2 X_1 + A_1 X_2, \sum_{i=3}^{k} A_i X_i = -1\right]$$

$$= P[A_1 X_1 + A_2 X_2 < A_2 X_1 + A_1 X_2] \cdot P\left[\sum_{i=3}^{k} A_i X_i = 1\right]$$

$$+ P[A_1 X_1 + A_2 X_2 > A_2 X_1 + A_1 X_2] \cdot P\left[\sum_{i=3}^{k} A_i X_i = -1\right]$$

$$= 2 \cdot P[A_1 = -A_2] \cdot P[X_1 = -X_2] \cdot P\left[\sum_{i=3}^{k} A_i X_i = 1\right]$$

$$= \frac{1}{2} \cdot P\left[\sum_{i=1}^{k-2} T_i = 1\right], \text{ where } T_i = A_i X_i \text{ i.i.d.}$$

$$= \frac{1}{2} \cdot P\left[\sum_{i=1}^{k-2} \frac{T_i + 1}{2} = \frac{k-1}{2}\right], \text{ where } \frac{T_i + 1}{2} \text{ follows a Bernoulli } p = \frac{1}{2}.$$

$$= C_{\frac{k-1}{2}}^{k-2} \cdot \frac{1}{2^{k-1}}$$

Let now $U = D_1 \otimes X_1 \oplus D_2 \otimes X_2 \oplus \cdots \oplus A_k \otimes X_k$.

From Kanerva's formula on the distances between two BSCs sharing k arguments (Kanerva, 1995), we get:

$$E[H(S, U)] = \frac{3}{2} \cdot C_{\frac{k-1}{2}}^{k-2} \cdot \frac{1}{2^{k-1}}$$

$$= \frac{3}{2} \cdot E[H(S, T)]$$

Finally, to obtain the transposition effect we need only to subtract both distances, and remember the relationship with similarities:

$$E[\text{Sim}(S, T)] - E[\text{Sim}(S, U)] = (1 - 2 \cdot E[H(S, T)]) - (1 - 2 \cdot E[H(S, U)])$$

$$= E[H(S, T)]$$

$$= C_{\frac{k-1}{2}}^{k-2} \cdot \frac{1}{2^{k-1}}$$

This shows that using the original BSCs, the transposition effect is maximum for strings of size 3 (at 0.25), and then decreases convexly with size.

Appendix C: Similarity measures for Seriol and Spatial Coding

The following table presents the similarity values obtained for Spatial Coding and Seriol, used in Table 6.

Table C1
Similarities for Spatial Coding (with end letter markers) and Seriol coding (with edge bigrams)

Condition	Stability (1–8)								Edge Effects (9–11)		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Prime	12345	1245	123345	123d45	12dd5	1d345	12d456	12d4d6	d2345	12d45	1234d
Target	12345	12345	12345	12345	12345	12345	123456	123456	12345	12345	12345
Spatial Coding	**1.00**	**0.79**	**1.00**	**0.92**	**0.71**	**0.86**	**0.88**	**0.75**	**0.71**	**0.86**	**0.71**
Seriol	**1.00**	**0.68**	**0.92**	**0.86**	**0.41**	**0.66**	**0.67**	**0.44**	**0.66**	**0.63**	**0.66**

Table C1
(Continued)

Condition	Transposition (12–15)				Relative Position (16–20)				
	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
Prime	12435	21436587	125436	13d45	12345	34567	13457	123267	123567
Target	12345	12345678	123456	12345	1234567	1234567	1234567	1232567	1232567
Spatial coding	**0.89**	**0.48**	**0.78**	**0.81**	**0.67**	**0.67**	**0.69**	**0.82**	**0.81**
Seriol	**0.85**	**0.45**	**0.64**	**0.63**	**0.61**	**0.61**	**0.58**	**0.77**	**0.83**

Note. Boldface type indicates similarities.

Supporting Information

Additional Supporting Information may be found in the online version of this article on Wiley InterScience:

Appendix S1. Test items: Exposure items.

Appendix S2. Test items: Novel items.

Please note: Wiley-Blackwell is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.