Optimization and Quantization in Gradient Symbol Systems: A Framework for Integrating the Continuous and the Discrete in Cognition

Paul Smolensky*, Matthew Goldrick†, and Donald Mathis*
*Department of Cognitive Science, Johns Hopkins University
†Department of Linguistics, Northwestern University

Abstract

Mental representations have continuous as well as discrete, combinatorial properties. For example, while predominantly discrete, phonological representations also vary continuously; this is reflected by gradient effects in instrumental studies of speech production. Can an integrated theoretical framework address both aspects of structure? The framework we introduce here, Gradient Symbol Processing, characterizes the emergence of grammatical macrostructure from the Parallel Distributed Processing microstructure (McClelland & Rumelhart, 1986) of language processing. The mental representations that emerge, Distributed Symbol Systems, have both combinatorial and gradient structure. They are processed through Subsymbolic Optimization-Quantization, in which an optimization process favoring representations that satisfy well-formedness constraints operates in parallel with a distributed quantization process favoring discrete symbolic structures. We apply a particular instantiation of this framework, λ-Diffusion Theory, to phonological production. Simulations of the resulting model suggest that Gradient Symbol Processing offers a way to unify accounts of grammatical competence with both discrete and continuous patterns in language performance.

The work discussed here was developed as one path for carrying out a research program that was already sketched by 1986[1]:

(1) A PDP approach to cognitive macrostructure

> "The basic perspective of this book is that many of the constructs of macrolevel descriptions … can be viewed as emerging out of interactions of the microstructure of distributed models. … although we imagine that rule-based models of language acquisition … may all be more or less valid approximate macrostructural descriptions,

[1] Important precedents include Hofstadter (1979, 1985). Other approaches to combining continuous activation spreading and symbolic structure, but without distributed representations (in the sense used here), include the ACT systems (Anderson & Lebiere, 1998), the LISA model (Hummel & Holyoak, 2003) and a range of hybrid architectures (Wermter & Sun, 2000).

> we believe that the actual algorithms involved cannot be represented precisely in any of those macrotheories.
>
> … as we develop clearer understandings of the microlevel models, we may wish to formulate rather different macrolevel models … PDP mechanisms provide a powerful alternative set of macrolevel primitives … [e.g.,] "Relax into a state that represents an optimal global interpretation of the current input." (Rumelhart & McClelland 1986b: 125–126)

The final sentence of this quotation states the first aspect of the microstructure that is critical for the work presented here. The other crucial microstructural feature is the PDP principle that representations are distributed patterns of activation. Our aim is to make mathematically precise the emergence of symbolic cognitive macrostructure from such microstructure. The sense of 'emergence' relevant here is that the new ('emergent') properties of the macrostructure are formally entailed by the basic properties of the microstructure; we do not refer to emergence through learning, and indeed the contributions of learning to emergence play no role in the work reported in this article.

The psychological reality of symbolic rules and representations is of course an issue that divides the field. Hypotheses range from the eliminativist extreme, which denies any degree of cognitive reality to symbolic descriptions, to the implementationalist extreme, which takes symbolic descriptions to be virtually exact accounts of internal cognitive function. The working hypothesis guiding the research program reported here is intermediate: (2).

(2) GSPH: the Gradient Symbol Processing Hypothesis (for grammar)
   a. Symbolic grammatical theory provides good approximations to the macrostructure of internal cognitive representations and functions. Psycholinguistic theory can profit by directly exploiting the detailed insights provided by grammatical theory.
   b. Symbolic macro-descriptions are in need of microlevel representations and algorithms, and of the macrolevel improvements that derive from this microstructure.

This is not the place to make a general theoretical and empirical case for the value of the GSPH (but see Goldrick, Baker, Murphy & Baese-Berk, 2011; Smolensky & Legendre, 2006). Our goal here is only to present a few of the results following from the GSPH, results that we believe offer new insights into the interaction between the symbolic and the gradient in cognition. Our descriptions of phenomena and mechanisms should all be understood as being prefaced by "according to the GSPH"— we of course recognize that our descriptions do not generally express consensus views.

Within the particular domain we focus upon here, phonological production, proposals span the spectrum from traditional implementationalist frameworks to quite eliminativist approaches (e.g., Gafos & Benus, 2006; Port & Leary, 2005). Multiple intermediate positions are also represented. The Articulatory Phonology framework (Browman & Goldstein, 1992; Davidson, 2006) deploys a number of continuous variables evolving in continuous time, along with discrete components (e.g., sets of gestures, landmarks for intergestural timing).

There are also hybrid approaches which allow both discrete morphological and phonological symbolic representations to directly interact with continuous phonetic representations (e.g., phonetic exemplars; see Pierrehumbert, 2006, for a review). Our proposal maintains a clear separation between phonological and phonetic representations (allowing only the former to interact with morphological representations), and is unique in incorporating gradience within symbolic phonological representations, as well as within non-symbolic phonetic representations.

We emphasize that we regard the GSPH truly as a working hypothesis. We would welcome future results showing how some of the structure that we must now simply assume to be in place can arise from learning, or results that show how to characterize, with some formal precision, a macrostructure that possesses the functional capabilities of our architecture but which deviates more significantly from symbolic theory. Work such as Plaut, McClelland, Seidenberg, & Patterson (1996) is quite promising on both accounts, and formal connections to the approach presented here would be extremely valuable.

Our topic, the emergence of macrostructure, has been a main theme in the work of Jay McClelland. We view our approach as fundamentally consistent with his, but complementary. McClelland's approach assigns preeminent importance to gradience, with approximately discrete symbolic structure emerging in particular cognitive contexts. As discussed below, our view is that symbolic combinatorial structure provides a highly productive framework for developing theories of cognition. Our approach is therefore to start with systems utilizing discrete symbolic constituents and incorporate gradience as required by the data. We anticipate that the two approaches will eventually meet somewhere in a middle ground where the discrete and the continuous interact in a rich a constructive fashion.

1. Introduction to Gradient Symbol Processing and phonological production

Our Gradient Symbol Processing Hypothesis (2) asserts that, to a good approximation, mental representations have a crucial property: they are systematic, structured combinations of discrete constituents (Fodor & Pylyshyn, 1988; Pylyshyn, 1984). Our work on the emergence of macro- from microstructure aims to address the question: *How do the continuous and the discrete combinatorial aspects of mental representation interact?* This question looms large in many domains of higher cognition. Two illustrative issues in language are given in (3).

(3) Discrete/continuous interaction: Examples in language (elaborated in Section 4.3)
   a. In phonological encoding (mapping lexical /roz+z/ 'ROSE+PL' to phonological "roses"), continuous activation-spreading computes outputs that are, to a good approximation, structured combinations of discrete speech sounds (or *segments*)—but speech error data reveal that these outputs are also gradient in subtle ways. Can these two aspects be accounted for within a single integrated architecture?
   b. In many arenas of linguistic performance, continuous variables such as frequency and similarity interact strongly with discrete grammatical structure. Can we derive such interaction from the cognitive microstructure of grammar?

A broad survey of cases of (3b) can be found in Bybee & McClelland (2005), which stresses the importance of the continuous variables, but discusses their influences within and upon the structural elements of grammars. McClelland & Vander Wyk (2006) provide a detailed study of cases of (3b) in the phonological grammar of English. Another type of strong interaction is gradience in the degree of grammatical productivity discussed in McClelland & Bybee (2007).

For concreteness, most of our discussion will focus on the representations within, and interaction between, two components proposed in the architectures of spoken language processing assumed by many researchers: *lexical processing* and *phonological encoding* (Dell, 1986; Garrett, 1975; Goldrick & Rapp, 2007; Levelt, Roelofs, & Meyer, 1999; Stemberger, 1985).

In these architectures, the state of the lexical component is assumed to be well approximated by a combinatorial representation composed of the stored sound structures of a set of morphemes chosen by a speaker to communicate a meaning—e.g., /roz/+/z/ for ROSE + PLURAL (slash-delimiters mark lexical representations; in generative phonological theory, these are termed 'underlying representations'). The state of the phonological component is well approximated by a combinatorial representation composed of a multi-set of phonological segments related in a particular order and grouped into constituents such as syllables [σ], stress feet (Ft) and prosodic words [PrWd]—e.g., [$_{\text{PrWd}}$ ($_{\text{Ft}}$ [$_{\sigma}$ró][$_{\sigma}$zəz])] 'roses' (square brackets denote phonological representations; in generative phonological theory, these are termed 'surface representations': Smolensky, Legendre, & Tesar, (2006:473–480) gives a mini-tutorial). This combinatorial representation serves as input to subsequent *phonetic processes* that compute the articulator trajectories corresponding to these symbolic phonological representations.

Both the lexical and phonological representations are discrete—to a good approximation. We shall see, however, that subtle gradient (i.e., non-discrete) effects are at work. For example, gradient differences in phonological representations give rise to small but systematic differences in the continuous representations of phonetics (e.g., slightly different voice onset times for consonants: Section 4.3).

In considering the relation between components of the cognitive system, we focus on relatively small time scales. For example, in the context of lexical and phonological processing, we consider a buffer of sufficiently modest size that it is a reasonable approximation to assume that the morphemes it contains are processed in parallel when computing the phonological representation. One parallel step of input-to-output mapping constitutes a single *relaxation* (or *settling*) of a component—as in (1). In production, the parallel activation of phonological elements from multiple morphemes is revealed by speech errors that anticipate phonological elements from other (nearby) morphemes (Dell, 1986). Similarly, in spoken word perception, listeners persist in representing ambiguous speech sounds over many segments; they do not commit to a single parse of the input until sufficient information is received (McMurray, Tanenhaus, & Aslin, 2009). (The integration of these parallel computations with longer-term serial processing is an important issue for future work.)

Pursuing the overall approach sketched in (1), we treat the discrete, symbolic, combinatorial characterizations of the inputs and outputs of a cognitive process such as phonological encoding as higher-level approximate descriptions of patterns of activity in a connectionist network: the macrostructure of the system is symbolic, the microstructure is PDP (see Figure 1). In the *Gradient Symbol Processing framework* that we present here, processing consists in continuous movement in a continuous state space of distributed activation patterns, a discrete subset of which constitutes the realizations of symbol structures. To produce an appropriately discrete output by the end of a relaxation, this continuous dynamics must end up (approximately) at one of these special points.

Although the representational space for phonological encoding is continuous, it is phonological, *not* phonetic. Continuous articulatory and acoustic dimensions are encoded in other representational spaces. We will return to this important point.

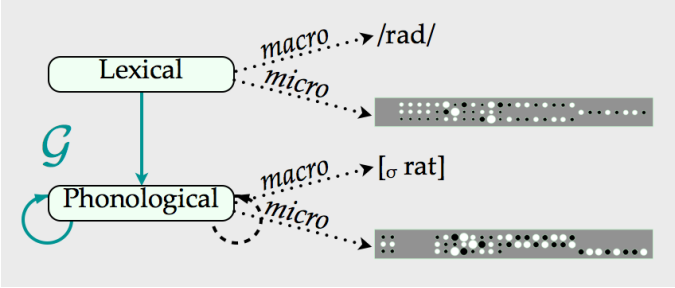


Figure 1. One parallel step of processing—one relaxation—in phonological encoding (German *Rad* 'wheel'). Input and output representations are Distributed Symbol Structures characterized at both macro- and microlevels. Evaluation (solid arrows) and quantization (dashed arrow) dynamics perform Gradient Symbol Processing.

Ignoring for a moment the connections drawn with a dashed arrow, Figure 1 indicates that there are feed-forward connections from the group of connectionist units hosting the lexical representation to that hosting the phonological representation. These, together with a set of recurrent connections among the phonological units, constitute the phonological grammar $\mathcal{G}$, in the following precise sense. If the pattern of activation over the lexical units is the discrete point in state space that is described symbolically as, say, /rad/—the German lexical form for *Rad* 'wheel'—then the solid connections will drive the phonological units towards the pattern of activity which is the discrete state described (simplifying) as [rat], the phonological form that the grammar $\mathcal{G}$ specifies as the grammatical pronunciation of *Rad.* (In isolation, this morpheme is pronounced with a final [t]; in other contexts, the corresponding segment is pronounced as [d]. This is German *final voicing neutralization*.)

The dashed arrow in Figure 1 indicates another set of recurrent connections among the phonological units: they drive the phonological units to the discrete set of states that have a combinatorial symbolic description. This is the technical core of the new contributions of the work reported here. (The remaining techniques were presented as the general Integrated Connectionist/Symbolic cognitive architecture in Smolensky & Legendre, 2006; pointers are given throughout the article.) The proposed theory of the dynamics these connections create is presented in Section 3. The need for such a dynamics is argued in Section 2, which

formulates the general computational framework of Gradient Symbol Processing. This architecture employs two functionally distinct but highly interdependent processes: *evaluation* of a continuum of alternative outputs, and *quantization* of this continuum so as to approximate a single discrete combinatorial structure as output (ideally, the best-evaluated—i.e., *optimal*—one). Empirical tests of the theory via specific simple models are discussed in Section 4.

In greater detail, the structure of the argument and the paper is summarized in (4), which can be consulted as the argument is developed.

(4) Synopsis of the argument
- a. Introducing the general approach: Gradient Symbol Processing (Section 1)
- b. Deriving the framework: Subsymbolic Optimization/Quantization (Section 2)
  - i. Psycholinguistic theory requires a discrete combinatorial macrolevel. (2.1)
  - ii. Psycholinguistic theory requires continuous similarity structure. (2.2–2.3)
  - iii. Hypothesis: Similarity is computed from vectorial encodings … (2.4)
  - iv. … that are distributed activation patterns at the microlevel. (2.5–2.5.2)
  - v. Psycholinguistic theory requires partial activation of constituents: blends. (2.6)
  - vi. Components of the mental architecture must produce (approximately) discrete output to resolve the ambiguity in blends of combinatorial structures. (2.7)
- c. Instantiating the framework: λ-Diffusion Theory (Section 3)
  - i. A quantization process creates attractors at pure symbolic output states. (3.1)
  - ii. Grammars optimize well-formedness: macro-Harmony. (3.2)
  - iii. Networks optimize well-formedness: micro-Harmony. (3.3)
  - iv. Networks compute grammatical representations. (3.4)
  - v. Optimization and quantization dynamics must operate in parallel to solve the Problem of Mutually Dependent Choices. (3.5)
- d. Instantiating the theory: Modeling phonological production (Section 4)
  Simulations suggest that λ-Diffusion Theory can:
  - i. solve the Problem of Mutually Dependent Choices; (4.1)
  - ii. provide insight into discrete and continuous phenomena in competence; (4.2)
  - iii. provide insight into discrete and continuous phenomena in performance: speech errors. (4.3)

2. Discreteness and continuity of mental representations

Our first task is to computationally integrate two facets of mental representations in higher cognitive domains such as phonological production: discrete *combinatorial* structure and continuous *similarity* structure.

2.1. Combinatorial structure

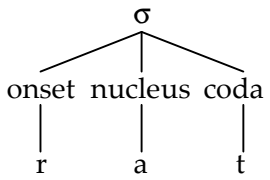
The GSPH (2) asserts that mental representations are systematic, structured combinations of constituent representations. According to many phonological theories, for example, the mental representation of the syllable (σ) that is the pronunciation of *Rad* 'wheel' in German can be described (simplifying) as in (5b), which we'll abbreviate as in (5a). Each constituent

can be analyzed as a *structural role* instantiated by a *filler* (5c) (Minsky, 1975; Rumelhart, 1975). The constituents of a given representation are connected via the fundamental combinatory operation of symbolic computation, *concatenation* (Partee, ter Meulen, & Wall 1990:432). Crucially for us, by adopting a *filler/role decomposition*, the representation can be viewed as an unordered set of *filler/role bindings* (5d) (Newell, 1980:142; Smolensky, 1990).

(5) Combinatorial structure (simplified) of a syllable σ in four equivalent notations

a. $[_{\sigma}$ rat]

b.



c. Constituents: roles and fillers

role	*filler*
σ-onset	r
σ-nucleus	a
σ-coda	t

d. Filler/role bindings: {a/σ-nucleus, t/σ-coda, r/σ-onset}

2.2. Similarity structure

Similarity of representations is a central psychological concept, used to explain many cognitive phenomena; a few examples are given in (6).

(6) Similarity-based psychological explanation: examples (with recent reviews)

a. Errors: the more similar an error response *E* is to the correct form, the more likely *E* (Goldrick, 2008).
b. Categorization: the more similar an item *X* is to the members/prototype of a category *C*, the more likely *X* is to be categorized as *C* (Kruschke, 2008).
c. Priming: the more similar a target *T* is to a prime *P*, the greater the facilitation of processing *T* when it is preceded by *P* (Gomez, Ratcliff, & Perea, 2008).

For the purposes of psychological explanation, it has proved fruitful to treat representational similarity as a continuous variable (unlike purely symbolic notions of similarity). This permits direct prediction of a number of continuous measures important for psychology; such is the case for each of the three citations in (6), as summarized in (7).

(7) Continuous similarity scale →

a. probability of error *E*
b. probability of classification as *C*
c. reaction time differences (primed vs. unprimed)

2.3. Similarity of combinatorial representations

To apply a continuous similarity notion to combinatorially structured representations S and S', we combine (i) the similarity of the fillers in S to those in S' with (ii) the similarity of the roles they fill. In the theory we adopt below, (8) will hold (see (13)).

(8) If $S = \{f_j/r_j\}_j$ and $S' = \{f'_k/r'_k\}_k$ are filler/role decompositions of structures S and S', then
$\text{sim}(S, S') = \Sigma_j \Sigma_k \text{sim}(f_j, f'_k)\,\text{sim}(r_j, r'_k)$

The contribution of *filler* similarity to psychological explanation of the type (6a) is illustrated in (9) (Shattuck-Hufnagel & Klatt, 1979:52).

(9) From

sim([k], [g]) > sim([k], [s]),

predict that the relative error probabilities of misproducing /kol/ 'coal' as [gol] 'goal' or as [sol] 'soul' obey[2]

p(/<u>k</u>ol/ → [<u>g</u>ol]) > p(/<u>k</u>ol/ → [<u>s</u>ol]).

The contribution of *role* similarity to psychological explanation of type (6a) is illustrated in (10) (Vousden, Brown, & Harley, 2000).

(10) From

sim(σ_2-onset, σ_1-<u>onset</u>) > sim(σ_2-onset, σ_1-<u>coda</u>),

predict that the relative error probabilities of producing target /kol rid/ 'coal reed' as [rol kid] "role keyed" or as [kor lid] "core lead" obey

p(/kol <u>r</u>id/→[<u>r</u>ol kid]) > p(/kol <u>r</u>id/→[ko<u>r</u> lid]).

Here, the tendency of such speech errors to preserve syllable position is derived from the general principle that if two roles correspond to the same structural position (e.g., onset) within two tokens of a given type (e.g., σ_1 and σ_2), then these roles are more similar than when they correspond to different positions, all else equal. Thus an erroneous output in which [r] appears in the onset of the incorrect syllable ("<u>r</u>ole") is more similar to the target ("coal <u>r</u>eed") than is the erroneous output in which [r] appears in the coda of the incorrect syllable ("co<u>r</u>e"). (See Section 4.3 below.)

2.4. Continuity + combinatorial structure

Gradient Symbol Processing unifies continuity of representations (and hence continuity of similarity) with combinatorial structure by pursuing a fundamental hypothesis of PDP: that at the microstructural level, mental representations are distributed patterns of activation over n simple numerical processing units—that is, vectors in $\mathbb{R}^n$ (Jordan, 1986; Rumelhart, Hinton, & McClelland, 1986; Smolensky, 2006a:150–159).

In a vector space such as $\mathbb{R}^n$, the combinatory operation is *linear combination*, i.e., weighted summation or *superposition*. In such a *superpositional combinatorial representation*[3]

[2] Here and throughout we use underlining to draw attention to the elements critical in comparisons.

(van Gelder, 1991), a constituent is a vector—e.g., (1, 2, 3)—and a composite structure is a vector—e.g., (31, 22, 13)—that is the sum of multiple constituent vectors—e.g., (31, 22, 13) = (1, 2, 3) + (30, 20, 10). It is in this precise sense that the output activation pattern in Figure 1 has constituent macrostructure than can be formally characterized as the structure $[_{\sigma}$ rat].

In fact, our representational space is a *Hilbert space*, a vector space with a *dot product* (or inner product) that can be used to define similarity in the standard way (11).

(11) $\text{sim}(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y} = \Sigma_k x_k y_k = \|\mathbf{x}\| \|\mathbf{y}\| \cos \measuredangle(\mathbf{x},\mathbf{y})$

(Here $\|\mathbf{x}\|$ is the Euclidean length of $\mathbf{x}$—i.e., $[\Sigma_k (x_k)^2]^{½}$—and $\measuredangle(\mathbf{x},\mathbf{y})$ is the angle formed in $\mathbb{R}^n$ by $\mathbf{x}$ and $\mathbf{y}$.) That distributed representations inherently encode similarity has long been emphasized as a central explanatory feature of PDP (Hinton, McClelland, & Rumelhart, 1986).

2.5. Filler/role binding with the tensor product

In the theory we pursue here, the activation pattern realizing a single constituent—a single filler/role binding—is defined as in (12) (Smolensky, 1990).

(12) [vector realizing filler/role binding] = [vector realizing filler] ⊗ [vector realizing role]

The tensor product ⊗ is a generalization of the matrix outer product; the elements of the vector $\mathbf{x} \otimes \mathbf{y}$ consist of all numbers arising by taking an element of $\mathbf{x}$ and multiplying it by an element of $\mathbf{y}$; e.g., $(1, 2, 3) \otimes (20, 10) = (20, 10; 40, 20; 60, 30)$. Given a distributed representation of fillers and a distributed representation of roles, this yields a distributed representation of constituents in which there are systematic relations between, for example, a given filler in one role and the same filler in a different role (Smolensky, 2006a:175 ff.).

Crucially, these *tensor product representations* (TPRs) provide a macrostructural level of *Distributed Symbol Systems* (Smolensky (2006a) gives a tutorial): this level abstracts away from the particular numbers giving the microstructure of the activation patterns. The calculus of TPRs enables representations with recursive structure, e.g., binary trees, and enables precise computation, in a single massively parallel step of a simple linear associator network, of any mapping in an important class of recursive functions (Smolensky, 2006b:324). TPRs provide a general means of realizing symbolic macrostructure in PDP microstructure.

Distributed Symbol Systems enable general explanatory principles of continuous similarity (6) in the context of combinatorial representations: Section 2.5.1 gives an example.

TPRs formalize ideas of 'conjunctive coding' in early PDP models (e.g., McClelland & Kawamoto, 1986), themselves preceded by 'distributed memory models' (Murdock, 1982; Pike, 1984). Tensor products serve as the basis for a number of connectionist architectures

---

[3] Superpositional *representation* (over constituents) is formally related to, but conceptually distinct from, superpositional *memory* (over exemplars) (Rumelhart & Norman, 1983/1988). Note that many PDP networks introduce nonlinearities in activation functions that restrict vectors to a subset of $\mathbb{R}^n$. See Smolensky & Tesar (2006) for discussion of superpositional combinatorial representations in such networks.

making use of 'vector symbolic' representations (Levy & Gayler, 2008). These sacrifice precision and analyzability by compressing full TPRs into smaller vector spaces; they typically rely on random vectors with specified properties, and, even assuming errorless computation, require "clean-up" processes unnecessary for bona fide TPRs (see Section 2.5.2). Nonlinearities are sometimes used to compress the range of activations. Two early proposals were the Holographic Reduced Representations of Plate (1991, 2000, 2003) and the Recursive Autoassociative Memory of Pollack (1990). Subsequent developments use a variety of compression schemes (for reviews: Gayler, 2003; Kanerva, 2009; Smolensky & Tesar, 2006).

2.5.1. Example: Similarity and word-recognition priming

Here we show how Distributed Symbol Systems enable similarity to explain priming effects in visual word recognition (6c). Relative to dissimilar controls, orthographically similar nonword primes (e.g., *honse* as a prime for HORSE) induce faster lexical decision times (Davis & Lupker, 2006; Forster & Davis, 1984). Recent studies have demonstrated *transposition priming*; facilitation is observed when a nonword prime equals the target with two letters transposed (e.g. *hosre* for HORSE; Perea & Lupker, 2003). This has been explained by assuming that mental representations of orthographic form are structured such that strings containing the same letter in distinct serial positions (e.g., *sr* vs. *rs* in *hosre* vs. *horse*) have non-zero similarity (Gomez et al., 2008).

TPRs allow us to utilize these explanations within a continuous combinatorial representational space. We can compute, for example, that if the roles r_1, r_2 are the first and second positions in a letter string, then (13) holds (illustrating (8)).

(13) $$\begin{aligned}\text{sim}(\texttt{AB}, \texttt{XY}) &= \text{sim}(\mathbf{A}\otimes\mathbf{r}_1 + \mathbf{B}\otimes\mathbf{r}_2, \mathbf{X}\otimes\mathbf{r}_1 + \mathbf{Y}\otimes\mathbf{r}_2) = (\mathbf{A}\otimes\mathbf{r}_1 + \mathbf{B}\otimes\mathbf{r}_2)\cdot(\mathbf{X}\otimes\mathbf{r}_1 + \mathbf{Y}\otimes\mathbf{r}_2)\\ &= [(\mathbf{A}\otimes\mathbf{r}_1)\cdot(\mathbf{X}\otimes\mathbf{r}_1) + (\mathbf{B}\otimes\mathbf{r}_2)\cdot(\mathbf{Y}\otimes\mathbf{r}_2)] + [(\mathbf{A}\otimes\mathbf{r}_1)\cdot(\mathbf{Y}\otimes\mathbf{r}_2) + (\mathbf{B}\otimes\mathbf{r}_2)\cdot(\mathbf{X}\otimes\mathbf{r}_1)]\\ &= [(\mathbf{A}\cdot\mathbf{X})(\mathbf{r}_1\cdot\mathbf{r}_1) + (\mathbf{B}\cdot\mathbf{Y})(\mathbf{r}_2\cdot\mathbf{r}_2)] + [(\mathbf{A}\cdot\mathbf{Y})(\mathbf{r}_1\cdot\mathbf{r}_2) + (\mathbf{B}\cdot\mathbf{X})(\mathbf{r}_2\cdot\mathbf{r}_1)]\\ &= [\text{sim}(\mathbf{A},\mathbf{X})\cdot\text{sim}(\mathbf{r}_1, \mathbf{r}_1) + \text{sim}(\mathbf{B},\mathbf{Y})\cdot\text{sim}(\mathbf{r}_2, \mathbf{r}_2)]\\ &\quad + [\text{sim}(\mathbf{A},\mathbf{Y})\cdot\text{sim}(\mathbf{r}_1, \mathbf{r}_2) + \text{sim}(\mathbf{B},\mathbf{X})\cdot\text{sim}(\mathbf{r}_2, \mathbf{r}_1)]\end{aligned}$$

So if, say, sim(A, B) = 0, sim(A, A) = 1 = sim(B, B), then sim(AB, BA) = 2 sim($\mathbf{r}_1$, $\mathbf{r}_2$). Thus the similarity of the string AB and its transposition BA will be non-zero if and only if the encoding of position 1 and position 2 "overlap"—have non-zero similarity (i.e., are not orthogonal). This then is the crucial requirement for an encoding scheme for letter strings to predict transposition priming via (6c) (Fischer-Baum & Smolensky, 2011; see also Hannagan, Dupoux, & Christophe, 2011).

2.5.2. Aside: The size of tensor product representations

Because of the prevalence of misconceptions on the subject, we digress to consider the size of a TPR: it is the product of the sizes of the fillers and roles that are bound together. For buffer sizes for which *human* parallel processing is plausible (Section 1), the size of TPRs is generally not excessive.

Given a 300-dimensional semantic space (allowing, with binary features, $2^{300} > 10^{90}$ concepts), chunks of conceptual structure encoded as binary trees of depth up to 5 (with up to 32 terminal nodes) require $300(2^{5+1}-1) = 18{,}900$ TPR units.[4] Strings of up to 20 words from a lexicon of 10,000 words can be encoded in a TPR of size 200,000 units using an extravagant coding, linearly independent vectors for all fillers and roles (enabling completely unconstrained mappings to another level of representation in a single layer of weights). These structure sizes seem generous relative to human parallel processing capacity.

The size of TPRs is often greatly exaggerated in the literature. For example, the case claimed by Marcus (2001:106) to require $24{,}300{,}000 = (10 \cdot 3)^5$ units actually requires $3{,}640 = 10(3^{5+1}-1)/2$: depth-5 trees with role vectors of size 3, and filler vectors of size 10. The error here is to mistake the correct form encoding, e.g., the string ABC, which is $\mathbf{A}\otimes\mathbf{r}_1 + \mathbf{B}\otimes\mathbf{r}_2 + \mathbf{C}\otimes\mathbf{r}_3$, for the incorrect expression $(\mathbf{A}\otimes\mathbf{r}_1)\otimes(\mathbf{B}\otimes\mathbf{r}_2)\otimes(\mathbf{C}\otimes\mathbf{r}_3)$.

That TPRs are not problematically large for cognitive modeling is attested by the fact that the "compressed" representations of models in the literature (see text prior to Section 2.5.1) tend, in fact, to be significantly *larger* than corresponding TPRs using even linearly independent filler and role vectors. Plate (2000) uses 2,048 units where 180 units suffice for a TPR (12 fillers × 15 roles). Gayler & Levy (2009) use 10,000 units in lieu of a $8^4 = 4{,}096$-unit TPR (8-D fillers in 4-fold products). The three models discussed in Hannagan, Dupoux, & Christophe (2010) each use 1000 units instead of TPRs requiring 256 or 64 units: these models *approximately* encode strings of length up to 8 with an alphabet of 8 symbols; with TPRs, the same 1000 units can *precisely* encode strings over 30 symbols up to length 33.

There may well be computational or empirical reasons that noisy, compressed representations (with their concomitant clean-up processes) enable better cognitive models than do TPRs (requiring no clean-up processes). But to our knowledge such arguments have yet to be provided; size (let alone efficiency) seems unlikely to provide those arguments.

2.6. Generating representations: Continuous activation and blends

In addition to continuous similarity, another continuous facet of mental representations has played an important explanatory role in many cognitive domains, including psycho-linguistics—even in frameworks other than PDP. During computation, a mental representation contains 'partial activation' of alternative structures, activation levels forming a continuum. The degree of activation of structure X at time t, $a_X(t)$, is essentially the amount of evidence accrued by time t that X is appropriate for the representation of the current target of comprehension or production. While very basic, this point, interpreted as in (14), is crucial.

(14) a. The activation of X, a_X, is an *evaluation* of X. The most appropriate structures—those receiving the best evaluations—are *optimal* in the current context. Processes computing activation values are *optimization* processes.

[4] Depth-d binary-branching trees require role vectors with size totaling $\sum_{k=0}^{d} 2^k = 2^{d+1} - 1$; the 2 here is the minimal size of role vectors for binary branching (Smolensky 2006c:304ff). If m- instead of 2-dimensional primitive role vectors are used, the total size is $\sum_{k=0}^{d} m^k = (m^{k+1} - 1)/(m - 1)$.

b. During the intermediate stages of computing activation values via continuous spreading-activation (evidence-gathering) algorithms, a mental representation typically contains multiple partially activated structures—a *blend*.

As a concrete example of (14b), consider the McClelland & Rumelhart (1981; Rumelhart & McClelland, 1982) model of visual letter perception and word recognition. Initially, activation flows from the units denoting features (line segments) in the stimulus to the units denoting letters; in a given position, the unit for the correct letter receives the most activation, but all letters sharing some of the features of the stimulus also receive some activation. Initially, there is a blend in which multiple letters are partially active; the more similar a letter is to the stimulus, the stronger its representation in the blend.

In a vector space, formalizing blends is straightforward. If $\mathbf{v}_\alpha$ is the vector encoding a letter α, then, say, $0.8\,\mathbf{v}_E + 0.6\,\mathbf{v}_F$ is simply a blend of the letters E and F in which the strengths of the letters E, F in the blend are 0.8, 0.6. A *pure* representation, as opposed to a blend, is exemplified by $1.0\,\mathbf{v}_E + 0.0\,\mathbf{v}_F = \mathbf{v}_E$.

Early in the processing of an input, then, mental representations are typically blends. The key question now is, *when a component relaxes into a final output state, are representations blends or pure?* It turns out that the combinatorial structure of representations plays an important role in determining the answer.

2.7. Ambiguity of blends of superpositional combinatorial representations: quantization

Consider a mental state **a**, a balanced blend of two syllables, [slɪt] 'slit' and [ʃrɛd] 'shred'. Assume for simplicity a representation in which the fillers are phonological segments and the roles are *first-segment*, *second-segment*, etc.[5] (the same result holds for the more psycholinguistically accurate structure (5)). Then we have the result in (15).

(15)
$$\begin{aligned} 0.5\,\mathbf{v}_{[slɪt]} + 0.5\,\mathbf{v}_{[ʃrɛd]} &= 0.5\,(\mathbf{s}\otimes\mathbf{r}_1 + \mathbf{l}\otimes\mathbf{r}_2 + \mathbf{ɪ}\otimes\mathbf{r}_3 + \mathbf{t}\otimes\mathbf{r}_4) + 0.5\,(\mathbf{ʃ}\otimes\mathbf{r}_1 + \mathbf{r}\otimes\mathbf{r}_2 + \mathbf{ɛ}\otimes\mathbf{r}_3 + \mathbf{d}\otimes\mathbf{r}_4) \\ &= 0.5\,[(\mathbf{s}+\mathbf{ʃ})\otimes\mathbf{r}_1 + (\mathbf{r}+\mathbf{l})\otimes\mathbf{r}_2 + (\mathbf{ɛ}+\mathbf{ɪ})\otimes\mathbf{r}_3 + (\mathbf{d}+\mathbf{t})\otimes\mathbf{r}_4] \\ &= 0.5\,(\mathbf{ʃ}\otimes\mathbf{r}_1 + \mathbf{l}\otimes\mathbf{r}_2 + \mathbf{ɪ}\otimes\mathbf{r}_3 + \mathbf{t}\otimes\mathbf{r}_4) + 0.5\,(\mathbf{s}\otimes\mathbf{r}_1 + \mathbf{r}\otimes\mathbf{r}_2 + \mathbf{ɛ}\otimes\mathbf{r}_3 + \mathbf{d}\otimes\mathbf{r}_4) \\ &= 0.5\,\mathbf{v}_{[ʃlɪt]} + 0.5\,\mathbf{v}_{[srɛd]} \end{aligned}$$

This blend of [slɪt] and [ʃrɛd] is identical to a balanced blend of [ʃlɪt] ('shlit') and [srɛd] ('sred'): this state is ambiguous.[6] This is *not* true of a symbolic state representing an equal degree of belief that the word is "slit" or "shred": the concatenatory combination operation of symbolic representation does not lead to the ambiguity we have seen arising from superpositional combination. This ambiguity also does *not* arise with completely local connectionist representations, in which the entire string [slɪt] is represented by a single unit,

[5] Using *contextual* roles (Smolensky, 1990; essentially, *n*-grams) rather than *positional* roles alters but does not eliminate blend ambiguity. If strings, e.g., ABC, are represented through bigrams, e.g., {BC, AB}, then $\mathbf{v}_{AB} + \mathbf{v}_{XY}$ is an unambiguous mixture, but an even blend of ABC and XBY equals an even blend of XBC and ABY (see also Prince & Pinker, 1988).

[6] Crucially, (under the standard requirement that role vectors be linearly independent) the superpositions involved in a *pure* state do *not* yield ambiguity; e.g., [slɪt] is not ambiguous with [stɪl], because $\mathbf{v}_{[slɪt]} = \mathbf{s}\otimes\mathbf{r}_1 + \mathbf{l}\otimes\mathbf{r}_2 + \mathbf{ɪ}\otimes\mathbf{r}_3 + \mathbf{t}\otimes\mathbf{r}_4 \neq \mathbf{s}\otimes\mathbf{r}_1 + \mathbf{l}\otimes\mathbf{r}_4 + \mathbf{ɪ}\otimes\mathbf{r}_3 + \mathbf{t}\otimes\mathbf{r}_2 = \mathbf{v}_{[stɪl]}$ (Smolensky, 1990).

completely dissimilar from the representation of [ʃlɪt].[7] The problem is specific to *superpositional combinatorial* representations.

Suppose that the representation in (15) is an intermediate state in the phonological component of speech perception; in this blended state, the phonological component has not yet committed to a single interpretation of the input. In many symbolic systems, this component could output a list with associated degrees of confidence ([slɪt], 0.5; [ʃrɛd], 0.5), and let downstream processes use their knowledge to choose among them. But in our PDP system this is not an option. We assume that it is exactly the phonological component which has the knowledge that 'shlit' and 'sred' are not possible English words; [ʃl] and [sr] are not possible English syllable onsets. For the phonological system to output the blend (15) is for that system to fail to apply its knowledge; downstream components may not have the knowledge needed to reject the possible interpretations 'shlit' and 'sred', so phonology cannot pass this decision on to them. It must choose among the alternatives that it knows to be possible words, committing to either the output 'slit' or the output 'shred.'

As shown below (Section 3.4), in speech production, blends in phonological representations result from compromises between conflicting *phonological* constraints, not from phonetic factors: these blends must be resolved in the phonological component, not in another component such as phonetics.

This point applies also for blends in which multiple outputs are significantly active but one dominates. For example, if during speech production the phonological component of an English speaker outputs a syllable with 0.6[t] + 0.4[d] in coda position, the corresponding phonetic representations will not provide the categorical structure necessary to select between the two active segments. This is because in English, the primary phonetic cue to the [t]–[d] contrast in coda is preceding-vowel duration (Peterson & Lehiste, 1960). Within the relevant range, there are no significant *phonetic* restrictions on vowel duration. This phonetic dimension therefore fails to supply the structure necessary to force the production of a pure [t] or [d] (although other dimensions may differ; see below). The phonetic component cannot resolve all ambiguities arising from blends; the phonological component must strongly commit to a *single* output. This leads to the principle in (16).

(16) In order for a component of the cognitive system to apply its unique knowledge concerning superpositional combinatorial representations, it must eventually resolve blends and relax into a pure state.

Importantly, it remains possible (and often necessary) for a component to choose its pure output based on continuous input from other components that are operating in parallel.

Although most blends constitute failure to apply a components' knowledge, in Gradient Symbol Processing a state of a component that is *very* close to a pure state will have nearly identical effects on other components as would that pure state itself. So in (16) we intend 'a pure state' to mean 'a state very close to a pure state'; we return to such states in Section 4.3.

[7] Similarly, if entire strings are represented by linearly independent distributed representations, no ambiguity arises. Superpositional combinatorial string representations are not linearly independent.

The process of settling on a single, (approximately) pure, symbolically-interpretable state from a continuum of alternatives will be called *quantization*.[8]

This notion of quantization should be distinguished from proposals like Quantal Theory (Stevens, 1972) and related approaches in phonetics (e.g., Dispersion Theory: Liljencrants & Lindblom, 1972; for recent reviews: Recasens & Espinosa, 2009; Stevens & Keyser, 2010). These phonetic theories essentially aim to explain why languages exploit only a small subset of the continuum of articulatory/acoustic states; they apply specifically to the inherently continuous representations of phonetics. But in Gradient Symbol Processing, the continuum of alternatives is defined by blends of *symbolic* states—in *any* symbolic domain, phonology happening to be one. Furthermore, these phonetic theories address the *structure* of phonetic knowledge. In contrast, quantization in Gradient Symbol Processing concerns the successful *application* of knowledge within any component mental process, given the constraints imposed by superpositional combinatorial representations (16).

2.8. The Optimization-Quantization Principle

Combining the conclusions of Sections 2.6 and 2.7 gives (17).

(17) In combinatorial domains, a mental process consists of
   a. evaluating a continuum of alternative possible output representations, and
   b. quantizing to produce a pure symbolic output—ideally, the best-evaluated or optimal one.

Quantization is challenging given distributed representations (see Hinton, McClelland, & Rumelhart, 1986, for detailed discussion of distributed representations). In activation-based models, selecting a single item within a set often means selecting a single unit, readily done through mutual inhibition among competing output units. As in the McClelland and Rumelhart (1981) model considered above, a single abstract neuron encoding a symbolic output can perform the job of attaining activation 1 for itself and activation 0 for its competitors (approximately). With distributed representations, quantization is much more difficult.

3. Processing: Subsymbolic Optimization-Quantization

In this section we develop a theory of the technical apparatus instantiating Gradient Symbol Processing. This system must perform the optimization and quantization processes needed to output a pure, ideally correct, combinatorial representation. The goal is a theory of processing that allows grammatical knowledge to be effectively exploited, within an activation-based computational architecture of the sort that has become the workhorse of psycholinguistic research. We begin with quantization.

[8] We thank Ian Coffman for suggesting this apt term.

3.1. Quantization: Projecting to the grid

The quantization process can be viewed as projecting the representational state to the *grid* formed by pure representations. Figure 2 illustrates this using a 2-dimensional surface, focusing on the opposition between two possible fillers (d/t) for two roles in a syllable (onset/coda).[9] Each dot corresponds to a pure syllable such as [dɛt] 'debt'. Between and around the four dots are states that are blends; one such blend is shown in the figure, but there is a continuum of blends filling out an entire 2-d plane. Since the representations are distributed, *each point of the grid corresponds to a distributed pattern*, a vector of n activation values.

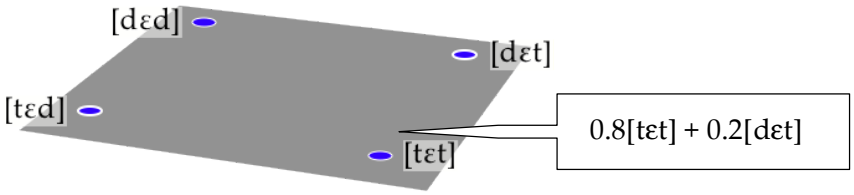


Figure 2. The four dots constitute a slice of the grid of pure states for CVC syllables.

We propose a spreading activation algorithm—a continuous quantization dynamics we call $\mathcal{D}_Q$—that creates an *attractor* at all and only the points of the grid, based on the competitive Lotka-Volterra equations (Baird & Eeckmann, 1993:Sec. 2.6). $\mathcal{D}_Q$ is a distributed non-linear winner-take-all dynamics, achieving a competitive effect like that of lateral inhibition but having attractors that are *distributed* activation patterns. It is implemented by recurrent, second-order connections among the phonological-representation units; these are the connections indicated by the dashed arrow in Figure 1. The dynamics is isotropic: all attractors are equally strong—it is the optimization dynamics discussed below, not the quantization dynamics, that pushes the system toward the optimal attractor basin. This analytical factorization contrasts with work such as Plaut et al. (1996) in which learned connections generate a rich mixture of combinatorial attractor basins and exceptional basins, reflecting the quasi-regular structure of the English spelling-to-pronunciation mapping. More recently, Dilkina, McClelland, and Plaut (2008) extend this type of architecture to language production, where learned connections implement a mixture of attractor basins over phonological output units. The top-down approach we develop here provides a more completely formally characterized system for constructing combinatorial attractors.

The quantization dynamics can be derived as follows. We wish to construct a network that has an attractor at every possible tensor product representation in the cognitive domain being modeled. To construct this, we begin with a localist encoding of each of the filler/role bindings that make up the TPRs. This localist network consists of an $n_f \times n_r$ matrix of units, where n_f and n_r are the number of filler and roles in the domain. If the activation of the unit

[9] In terms of the filler vectors $\{\mathbf{f}_d, \mathbf{f}_t\}$ and role vectors $\{\mathbf{r}_{onset}, \mathbf{r}_{coda}\}$, the center of these four points is $\mathbf{c} = \frac{1}{2}(\mathbf{f}_d+\mathbf{f}_t)\otimes(\mathbf{r}_{onset}+\mathbf{r}_{coda})$, one axis in the grid is $\mathbf{v}_o = \frac{1}{2}(\mathbf{f}_d-\mathbf{f}_t)\otimes\mathbf{r}_{onset}$, the other is $\mathbf{v}_c = \frac{1}{2}(\mathbf{f}_d-\mathbf{f}_t)\otimes\mathbf{r}_{coda}$; the four grid points are then $\mathbf{c}+\mathbf{v}_o+\mathbf{v}_c$ (dɛd), $\mathbf{c}+\mathbf{v}_o-\mathbf{v}_c$ (dɛt), $\mathbf{c}-\mathbf{v}_o+\mathbf{v}_c$ (tɛd), $\mathbf{c}-\mathbf{v}_o-\mathbf{v}_c$ (tɛt). Thus the surface is the plane through $\mathbf{c}$ spanned by $\mathbf{v}_0$ and $\mathbf{v}_c$. This plane is a slice through a higher-dimensional state space through which the system moves when settling to a grid point.

in the *f*th row and *r*th column C_{fr} equals 1, then constituent *f*/*r* is part of the symbol structure currently represented by the network. A set of bindings, a pure grid state, is represented by an activation pattern in which each column (role) contains exactly one unit (filler) with activation 1, all others having activation 0. (To represent structures containing unfilled roles, we incorporate a 'null' filler, and we assume pure states have at most one filler per role.)

To create attractors, weights are added to this localist network to give each column of units competitive Lotka-Volterra dynamics (Baird & Eeckmann, 1993[10]).

(19) Competitive Lotka-Volterra dynamics over constituents within each role *r*

$$dC_{fr}/dt = C_{fr} - C_{fr} \sum_{f'} a_{rff'} C_{f'r}$$

where $a_{rff'}$ is the strength[11] of the competition between C_{fr} and $C_{f'r}$.

These dynamics turn each column of units (role) into a localist winner-take-all pool. Within each column there are therefore exactly n_f attractor states, each of which has one unit with $C_{fr} = 1$ and all the others $C_{f'r} = 0$. Note that the connection weights $a_{rff'}$ multiply the product of two activations—these are *sigma-pi* units (Rumelhart, Hinton & McClelland, 1986)—and that these connections are confined to units in the same column *r*. Critically, because the connections do not cross column boundaries, the Lotka-Volterra dynamics in the columns operate independently—yielding a combinatorial set of attractors corresponding to all possible symbol structures in the domain.

Next we make these attractors distributed TPR states. Let $\boldsymbol{C}$ be the $n_f \times n_r$ matrix of activations in the localist network. The corresponding distributed TPR of that structure, $\boldsymbol{S}$, can be obtained as in (20).

(20) Distributed TPR from localist network

$$\boldsymbol{S} = \boldsymbol{FCR}^T$$

where $\boldsymbol{F}$ is an $n_f \times n_f$ matrix whose columns are the possible filler vectors in the domain, and $\boldsymbol{R}$ is an $n_r \times n_r$ matrix whose columns are the possible role vectors.[12]

This transformation from localist to distributed representations can be viewed as a linear substitution of variables (Smolensky, 2006c): the elements of $\boldsymbol{S}$, the variables $S_{\alpha\beta}$, are each defined by a linear combination of the variables C_{fr}. Alternatively, the local representation can be viewed as a description of the distributed network in a 'conceptual' coordinate system (Smolensky, 1986b). The $S_{\alpha\beta}$ are activations of units in a distributed network whose dynamics are derived from the localist network above; differentiating (20) gives (21).

---

[10] For simplicity we assume that the inherent growth rate of each constituent's activation (Baird & Eeckmann's (1993: 13) u_s term) is 1.

[11] In our simulations, for all *r*, $a_{rff'} = 1$ if $f' = f$, otherwise $a_{rff'} = 2$. These values satisfy the conditions for stability of this dynamical system (Baird & Eeckmann, 1993). Then $a_{rff'} = 2 - \delta_{ff'}$, with $\delta_{ff'} = [1 \text{ if } f = f' \text{ else } 0]$. For use in note 13, we define $A_{frf'r'} = \delta_{rr'} a_{rff'}$; then (19) becomes $dC_{fr}/dt = C_{fr} - C_{fr} \Sigma_{f'r'} A_{frf'r'} C_{f'r'}$.

[12] The TPR unit in row (filler index) α, column (role index) β has activation:

$$S_{\alpha\beta} = [\Sigma_{fr} C_{fr} \mathbf{f}_f \otimes \mathbf{r}_r]_{\alpha\beta} = \Sigma_{fr} C_{fr} [\mathbf{f}_f]_\alpha [\mathbf{r}_r]_\beta = \Sigma_{fr} C_{fr} [\boldsymbol{F}]_{\alpha f} [\boldsymbol{R}]_{\beta r} = \Sigma_{fr} [\boldsymbol{F}]_{\alpha f} C_{fr} [\boldsymbol{R}^T]_{r\beta} = [\boldsymbol{FCR}^T]_{\alpha\beta}.$$

(21) Dynamics of units in the distributed network using variables in the localist network

$$dS_{\alpha\beta}/dt = \sum_{fr} F_{\alpha f}\left(\, dC_{fr}/dt \,\right) R^{T}_{r\beta}$$

In order to re-express these dynamics solely in terms of the $S_{\alpha\beta}$ variables, C_{fr} must be written as a function of $S_{\alpha\beta}$. This is possible (22) if the filler and role matrices $\boldsymbol{F}$ and $\boldsymbol{R}$ are invertible. This follows from an additional, independent assumption of lossless tensor product representations: filler and role vectors are linearly independent (Smolensky, 2006c).

(22) Localist variables expressed as combinations of distributed variables

$$C_{fr} = \sum_{\alpha\beta} [F^{-1}]_{f\alpha} S_{\alpha\beta} [R^{-1}]^{T}_{\beta r} \qquad \text{i.e., } \boldsymbol{C} = \boldsymbol{F}^{-1}\boldsymbol{S}[\boldsymbol{R}^{-1}]^{T}$$

This can be substituted into equation (19); substituting the result into (21) yields the dynamics of the distributed network solely in terms of $S_{\alpha\beta}$. Crucially, the second-order dynamics that hold for the localist network remain second-order in the distributed network after this substitution. In transforming from $\boldsymbol{C}$ to $\boldsymbol{S}$, the variables C_{fr} are multiplied by the appropriate subset of elements of $\boldsymbol{F}$ and $\boldsymbol{R}$ but not by other variables $C_{f'r'}$; $dS_{\alpha\beta}/dt$ therefore remains second-order.

We can now define a new weight matrix for the distributed network that exactly implements the attractor dynamics of the localist network—except that the attractors in the distributed network are now the distributed TPRs. To derive the equation for these weights, it is necessary to re-express the above equations in vectorized form, so that the $\boldsymbol{C}$ and $\boldsymbol{S}$ matrices become vectors $\boldsymbol{c}$ and $\boldsymbol{s}$; these are related by the $n_f n_r \times n_f n_r$ matrix $\boldsymbol{M}$, the Kronecker product of **R** and **F**: see (23a). Carrying this through (using the parameters for our simulations: see note 11) gives the dynamics of the distributed network: (23b).

(23) State vector and dynamics of the distributed network[13]

a. $\boldsymbol{s} = \boldsymbol{M}\boldsymbol{c}$, $\boldsymbol{c} = \boldsymbol{M}^{-1}\boldsymbol{s}$; i.e., $s_\mu = \Sigma_u M_{\mu u} c_u$, $c_u = \Sigma_\mu M^{-1}_{u\mu} s_\mu$

b. $$ds_\mu/dt = s_\mu - \sum_{\mu'\mu''} W_{\mu\mu'\mu''} s_{\mu'} s_{\mu''} \qquad W_{\mu\mu'\mu''} = \sum_{u=(f,r)} \sum_{u'=(f',r')} M_{\mu u} M^{-1}_{u\mu'} M^{-1}_{u'\mu''} \delta_{rr'} \left(2 - \delta_{ff'} \right)$$

In the three-dimensional weight matrix $\boldsymbol{W}$, the element $W_{\mu\mu'\mu''}$ is the weight of the product of $s_{\mu'}$ and $s_{\mu''}$ in the input to the sigma-pi unit s_μ.

Note that this architecture is inherently scalable. These dynamics can be implemented for any tensor product representational scheme. Furthermore, the complexity of the quantization dynamics does not increase with representational size. Given the modular structure of the Lotka-Volterra dynamics, the complexity of the quantization dynamics is determined by the competition within each role—not the number of roles. Thus $(n_F)^{n_R}$

---

[13] Let the index pair $(\alpha,\beta) = \mu$, and let $u = (f,r)$ and $u' = (f',r')$; then $M_{\mu u} = F_{\alpha f} R_{\beta r}$ and $M^{-1}_{u\mu} = F^{-1}_{f\alpha} R^{-1}_{r\beta}$. With $c_u = c_{(f,r)} = C_{fr}$ and $s_\mu = s_{(\alpha,\beta)} = S_{\alpha\beta}$, (22) becomes $c_u = \Sigma_\mu M^{-1}_{u\mu} s_\mu$; i.e., $\boldsymbol{c} = \boldsymbol{M}^{-1}\boldsymbol{s}$ and inversely $\boldsymbol{s} = \boldsymbol{M}\boldsymbol{c}$: $s_\mu = \Sigma_u M_{\mu u} c_u$. Differentiating: $ds_\mu/dt = \Sigma_u M_{\mu u} dc_u/dt = \Sigma_u M_{\mu u}(c_u - c_u \Sigma_{u'} A_{uu'} c_{u'})$, using note 11, with $A_{uu'} = \delta_{rr'}(2 - \delta_{ff'})$. This becomes, from (23a), $ds_\mu/dt = s_\mu - \Sigma_{uu'\mu'\mu''} M_{\mu u}([M^{-1}_{u\mu'} s_{\mu'}] A_{uu'} [M^{-1}_{u'\mu''} s_{\mu''}]) = s_\mu - \Sigma_{\mu'\mu''} W_{\mu\mu'\mu''} s_{\mu'} s_{\mu''}$ where the weight matrix is given by $W_{\mu\mu'\mu''} = \Sigma_{uu'} M_{\mu u} M^{-1}_{u\mu'} M^{-1}_{u'\mu''} A_{uu'}$.

attractors are generated by combining n_F attractors for each of n_R roles.

3.2. Optimization I: Grammars as numerical evaluation functions

Putting aside quantization for the moment, we turn to evaluation/optimization. In phonological production, the evaluator of alternative outputs is the *phonological grammar* $\mathcal{G}$. The key to incorporating grammar into a continuous PDP network is to realize $\mathcal{G}$ as a numerical *Harmony function* $H_{\mathcal{G}}$; this is called a *Harmonic Grammar* (Legendre, Miyata, & Smolensky, 1990, 2006; Pater, 2009). The arguments to the function $H_{\mathcal{G}}$ are (i) a lexical form, such as /rad/ (German 'wheel'), and (ii) a candidate pronunciation, e.g., [rat]. The numerical value $H_{\mathcal{G}}$(/rad/, [rat]) is the grammar's evaluation of how good [rat] is as a pronunciation of /rad/. This is computed by grammatical well-formedness constraints such as those in (24).[14]

(24) Harmonic Grammar tableau for German 'wheel'

	weights:	3	2	$H_{\mathcal{G}}$
	/rad/ →	MARK$_{voi}$	FAITH$_{voi}$	
a.	[rad]	*		–3
b.	☞ [rat]		*	–2

In (24) we consider two alternative pronunciations—*candidates*—*a* and *b*; candidate *b* is correct for the German grammar. The stars mark constraint violations. The constraint MARK$_{voi}$ is violated by final voiced stop consonants like the [d] in (24*a*)[15] but satisfied by the final voiceless [t] of (24*b*). The constraint FAITH$_{voi}$ requires that the pronounced form be faithful to the segments' voicing features in the lexical form: this is violated by [rat] because it is not faithful to the voicing in the lexical form's final /d/; but [rad] satisfies FAITH$_{voi}$.

For this lexical form /rad/, the two constraints here *conflict* in the technical sense that no candidate pronunciation satisfies them both; the competition goes to the candidate violating the *weakest* constraint. For a Harmonic Grammar has a *weight* $w_{\mathbb{C}}$ for each constraint $\mathbb{C}$; each violation of $\mathbb{C}$ lowers the Harmony of a candidate by $w_{\mathbb{C}}$. In (24), the weights given in the first row yield the Harmony values in the final column. The highest-Harmony option, the optimal output, is *b*, [rat]: this is the correct pronunciation for German.

In the English grammar, however, the constraint weights are reversed, and final lexical /d/ is pronounced faithfully, as [d]: FAITH$_{voi}$ is now stronger than MARK$_{voi}$. This bit of cross-linguistic variation between English and German consists of two different strategies (determined by relative weights) for resolving the conflict between two constraints.

This grammatical framework is closely related to *Optimality Theory* (Prince & Smolensky, 1991, 1993/2004), in which constraint strength is grammatically encoded as a rank within a hierarchy (see Legendre, Sorace, & Smolensky, 2006 for comparisons). A substantial body of work within these frameworks—the vast majority within Optimality Theory—has shown

[14] Our discussion adopts the standard assumption that German stops like /d,t/ differ in the feature [voice]; use of the feature [spread glottis] instead (Jessen & Ringen, 2002) would change nothing here.
[15] In traditional linguistic terminology, a dispreferred element like [d] is called *marked* (Jakobson, 1962; Trubetzkoy, 1939/1969); here, this means it violates the well-formedness constraint MARK$_{voi}$.

that viewing grammars (phonological, syntactic, semantic, etc.) as Harmony optimizers provides insight for linguistic theory (see the electronic archive http://roa.rutgers.edu/).

3.3. Optimization II: Networks as optimizers

The upshot of the previous subsection is that the output of the phonological encoding process (a pronunciation) should be the representation that maximizes Harmony, given its input (a lexical representation). How can such optimal states be computed?

Among the earliest major results about the global properties of PDP networks is that summarized in (25) (Cohen & Grossberg, 1983; Golden, 1986, 1988; Hinton & Sejnowski, 1983, 1986; Hopfield, 1982,1984; Smolensky, 1983, 1986a; for a tutorial, see Smolensky 2006b).

(25) For many types of neural network $\mathcal{N}$, local rules for spreading activation have an emergent property:
   a. the Harmony $H_{\mathcal{N}}$ of the network as a whole increases over time, where
   b. $H_{\mathcal{N}}(\mathbf{a})$ is the *well-formedness* of the activation pattern **a** spanning the network.[16]

Such networks, then, compute optimal representations: Harmony maxima. Whereas *deterministic* spreading activation algorithms lead to *local* Harmony maxima—states with higher Harmony than any neighboring state—computing *global* Harmony maxima requires *stochastic* spreading activation algorithms, which exploit randomness. And it is the global Harmony maxima we need for grammatical outputs (see Legendre et al., 2006, for discussion). For our stochastic Harmony-maximizing network, we choose a simple *diffusion process* (Movellan, 1998; Movellan & McClelland, 1993): a probabilistic search algorithm that increases Harmony by gradient ascent on average, but with random deviations superimposed; the variance of these deviations is proportional to T (the 'temperature'), a parameter which decreases to 0 during computation. This process, which we call $\mathcal{D}_{H_{\mathcal{N}}}$, is defined in (26), which also states the relevant emergent property of this process.

(26) Let the random process $\mathcal{D}_{H_{\mathcal{N}}}$ be defined by the stochastic differential equation[17]

$$da_\beta = \left(\sum_\gamma W_{\beta\gamma} a_\gamma - a_\beta\right) dt + \sqrt{2T}\, dB_\beta = \left(\partial H_{\mathcal{N}} \big/ \partial a_\beta\right) dt + \sqrt{2T}\, dB_\beta$$

[16] $H_{\mathcal{N}}(\mathbf{a})$ is the extent to which **a** satisfies the micro-constraints encoded in the connections and units; $W_{\beta\gamma} = -5$ encodes the constraint "units β and γ should not be active simultaneously (strength = 5)". Algebraically, $H_{\mathcal{N}}(\mathbf{a}) \equiv H^0_{\mathcal{N}}(\mathbf{a}) + H^1_{\mathcal{N}}(\mathbf{a})$, where $H^0_{\mathcal{N}}(\mathbf{a}) \equiv \Sigma_{\beta\gamma}\, a_\beta W_{\beta\gamma} a_\gamma$ depends on the weights and $H^1_{\mathcal{N}}(\mathbf{a}) \equiv -\Sigma_\beta \int^{a_\beta} f^{-1}(a)\, da$ depends on the activation function f of the units in $\mathcal{N}$. We use linear units, with activation function $f(a) = a$, yielding $H^1_{\mathcal{N}}(\mathbf{a}) = -\frac{1}{2}\|\mathbf{a}\|^2$. We assume the presence of a 'bias unit' with constant activation value $a_0 = 1$; this just simplifies notation: $W_{\beta 0}$ is the bias on unit β. We also assume that the (arbitrary) scale of $\mathbb{W}$ has been chosen such that $H_{\mathcal{N}}(\mathbf{a})$ is bounded above.

[17] The difference equation used in the computer simulations (following the notation in (26)) is

$$\Delta a_\beta(t + \Delta t) = \left(\sum_\gamma W_{\beta\gamma} a(t)_\gamma - a_\beta(t)\right) \Delta t + \sqrt{2T\Delta t}\, N_t(0,1)$$

where each $N_t(0, 1)$ is a pseudo-random draw from a standard normal distribution; the variance of random disturbances is thus $2T\Delta t$.

(where $\mathbf{a} \equiv (a_1, \ldots, a_n)$ is the activation pattern of the network, $\mathbb{W} \equiv \{W_{\beta\gamma}\}$ is the connection-weight matrix, $H_{\mathcal{N}}$ is the Harmony of the network state, and B is a Wiener process, a mathematical model of Brownian motion). Then $\mathcal{D}_{H_{\mathcal{N}}}$ converges to a probability distribution in which the probability of an activation pattern $\mathbf{a}$ is

$$p(\mathbf{a}) \propto e^{H_{\mathcal{N}}(\mathbf{a})/T}$$

As $T \to 0$, the probability that the network is in a globally-maximum-Harmony state approaches 1. (Geman & Geman, 1984)

Note that the stochastic aspect of this dynamics, the 'thermal noise', is responsible for producing *correct* responses—for finding *global* Harmony optima. Because, when given limited processing time, these methods are not guaranteed to succeed, this dynamics will sometimes produce errors: but not because noise or damage—unmotivated for the correct functioning of the system—has been injected for the sole purpose of generating errors.

3.4. Optimization III: Networks as grammars

Section 3.2 showed how to formalize a grammar $\mathcal{G}$ as a numerical function, $H_{\mathcal{G}}$—a measure of *grammatical* Harmony (well-formedness), the discrete global optima of which are the grammatical representations. Section 3.3 showed how stochastic neural networks can compute globally optimal representations, with respect to the *network* Harmony function $H_{\mathcal{N}}$. These results concerning maximization of macrostructural $H_{\mathcal{G}}^{[\text{macro}]}$ and of microstructural $H_{\mathcal{N}}^{[\text{micro}]}$ well-formedness can be unified because of yet another result:

(27) Given a second-order Harmonic Grammar $H_{\mathcal{G}}$, we can design a neural network $\mathcal{N}_{\mathcal{G}}$ such that for any representation s on the grid of pure states:

$$H_{\mathcal{N}_{\mathcal{G}}}^{[\text{micro}]}(\mathtt{s}) = H_{\mathcal{G}}^{[\text{macro}]}(\mathbf{a}_{\mathtt{s}}),$$

where $\mathtt{s}$ is the symbolic macrolevel description of s and $\mathbf{a}_{\mathtt{s}}$ is the activation vector realizing $\mathtt{s}$, the numerical values of which constitute the connectionist microlevel description of s (Smolensky, 2006c:330 ff.) In such a network $\mathcal{N}_{\mathcal{G}}$, the dynamics $\mathcal{D}_{H_{\mathcal{N}}}$ (26) is called $\mathcal{D}_{\mathcal{G}}$: it seeks optima of $H_{\mathcal{G}}$.

A Harmonic Grammar is 'second order' if each individual constraint considers no more than two constituents at a time (as is the case for $\text{FAITH}_{\text{voi}}$ and MARK_{voi} in (24)). (As shown by Hale and Smolensky (2006), although simple, such grammars are sufficiently expressive to specify formal languages at all complexity levels of the Chomsky Hierarchy.) In the theory we propose here, the second-order constraint $\mathbb{C}_{\text{AB}}[h]$ that assesses a Harmony reward of h (negative if a penalty) for each co-occurrence of constituents A and B is encoded as the weight matrix $\frac{1}{2}h[\mathbf{v}_{\text{A}}\mathbf{v}_{\text{B}}^{\text{T}} + \mathbf{v}_{\text{B}}\mathbf{v}_{\text{A}}^{\text{T}}]$; a first-order constraint $\mathbb{C}_{\text{A}}[m]$ assessing Harmony m for each occurrence of A is encoded as the bias vector $m\mathbf{v}_{\text{A}}$.[18] The weight matrix $\mathbb{W}_{\mathcal{G}}$

[18] $\{\mathbf{v}_{\text{X}}\}$ is the basis dual to $\{\mathbf{a}_{\text{X}}\}$, the activation vectors realizing the constituents {X}. That is, $\mathbf{v}_{\text{X}} \cdot \mathbf{a}_{\text{Y}} = \mathbf{v}_{\text{X}}^{\text{T}}\mathbf{a}_{\text{Y}} = \delta_{\text{XY}} = [1 \text{ if X = Y, else } 0]$; the $\{\mathbf{v}_{\text{X}}\}$ are the rows of the inverse of the matrix with columns $\{\mathbf{a}_{\text{x}}\}$.

implementing the second-order Harmonic Grammar $H_{\mathcal{G}}$ is simply the sum (superposition) of all connection weights and biases contributed by all the constraints of $H_{\mathcal{G}}$.

It is crucial that in general, the state in $\mathbb{R}^n$ with highest evaluation—with maximal Harmony—proves to be not a pure structure but a blend of well-formed constituents. To illustrate this important point, consider a dimension of activation space, a, encoding the [±voice] feature of the final consonant in (24) ([d] vs. [t]). $\text{FAITH}_{\text{voi}}$ (strength φ) favors higher values of a (i.e., [+voice], matching the lexical form /rad/) while MARK_{voi} (strength μ) favors lower values of a (i.e., [–voice]). It is not surprising that the optimal compromise turns out to be a value that is primarily low, but pulled up somewhat relative to the situation where the force from $\text{FAITH}_{\text{voi}}$ is downward (/rat/).[19]

In general, then, the optimum is a blend of constituents favored by various constraints. Therefore, in addition to the Harmony-maximizing optimization dynamics $\mathcal{D}_{\mathcal{G}}$ pushing the representation towards grammatical well-formedness, the discretizing, quantization dynamics $\mathcal{D}_{\text{Q}}$ discussed in Section 3.1 is truly needed in order to push the representation towards the grid—to produce a pure response.

To complete the micro-/macro- integration, we now annotate Figure 1, giving Figure 3.

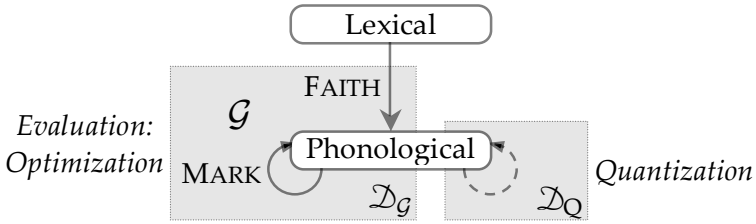


Figure 3. The functional interpretation of the combined dynamics.

The solid arrows encode the grammar $\mathcal{G}$: the connections between the lexical and phonological components encode the FAITHFULNESS constraints (requiring a match, like $\text{FAITH}_{\text{voi}}$ in (24)), while the connections within the phonological component encode the MARKEDNESS constraints (requiring good sound structure, like MARK_{voi} in (24)). Together these solid-arrow connections generate the *optimization dynamics* $\mathcal{D}_{\mathcal{G}}$, which favors representations that are well formed under $\mathcal{G}$. The dashed-arrow connections generate the *quantization dynamics* $\mathcal{D}_{\text{Q}}$ of Section 3.1, which favors grid states—pure symbolic structures.

3.5. The Problem of Mutually-Dependent Choices

How must the optimization dynamics $\mathcal{D}_{\mathcal{G}}$ and quantization dynamics $\mathcal{D}_{\text{Q}}$ be combined? To address this important issue, it proves easier to shift our working example to one in

[19] Specifically, following note 16, the Harmony function is defined as $H(a) = H^0(a) + H^1(a) = \varphi a - \mu a - \frac{1}{2}a^2$ (φ, μ > 0). The scale of {φ, μ} is arbitrary (provided H is bounded above), so we can choose them to satisfy φ+μ = 1, in which case we can rewrite the Harmony as $H(a) = -\frac{1}{2}\varphi[a - 1]^2 - \frac{1}{2}\mu[a - (-1)]^2 + \frac{1}{2}$ which can be interpreted as follows. A penalty of strength φ is paid for the deviation of a from the state that best satisfies $\text{FAITH}_{\text{voi}}$ in isolation (+1), and a penalty of strength μ for deviation of a from the state satisfying MARK_{voi} (–1). The value of a maximizing $H(a)$ is easily seen to be $a_{\text{opt}} = \varphi - \mu = \varphi \cdot (1) + \mu \cdot (-1)$, a weighted average of the states satisfying $\text{FAITH}_{\text{voi}}$ and MARK_{voi} (e.g., for (φ, μ) = (0.1, 0.9), we have $a_{\text{opt}} = 0.1 - 0.9 = -0.8$). We thank Colin Wilson for suggesting this general analysis.

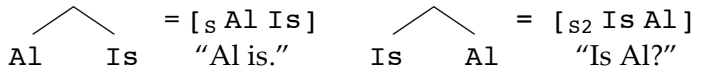
syntax—the simplest, stripped-down case adequate to illustrate the key problem. We also shift to a formal language theory perspective, seeking to design an input-free network that will generate grammatical strings of a language, making arbitrary choices on each run.

The rewrite-rule grammar $\mathcal{G}$ in (28a) generates a formal language $\mathcal{L}$ containing only two sentences, the trees in (28b). This grammar involves only MARKEDNESS constraints and the lower component of Figure 3; there is no input and hence no need for FAITHFULNESS or even an upper component. (The lower component is now computing a syntactic rather a phonological structure, but formally the model is the same.)

(28) The 'Two-Trees' domain

a. A nanogrammar $\mathcal{G}$ b. Its nanolanguage $\mathcal{L}$

Start symbols: {S, S2}
S → Al Is
S2 → Is Al



As discussed above, the optimum for Harmonic Grammars is typically a blend state; here, it is proportional to ([S Al Is] + [S2 Is Al])— an equal blend of the two grammatical trees. To avoid this blend state and yield one of the two optimal pure states, the optimization and quantization dynamics must be coordinated. As the quantization dynamics is forcing a choice of a single filler for each role, the optimization dynamics must ensure that the choices made in different roles are mutually compatible according to the grammar. If the network starts to favor, say, Is for the left-child role, then it must also be driven to favor S2 for the root node role, as well as Al for the right-child role. The choices among fillers for each of the three roles, forced by the quantization dynamics, are *mutually dependent*; the dependencies are determined by the grammar, that is, are encoded in the optimization dynamics. Thus the optimization dynamics $\mathcal{D}_\mathcal{G}$ and the quantization dynamics $\mathcal{D}_\mathrm{Q}$ must operate *simultaneously*. But in order for the final state to be a grid state, the quantization dynamics must be dominant by the end of the relaxation process: the optimization dynamics is opposing the quantization dynamics' push to the grid. To meet these requirements, we have adopted the simplest solution we could devise: the λ-*method*.

(29) The λ-method for combining optimization and quantization

The total dynamics $\mathcal{D}$ is a weighted superposition of the optimization and quantization dynamics, with the weight shifting gradually from optimization to quantization. As computation time t proceeds, the weighting parameter λ_t goes from 1 to 0, and the total dynamics shifts gradually from pure optimization to pure quantization. At time t,

$$\mathcal{D}_t = \lambda_t \mathcal{D}_\mathcal{G} + (1 - \lambda_t) \mathcal{D}_\mathrm{Q}$$

(That is to say, the rate/direction of change of the activation vector over time is a λ_t-weighted sum of the rates/directions of change specified by the two dynamics.)

We can visualize the λ-method as in Figure 4. As $\lambda \to 0$, the Harmony surface in effect grows steeper and steeper peaks at the grid points, as blend states are penalized more and more. ("In effect" because $\mathcal{D}_\mathrm{Q}$ is not actually the gradient of any Harmony function; these

figures are schematic, as are the λ values.) The network state is like an ant climbing uphill as the surface beneath constantly shifts; the goal is to end up at the highest peak.

It is worth pointing out that *the only discrete representation ever evaluated—the only one ever constructed—is the output itself.* The process does *not* evaluate and compare multiple candidate symbolic outputs—notwithstanding naïve interpretation of symbolic tableaux like (24).

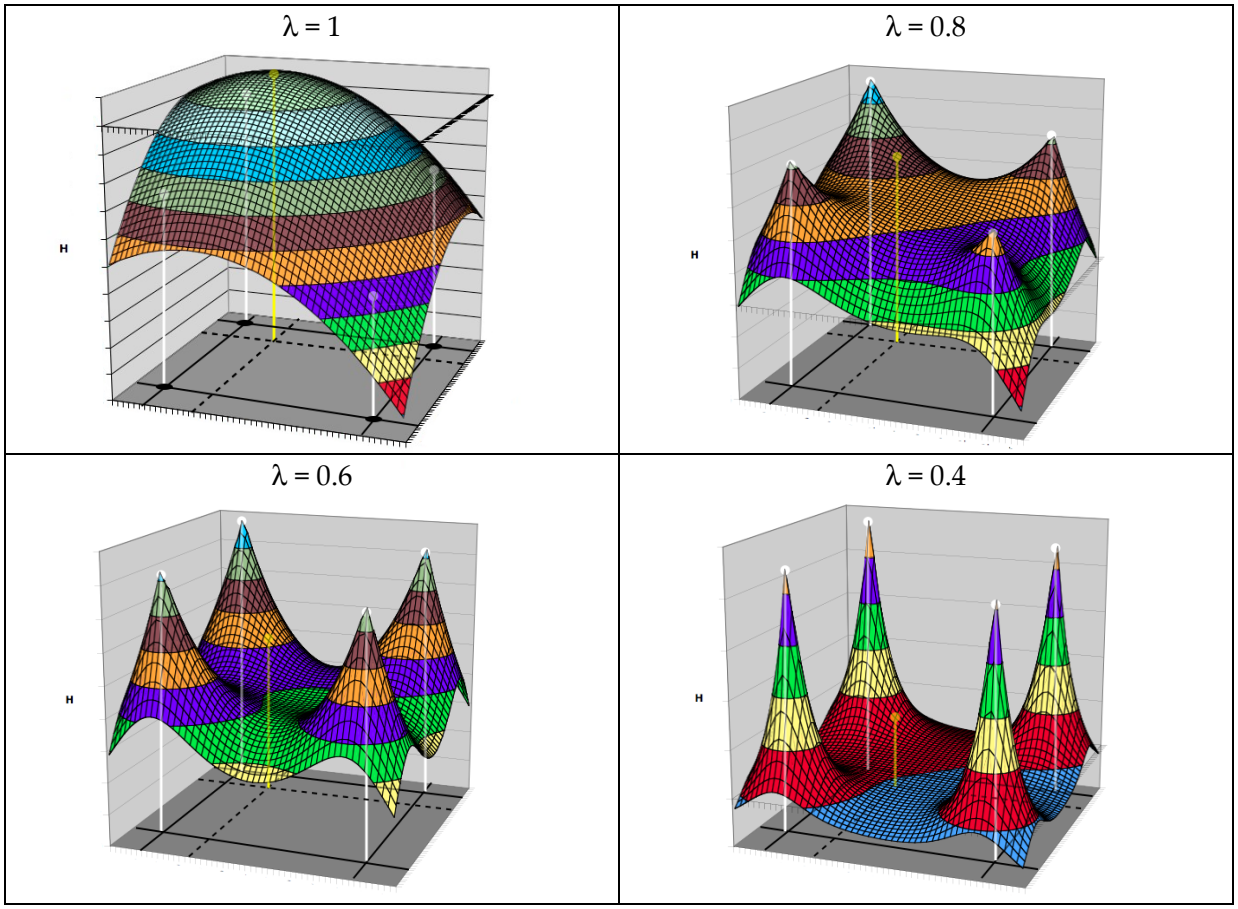


Figure 4. The effective Harmony surface as λ → 0 during computation (schematic). The correct output is the grid point corresponding to the highest peak. The solid lines on the floor intersect at the grid states; the dashed lines, at the blend that optimizes Harmony.

3.6. Computation in Gradient Symbol Processing: Summary

The particular instantiation of Subsymbolic Optimization-Quantization we have proposed here is λ-*Diffusion Theory*, summarized in (30).

(30) λ-Diffusion Theory (an instance of Subsymbolic Optimization-Quantization)
- a. *Optimization:* by diffusion dynamics (26) with dynamic randomness
- b. *Quantization:* by competitive Lotka-Volterra dynamics (23)
- c. *Combination:* by dynamically-weighted superposition, the λ-method (29)

In many connectionist models (including many PDP models), when a single response is required, there is (explicitly or implicitly) a layer of localist units, one per response, with each unit inhibiting all the others, generating a winner-take-all dynamics in which one unit typically ends up with all the activation: this is the response selection dynamics of these models, the counterpart to our quantization. To apply such an approach to the general problem under consideration here, where selection is not among a fixed set of atomic responses, but rather among an open-ended set of combinatorial structures, a single unit would need to be dedicated to each possible combinatorial output (similar to what Pinker & Prince (1988) dub the 'whole-string binding network' of Rumelhart & McClelland (1986a)). The approach we are proposing avoids this, using combinatorially-structured distributed representations as the attractors of the selection dynamics: our approach generates $(n_F)^{n_R}$ attractors by combining n_F attractors constructed for each of n_R roles. This top-down approach complements previous work with learned combinatorial attractors (e.g., Dilkina et al., 2008; Plaut et al., 1996) by providing a formally specified, scalable architecture for selection in these types of representational domains.

The issue of quantization has received considerable attention in architectures using compressed tensor product representations (Section 2.5). To eliminate the noise introduced by compression, 'clean-up' processes use the noisy retrieved vectors to select the best-matching filler representation. As in our framework, Levy & Gayler (2009) and Gayler & Levy (2009) utilize two interleaved dynamical processes: parallel evaluation of possible distributed output representations in a hill-climbing procedure, and a distributed version of winner-take-all. In Levy and Gayler's theory, the relative contribution of these two processes is constant; in our λ-method, the relative weighting of quantization increases as computation proceeds. A second difference is that we utilize stochastic optimization—a necessary feature for finding global Harmony maxima (Section 3.3) and a critical component of our explanation of empirical phenomena in language processing (Section 4.3).

4. Empirical tests

Having motivated and laid out our framework, Gradient Symbol Processing, and a specific instantiation, λ-Diffusion Theory, we now ask whether the theory can address empirical issues in linguistic competence and performance, via specific models constructed within the theory. Our ultimate goal is to develop analytic results proving that the theory (or one of its models) has certain key properties, but at this point we can only report model-simulation results concerning these properties. Open-source simulation code and full documentation can be downloaded from the online supplemental materials at http://faculty.wcas.northwestern.edu/matt-goldrick/gsp.

4.1. *Is the Problem of Mutually-Dependent Choices solved?*

To test whether λ-Diffusion Theory can allow us to handle the critical problem identified in Section 3.5, we modeled the Two-Trees nanogrammar of (28). The network is designed following the general method underlying (27) for constructing a Harmonic Grammar $H_{\mathcal{G}}$ for a rewrite-rule grammar $\mathcal{G}$ (Hale and Smolensky, 2006). A rewrite rule `S` → `X Y` (with `S` a

legal start symbol), contributes to $H_{\mathcal{G}}$ 3 'positive' constraints, each adding their weight (w) when satisfied: '`S` has left-child `X` (2)' and '`S` has right-child `Y` (2)', 'tree-root position has `S`' (1); the rule also contributes 3 negative constraints, each adding their weight ($-w$) when violated: 'no `X` (–1)', 'no `Y` (–1)', and 'no `S` (–3)'. Thus, if an `X` or `Y` appears, it incurs negative Harmony, which can be offset only if it is a legal child of `S`; if an `S` appears, its negative Harmony can be offset by having two legal children and by being at the root of the tree. The net result of all the constraints contributed by all the rules of $\mathcal{G}$ is that every tree generated by $\mathcal{G}$ has Harmony 0, while all other trees have negative Harmony. (A requirement is that $\mathcal{G}$ first be put into 'Harmonic Normal Form', in which every branching symbol can be expanded in only one way; this is why two branching start symbols are necessary in (28a).)

Each constraint in $H_{\mathcal{G}}$ contributes weights to the network $\mathcal{N}_{\mathcal{G}}$ realizing $H_{\mathcal{G}}$ in accord with the text following (27): 'no `S` (–3)' contributes $-3\mathbf{v}_{\mathtt{S}}$ to the total network bias; '`S` has left-child `X` (2)' contributes $\frac{1}{2} \cdot 2 \cdot [\mathbf{v}_{\mathtt{S}/r_x} \mathbf{v}_{\mathtt{X}/r_{0x}}{}^{\mathrm{T}} + \mathbf{v}_{\mathtt{X}/r_{0x}} \mathbf{v}_{\mathtt{S}/r_x}{}^{\mathrm{T}}]$ to the total weight matrix. Here r_x is the role of occupying position x in the tree, and $0x$ is the left child of x; in general, there is a contribution for all tree positions x, but in the nanolanguage (28b), only x = tree-root applies.

A set of distributed, orthogonal, normalized role vectors in $\mathbb{R}^3$ were pseudorandomly generated to implement the three positions in the simple trees (root, left child, right child); a set of vectors in $\mathbb{R}^4$ were similarly generated to implement the possible fillers for each of these positions (`S`, `S2`, `Al`, `Is`). Grid states consisted of all possible role/filler bindings (e.g., not just [$_{\mathtt{S}}$ `Al Is`] and [$_{\mathtt{S2}}$ `Is Al`] but also [$_{\mathtt{S}}$ `Is Is`], [$_{\mathtt{Al}}$ `Al S`], etc.) Following Section 3.5, for the Problem of Mutually-Dependent Choices, we do not consider an input: both grammatical outcomes are equally well formed; the input to the network was therefore set to 0. Temperature and λ were initially set to relatively high values and slowly decayed exponentially. The network was considered to have settled on a solution when the rate of change of activations fell below a certain threshold.

The results of 100 runs of a simulation of the Two-Trees Model suggest that λ-Diffusion Theory solves, with a high degree of accuracy, the particular Problem of Mutually-Dependent Choices posed in Section 3.5 (two runs are shown in Figure 5). In every run, the network converged to one of the equally well-formed grammatical trees (54% [$_{\mathtt{S}}$ `Al Is`] and 46% [$_{\mathtt{S2}}$ `Is Al`]). By superimposing optimization and selection, our framework enables grammatical computation over combinatorial representations in a continuous space.

*4.2. Can discrete and continuous **competence** phenomena be explained?*

Many interesting languages can be specified with a Harmonic Grammar, so λ-Diffusion Theory can be applied to computing grammatical expressions in these languages: in our simulations, when the computational parameters T and λ are lowered sufficiently slowly, with high probability the system settles on an optimal—grammatical—representation. The microgrammar (24) for German final voicing neutralization, for example, has been implemented, achieving 100% accuracy in simulations. Gradient Symbol Processing allows us to go further, and account for the empirically documented *incompleteness* of German final voicing neutralization. As in the empirical data, these simulations show that the final *t*'s output for /rad/ → [rat] 'wheel' and for /rat/ → [rat] 'advice' differ slightly: the former is

slightly closer than the latter to /d/, showing a gradient trace of the underlying lexical form. Space limitations prohibit further discussion, but we point out that the current approach theoretically unifies the gradient incompleteness of *d* ~ *t* neutralization in the German *competence* theory (i.e., in grammatical outputs) and the gradient difference in *performance* when *d* is pronounced as *t* in errors: see the next section.

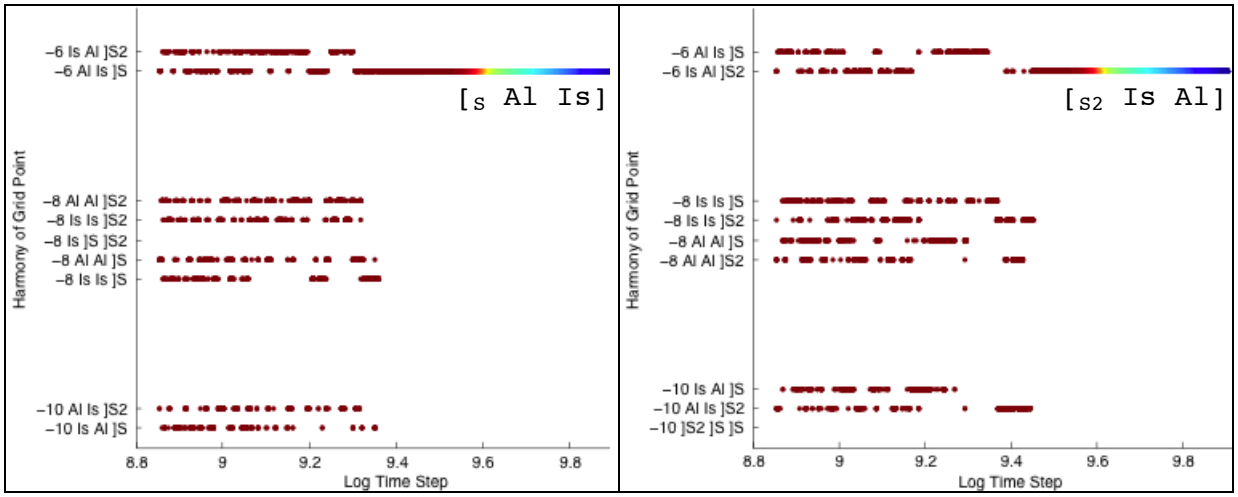


Figure 5. Two runs of a simulation of the Two-Trees Model generating two different trees grammatical in the language (28b). At each time step (horizontal axis), the graph shows (on the vertical axis) the grid state (pure tree) nearest to the current state (i.e., the currently visited $\mathcal{D}_Q$-attractor basin). Red (early) indicates larger and blue (late) smaller distance to the grid. Grid points are arranged vertically by their Harmony; for visibility, in each run, points with the same Harmony are separated arbitrarily.

*4.3. Can discrete and continuous **performance** phenomena be explained?*

In the Gradient Symbol Processing framework, the competence and performance of the cognitive system are deeply connected, allowing a unified account of discrete and continuous patterns in experimental data. In this section, we focus on one specific aspect of grammatical knowledge, FAITHFULNESS constraints; in conjunction with our computational principles, these allow us to formalize similarity-based psychological explanations (Section 2.2) of both discrete and continuous performance phenomena in speech production. As reviewed above (Section 2.3) similar sounds are more likely to interact in speech errors than dissimilar sounds; furthermore, sounds are more likely to interact when they occur in similar syllable positions. In Gradient Symbol Processing, this macrostructural sensitivity to representational similarity emerges from the microstructure of computation.

When λ-Diffusion is forced to produce outputs quickly (as participants must do in a tongue-twister task; Wilshire, 1999), we expect errors to result. As summarized in (31) below, we hypothesize that the distribution of these errors will reflect the stochastic structure of Harmony optimization (26).

(31) Harmonic Error Hypothesis: The probability of a correct or incorrect response x, $p(x)$, is an exponentially increasing function of $H_{\mathcal{G}}(x)$:

$$p(x) \propto \exp(H_{\mathcal{G}}(x)/T), \text{ for some } T$$

Equivalently: $\log p(x) \propto H_{\mathcal{G}}(x) - k$, for some k

Similarity-based explanations of speech error patterns are a specific instantiation of this hypothesis. FAITHFULNESS constraints are violated by phonological representations that fail to preserve the structure of the input along some particular dimension. All else being equal, output structures that better match the structure of the input will therefore have higher Harmony than those that do not. The probability of an error will therefore be a function of its similarity to the target (defined precisely by the structure of FAITHFULNESS constraints).

To test the Harmonic Error Hypothesis, we instantiated λ-Diffusion Theory in the *Tongue-Twister Model* of a tongue-twister task. This model produced sequences of two CVC syllables (e.g., "sag can"). Roles distinguishing syllable number (first, second: r_1, r_2) and syllable position (onset, coda: r_{Onset}, r_{Coda}) were realized by pseudo-random vectors in $\mathbb{R}^2$ constrained to satisfy $\text{sim}(\mathbf{r}_{\sigma_1}, \mathbf{r}_{\sigma_2}) = 0.25$, $\text{sim}(\mathbf{r}_{\text{Onset}}, \mathbf{r}_{\text{Coda}}) = 0.1$. The similarity structure of the role vectors encodes the greater similarity of segments in the same prosodic position across syllables vs. different positions within the same syllable (this parallels the structure of vectors in Vousden et al.'s (2000) model). These role vectors were combined into recursive distributed role vectors (e.g., $\mathbf{r}_{\text{Onset}/\sigma_1} = \mathbf{r}_{\text{Onset}} \otimes \mathbf{r}_{\sigma_1}$; Smolensky, 2006a:182 ff.) yielding vectors in $\mathbb{R}^4$. Distributed filler vectors in $\mathbb{R}^4$ represented four possible consonants. These consisted of a pair of highly similar consonants (e.g., /k/ and /g/; dot product of filler vectors: 0.5) and a pair of less similar consonants (e.g., /s/ and /n/; dot product of vectors: 0.25); across pairs, similarity was low (dot product: 0.1). A set of filler vectors meeting these conditions were generated pseudo-randomly, once for this model. FAITHFULNESS constraints (e.g., 'onset of input syllable 1 = onset of output syllable 1') penalized output representations that were not identical to the input. No MARKEDNESS constraints were present in the modeled grammar.

Production of two different tongue twisters was modeled. The first target syllable in each sequence was the same (e.g., "sag"). The second target syllable was constructed such that similar consonants occurred in the same syllable positions (e.g., "sag knack") or opposite positions (e.g., "sag can"). When λ was allowed to slowly decay from a high starting value (1.0), the system produced both target sequences correctly in each of 100 runs. To simulate the increased speed of the tongue twister task, the initial value of λ was decreased (to 0.015). This causes the network's response time to substantially decrease; at this faster rate, it produced many errors. As shown in Figure 6, the results were consistent with the qualitative patterns observed in experimental speech-error data. Errors on the first syllable (identical across sequences) are more likely to involve more similar segments, and are more likely to involve segments in the same syllable position.

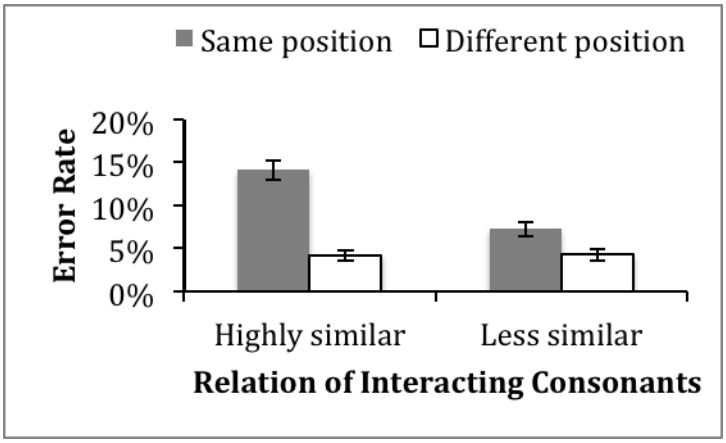


Figure 6. First-syllable error rates in 1,000 runs of a simulation of the Tongue-Twister Model productions of two tongue-twister sequences. Error bars indicate standard error.

The Harmonic Error Hypothesis (31) goes beyond qualitative patterns to make *quantitative* predictions about the relative probability of errors. The results in Figure 7 suggest that these predictions are fairly accurate; the Harmony of an output form is a good predictor of its output probability. This suggests that in λ-Diffusion Theory, the properties of performance errors are closely connected to the computational principle of stochastic Harmony optimization—the key to achieving *competence* within Gradient Symbol Processing. In future work, we plan to explore the degree to which these quantitative predictions account for the empirical distributions of speech errors arising in phonological encoding.

In addition to accounting for discrete phenomena such as likelihood of error outcomes, the concept of similarity has played a role in understanding the continuous properties of speech errors. Recent work has shown that the phonetic properties of speech errors reflect properties of the intended target (see Pouplier & Goldstein, 2010, for a recent review of findings from articulatory and acoustic studies). For example, in an error like 'big' → "pig", the [p] tends to have a shorter voice onset time (VOT) compared to correctly produced instances of 'pig' (Goldrick & Blumstein, 2006). Speech error outcomes thus tend to be slightly similar to the intended target *within continuous phonetic space*.

Our framework allows us to use the same principles that govern discrete error outcomes to account for these continuous error phenomena. For example, if the target grid point is [b], but too-rapid processing causes the network to converge to the region of the grid point for [p], FAITHFULNESS constraints will pull the network's output towards the grid point corresponding to the target [b]. A primary feature of similarity encoded through distributed representations is that similar inputs are mapped to similar outputs (Hinton, McClelland, & Rumelhart, 1986:81 ff.); we therefore assume that, through the phonetic interpretation process (not modeled), such a deviation in the *phonological* representation will manifest itself *phonetically* as a deviation towards the phonetic properties of the faithful output (specifically, a shorter VOT).

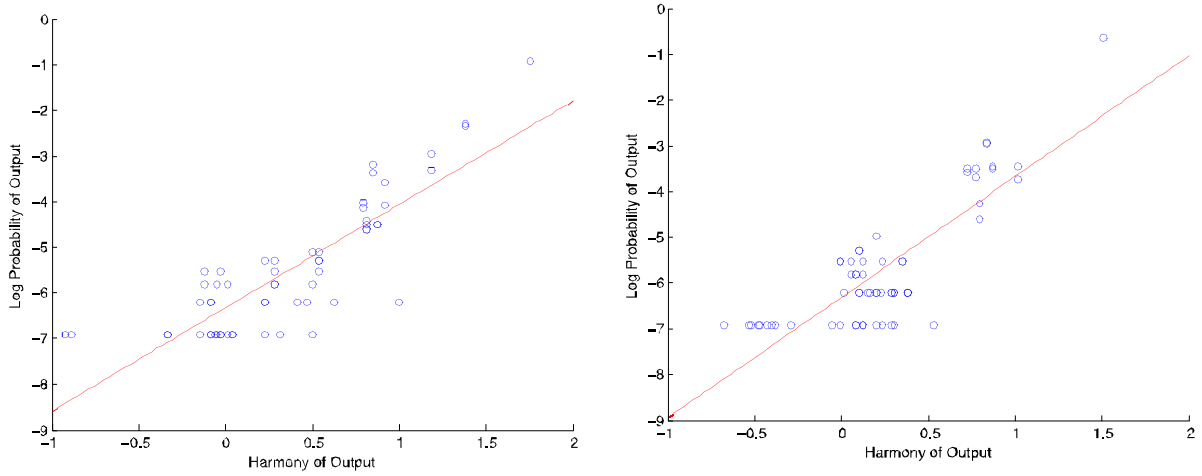


Figure 7. Harmony of grid point (horizontal axis) vs. log probability that grid point was selected as the network output (vertical axis) in 1,000 simulated productions of two tongue-twister sequences (left panel: "sag knack"; right panel: "sag can"). Solid line indicates linear regression fit; compare (31).

To test this hypothesis, we focused on the most frequent errors in the simulation above (involving similar consonants in the same syllable position; e.g., "sag knack" → "sack knack"). Following experimental studies of speech errors, we compared these [k] error outcomes to correctly produced [k]s in the same sequence (e.g., correctly produced coda /k/ in "knack"). The threshold for network settling was such that λ did not decay to 0 (at settling time, λ ≈ .01). As shown in Figure 8, the [k] filler is significantly less active in errors, reflecting the influence of FAITHFULNESS constraints on the continuous aspects of phonological encoding. As discussed above, this variation in the continuous output of the phonological component will alter the input to phonetic processing, producing the observed effects in the articulatory and acoustic properties of errors.

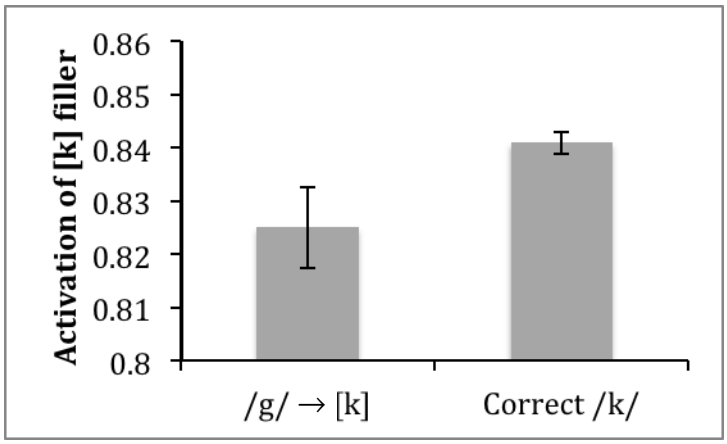


Figure 8. Mean activation of the [k] filler (the dot product of the distributed representation for [k] and the representation of the corresponding constituent of the output) in errors and correct productions. Error bars indicate standard error.

Note that although this discussion has focused on the relationships between similarity and errors induced by FAITHFULNESS, our error hypothesis (31) also makes quantitative predictions about the relationship between error probability and other aspects of the grammar (i.e., MARKEDNESS; see Goldrick & Daland, 2009, for a recent review of relevant speech error data). We plan to examine these predictions more closely in future work.

5. Summary and conclusion

The Gradient Symbol Processing framework developed here aims to account for the emergence (i.e., the formal entailment) of the macrostructural descriptions of grammatical theory from the microstructural algorithms that underlie language processing. Pursuing this PDP research program has, we believe, led to new insights into a central issue in cognition: the relationship between the continuous and the discrete aspects of mental representation and processing. By specifying how discrete structural knowledge emerges from a continuous representational and processing substrate, Gradient Symbol Processing supports a theoretical unification of discrete and continuous empirical phenomena. The same grammatical principles that specify discrete phonological competence also account for both discrete and continuous patterns in speech errors.

Acknowledgements

This research was supported in part by National Science Foundation Grant BCS0846147 (MG, DM), by the Krieger School of Arts and Sciences at Johns Hopkins (DM) and by a Blaise Pascal International Research Chair, funded by the Ile-de-France department and the French national government, and hosted by the Department of Cognitive Studies of the Ecole Normale Supérieure, Paris (PS). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or other sponsors. For helpful feedback, we thank colleagues in the Department of Cognitive Science at Johns Hopkins, particularly Colin Wilson, colleagues in Paris, particularly Emmanuel Dupoux, and audiences at Cambridge University, the 2009 GLOW Conference, the 5th International Workshop on Language Production, Oxford University, Max Planck Institute for Psycholinguistics, Northwestern University, Royal Netherlands Academy of Arts and Sciences, University of Arizona, University of Chicago, the Workshop on Dynamical Systems in Language, and Yale University.

References

Anderson, J. R., & Lebiere, C. J. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Baird, B., & Eeckman, F. (1993). A normal form projection algorithm for associative memory. In M. H. Hassoun (Ed.), *Associative neural memories* (pp. 135-166). New York, NY: Oxford University Press.

Barlow, H. B. (1972). Single units and sensations: A neuron doctrine for perceptual psychology? *Perception, 1*, 371–392.

Bird, H. (1998). Slips of the ear as evidence for the postperceptual priority of grammaticality. *Linguistics, 36*, 469–516.

Bowers, J. S. (2002). Challenging the widespread assumption that connectionism and distributed representations go hand-in-hand. *Cognitive Psychology, 45*, 413–445.

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.

Browman, C. P., & Goldstein, L. (1992). Articulatory phonology: An overview. *Phonetica, 49*, 155–180.

Bybee, J., & McClelland, J. L. (2005). Alternatives to the combinatorial paradigm of linguistic theory based on domain general principles of human cognition. *The Linguistic Review, 22*, 381–410.

Churchland, P. S. & Sejnowski, T. J. (1992). *The computational brain.* Cambridge, MA: MIT Press.

Cohen, M. A., & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, 13*, 815–825.

Davidson, L. (2006). Phonotactics and articulatory coordination interact in phonology: Evidence from non-native production. *Cognitive Science, 30*, 837–862.

Davis, C. J., & Lupker, S. J. (2006). Masked inhibitory priming in English: Evidence for lexical inhibition. *Journal of Experimental Psychology: Human Perception and Performance, 32*, 668–687.

Dell, G. S. (1986). A spreading activation theory of retrieval in sentence production. *Psychological Review, 93*, 283–321.

Dilkina, K., McClelland, J. L., & Plaut, D. C. (2008). A single-system account of semantic and lexical deficits in five semantic dementia patients. *Cognitive Neuropsychology, 25*, 136–164.

Feldman, J. (1989). Neural representation of conceptual knowledge. In L. Nadel, L. A. Cooper, P. Culicover, & R. M. Harnish (Eds.), *Neural Connections, Mental Computation* (pp. 68–103). Cambridge, MA: MIT Press.

Feldman, J. A., & Ballard, D. H. (1982). Connectionist models and their properties. *Cognitive Science, 6*, 205–254.

Fischer-Baum, S. & Smolensky, P. (2011). Positive-overlapping letter position representation in reading: An axiomatic analysis of transposition priming. Paper presented at the 44th annual meeting of the Society for Mathematical Psychology, 18 July, Boston, MA.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*, 3–71.

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.

Gafos, A.I., & Benus, S. (2006). Dynamics of phonological cognition. *Cognitive Science, 30*, 1–39.

Garrett, M. F. (1975). The analysis of sentence production. In G. H Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 9, pp. 133–177). New York: Academic Press.

Gayler, R. W. (2003). Vector Symbolic Architectures answer Jackendoff's challenges for cognitive neuroscience. In Peter Slezak (Ed.), *ICCS/ASCS International Conference on Cognitive Science* (pp. 133–138). Sydney, Australia: University of New South Wales.

Gayler, R. W., & Levy, S.D. (2009). A distributed basis for analogical mapping. In B. Kokinov, K. Holyoak, & D. Gentner (Eds.), *New frontiers in analogy research; Proceedings of the Second International Analogy Conference—Analogy 09* (pp. 165–174. Sofia, Bulgaria: New Bulgarian University Press).

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 6*, 721–741.

Golden, R. M. (1986). The "Brain-State-in-a-Box" neural model is a gradient descent algorithm. *Mathematical Psychology, 30–31*, 73–80.

Golden, R. M. (1988). A unified framework for connectionist systems. *Biological Cybernetics, 59*, 109–120.

Goldrick, M. (2008). Does like attract like? Exploring the relationship between errors and representational structure in connectionist networks. *Cognitive Neuropsychology, 25*, 287–313.

Goldrick, M., & Blumstein, S. E. (2006). Cascading activation from phonological planning to articulatory processes: Evidence from tongue twisters. *Language and Cognitive Processes, 21*, 649–683.

Goldrick, M., Baker, H. R., Murphy, A., & Baese-Berk, M. (2011). Interaction and representational integration: Evidence from speech errors. *Cognition, 121*, 58-72.

Goldrick, M., & Daland, R. (2009). Linking speech errors and phonological grammars: Insights from Harmonic Grammar networks. *Phonology, 26*, 147–185.

Goldrick, M., & Rapp, B. (2007). Lexical and post-lexical phonological representations in spoken production. *Cognition, 102*, 219–260.

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

Hale, J., & Smolensky, P. (2006). Harmonic grammars and harmonic parsers for formal languages. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to Optimality-Theoretic grammar. Vol. 1: Cognitive architecture* (pp. 393–415). Cambridge, MA: MIT Press.

Halford, G. S., Wilson, W. H., & Phillips, S. (1998). Processing capacity defined by relational complexity: Implications for comparative, developmental and cognitive psychology. *Behavioral and Brain Sciences, 21*, 803–865.

Hannagan, T., Dupoux, E., & Christophe, A. (2011). Holographic string encoding. *Cognitive Science, 35*, 79–118.

Hinton, G. E., & Anderson, J. A. (Eds.). (1981). *Parallel models of associative memory*. Mahwah, New Jersey: Erlbaum.

Hinton, G. E., & Sejnowski, T. J. (1983). Optimal perceptual inference. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann Machines. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 282–317). Cambridge, MA: MIT Press.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 77–109). Cambridge, MA: MIT Press.

Hofstadter, D. R. (1979). *Gödel, Escher, Bach: An eternal golden braid.* Basic Books.

Hofstadter, D. R. (1985). Waking up from the Boolean dream, or, subcognition as computation. In D. R. Hofstadter, *Metamagical themas: Questing for the essence of mind and pattern* (pp. 631–665). Basic Books.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA, 79*, 2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences USA, 81*, 3088–3092.

Hummel, J. E., & Holyoak, K. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review, 110(2)*, 220–264.

Jakobson, R. (1962). *Selected Writings I: Phonological Studies*. The Hague: Mouton.

Jordan, M. I. (1986). An introduction to linear algebra in parallel distributed processing. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 365–422). Cambridge, MA: MIT Press.

Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation, 1*, 139–159.

Kruschke, J. K. (2008). Models of categorization. In R. Sun (Ed.), *The Cambridge handbook of computational psychology* (pp. 267-301). New York, Cambridge University Press.

Legendre, G., Miyata Y. & Smolensky, P. (1990a). Harmonic Grammar—A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 388–395). Hillsdale, NJ: Lawrence Erlbaum.

Legendre, G., Miyata, Y., & Smolensky, P. (2006). The interaction of syntax and semantics: A Harmonic Grammar account of split intransitivity. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to Optimality-Theoretic grammar. Vol. 1: Cognitive architecture* (pp. 417–452). Cambridge, MA: MIT Press.

Legendre, G., Sorace, A., & Smolensky, P. (2006). The Optimality Theory-Harmonic Grammar connection. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to Optimality-Theoretic grammar. Vol. 2: Linguistic and philosophical implications* (pp. 339–402). Cambridge, MA: MIT Press.

Levelt, W. J. M., Roelofs, A., & Meyer, A. S. (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences, 22*, 1–75.

Levy, S. D., & Gayler, R. W. (2008). Vector Symbolic Architectures: A new building material for Artificial General Intelligence. *Proceedings of the First Conference on Artificial General Intelligence (AGI-08)*. IOS Press.

Levy, S. D., & Gayler, R. W. (2009b). "Lateral inhibition" in a fully distributed connectionist architecture. In A. Howes, D. Peebles, & R. Cooper (Eds.), *Proceedings of the 9th International Conference on Cognitive Modeling — ICCM 2009* (pp. 318–323). Manchester, UK.

Liljencrants, J., & Lindblom, B. (1972). Numerical simulation of vowel quality systems: The role of perceptual contrast. *Language, 48*, 839–862.

Luce, P. A., & Pisoni, D. B. (1998). Recognizing spoken words: The neighborhood activation model. *Ear & Hearing, 19*, 1–36.

Marcus, G. F. (2001). *The algebraic mind: Integrating connectionism and cognitive science*. Cambridge, MA: MIT Press.

McClelland, J. L. & Bybee, J. (2007). Gradience of Gradience: A reply to Jackendoff. *The Linguistic Review, 24*, 437–455.

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the microstructure of cognition. Vol. 2: Psychological and biological models* (pp. 272–325). Cambridge, MA: MIT Press.

McClelland, J. L. & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review, 88*, 375–407.

McClelland, J. L., Rumelhart, D. E., & the PDP Research Group (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 2: Psychological and biological models.* Cambridge, MA: MIT Press.

McClelland, J. L. & Vander Wyk, B. (2006). *Graded constraints in English word forms*. Unpublished manuscript, Department of Psychology, Carnegie Mellon University.

McMurray, B. Tanenhaus, M. K., & Aslin R. N. (2009). Within-category VOT affects recovery from "lexical" garden-paths: Evidence against phoneme-level inhibition. *Journal of Memory and Language, 60*, 65–91.

Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision* (pp. 211–277). McGraw-Hill.

Movellan, J. R. (1998). A learning theorem for networks at detailed stochastic equilibrium. *Neural Computation, 10*, 1157–1178.

Movellan, J. R., & McClelland, J. L. (1993). Learning continuous probability distributions with symmetric diffusion networks. *Cognitive Science, 17*, 463–496.

Murdock, B. B., Jr. (1982). A theory for storage and retrieval of item and associative information. *Psychological Review, 89*, 316–338.

Newell, A. (1980). Physical symbol systems. *Cognitive Science, 4*: 135–183.

Page, M. (2000). Connectionist modelling in psychology: A localist manifesto. *Behavioral and Brain Sciences, 23*, 443–512.

Partee, B. H., ter Meulen, A., & Wall, R. E. (1990). *Mathematical methods in linguistics.* Boston, MA, Kluwer Academic Publishers.

Pater, J. (2009). Weighted constraints in generative linguistics. *Cognitive Science, 33*, 999–1035.

Perea, M. & Lupker, S. J. (2003). Transposed-letter confusability effects in masked form priming. In S. Kinoshita & S. J. Lupker (Eds.), *Masked priming: State of the art* (pp. 97–120). Hove, UK: Psychology Press,.

Peterson, G. E., & Lehiste, I. (1960). Duration of syllable nuclei in English. *Journal of the Acoustical Society of America, 32*, 693–703.

Pierrehumbert, J. (2006) The next toolkit. *Journal of Phonetics, 34*, 516–530.

Pike, R. (1984). Comparison of convolution and matrix distributed memory systems for associative recall and recognition. *Psychological Review, 91*, 281–294.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28*, 73–193.

Plate, T. A. (1991). Holographic Reduced Representations: Convolution algebra for compositional distributed representations. *Proceedings of the International Joint Conference on Artificial Intelligence*.

Plate, T. A. (2000). Analogy retrieval and processing with distributed vector representations. *Expert Systems: The International Journal of Knowledge Engineering and Neural Networks; Special Issue on Connectionist Symbol Processing, 17(1)*, 29–40.

Plate, T.A. (2003). *Holographic reduced representation: Distributed representation of cognitive structure.* Stanford: CSLI.

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(1)*, 56–115.

Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence, 36*, 77–105.

Port, R. F., & Leary, A. (2005). Against formal phonology. *Language, 85*, 927–964.

Pouget, A., Dayan, P, & Zemel, R. (2000). Information processing with population codes. *Nature Reviews Neuroscience 1*, 125–132.

Pouplier, M., & Goldstein, L. (2010). Intention in articulation: Articulatory timing in alternating consonant sequences and its implications for models of speech production. *Language and Cognitive Processes, 25*, 616–649.

Prince, A., & Pinker, S. (1988). Wickelphone ambiguity. *Cognition, 30*, 188–190.

Prince, A., & Smolensky, P. (1991). Notes on connectionism and Harmony Theory in linguistics (Technical report CU-CS-533-91). Boulder, CO: Computer Science Department, University of Colorado at Boulder.

Prince, A., & Smolensky, P. (1993/2004). *Optimality theory: Constraint interaction in generative grammar.* Technical report TR-2, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ. Technical report CU-CS-696-93, Department of Computer Science, University of Colorado, Boulder. Revised version, 2002: ROA-537-0802, Rutgers Optimality Archive, http://roa.rutgers.edu. Published 2004, Oxford: Blackwell.

Pylyshyn, Z. W. (1984). *Computation and cognition: Toward a foundation for cognitive science.* Cambridge, MA, MIT Press.

Recasens, D. & Espinosa, A. (2009). Dispersion and variability in Catalan five and six peripheral vowel systems. *Speech Communication, 51,* 240–258.

Rumelhart, D. E. (1975). Notes on a schema for stories. In D. G. Bobrow and A. Collins, (Eds.), *Representation and understanding* (pp. 211–236). Academic Press.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for parallel distributed processing. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 110–146). Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review, 89*, 60–94.

Rumelhart, D. E., & McClelland, J. L. (1986a). On learning the past tenses of English verbs. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the microstructure of cognition. Vol. 2: Psychological and biological models* (pp. 216–271). Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1986b). PDP models and general issues in cognitive science. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 110–146). Cambridge, MA: MIT Press.

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

Rumelhart, D. E., & Norman, D. A. (1983/1988). Representation in memory. (1983). Technical Report No. 116, La Jolla, CA: UCSD Center for Human Information Processing. (1988). In R. C. Atkinson, R. J. Herrnstein, G. Lindzey, & R. D. Luce (Eds.), *Steven's handbook of experimental psychology.* New York, NY: Wiley.

Sanger, T. D. (2003). Neural population codes. *Current Opinion in Neurobiology, 13*, 238–249

Shattuck-Hufnagel, S., & Klatt, D. H. (1979). The limited use of distinctive features and markedness in speech production: Evidence from speech error data. *Journal of Verbal Learning and Verbal Behavior, 18*, 41–55.

Smolensky, P. & Legendre, G. (2006). *The harmonic mind: From neural computation to Optimality-Theoretic grammar (Vol. 1: Cognitive architecture; Vol. 2: Linguistic and philosophical implications).* Cambridge, MA: MIT Press.

Smolensky, P. (1983). Schema selection and stochastic inference in modular environments. *Proceedings of the National Conference on Artificial Intelligence* (pp. 378–382).

Smolensky, P. (1986a). Information processing in dynamical systems: Foundations of Harmony Theory. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 194–281). Cambridge, MA: MIT Press.

Smolensky, P. (1986b). Neural and conceptual interpretations of parallel distributed processing models. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 2: Psychological and Biological Models* (pp. 390-431). Cambridge, MA: MIT Press/Bradford Books

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

Smolensky, P. (2006a). Formalizing the principles I: Representation and processing in the mind/brain. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to Optimality-Theoretic grammar. Vol. 1: Cognitive architecture* (pp. 147–205). Cambridge, MA: MIT Press.

Smolensky, P. (2006b). Optimization in neural networks: Harmony maximization. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to Optimality-Theoretic grammar. Vol. 1: Cognitive architecture* (pp. 355–403). Cambridge, MA: MIT Press.

Smolensky, P. (2006c). Tensor product representations: Formal foundations. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to Optimality-Theoretic grammar. Vol. 1: Cognitive architecture* (pp. 271–344). Cambridge, MA: MIT Press.

Smolensky, P., Legendre, G., & Tesar, B. B. (2006). Optimality Theory: The structure, use and acquisition of grammatical knowledge. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to Optimality-Theoretic grammar. Vol. 1: Cognitive architecture* (pp. 453–544). Cambridge, MA: MIT Press.

Smolensky, P., & Tesar, B. B. (2006). Symbolic computation with activation patterns. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to Optimality-Theoretic grammar. Vol. 1: Cognitive architecture* (pp. 235–270). Cambridge, MA: MIT Press.

Stemberger, J. P. (1985). An interactive activation model of language production. In A. W. Ellis (Ed.) *Progress in the psychology of language* (Vol. 1, pp. 143–186). Hillsdale, NJ: Lawrence Erlbaum Associates.

Stevens, K. N. (1972). The quantal nature of speech: Evidence from articulatory-acoustic data. In P. B. Denes, & E. E. David Jr. (Eds.), *Human communication: A unified view* (pp. 51–66). New York: McGraw Hill.

Stevens, K. N., & Keyser, S. J. (2010). Quantal theory, enhancement and overlap. *Journal of Phonetics, 38*, 10-19.

Trubetzkoy, N. (1939/1969). *Principles of phonology* (translation of *Grundzüge der Phonologie*). Berkeley: University of California Press.

van Gelder, T. (1991). What is the "D" in PDP? A survey of the concept of distribution. In W. M. Ramsey, S. P. Stich, & D. E. Rumelhart (Eds.), *Philosophy and connectionist theory* (pp. 33–59). Hillsdale, NJ: Lawrence Erlbaum Associates.

Vousden, J. I., Brown, G. D. A., & Harley, T. A.. (2000). Serial control of phonology in speech production: A hierarchical model. *Cognitive Psychology, 41*, 101–175.

Wermter, S., & Sun, R. (Eds.). (2000). *Hybrid neural systems.* Heidelberg: Springer.

Wilshire, C. E. (1999). The "tongue twister" paradigm as a technique for studying phonological encoding. *Language and Speech, 42*, 57–82.