Dynamic encoding of structural uncertainty in gradient symbols

Pyeong Whan Cho
Department of Cognitive Science
Johns Hopkins University
pcho4@jhu.edu

Matthew Goldrick
Department of Linguistics
Northwestern University
matt-goldrick@northwestern.edu

Richard L. Lewis
Department of Psychology
University of Michigan
rickl@umich.edu

Paul Smolensky
Department of Cognitive Science
Johns Hopkins University
smolensky@jhu.edu

Abstract

An important achievement in modeling online language comprehension is the discovery of the relationship between processing difficulty and surprisal (Hale, 2001; Levy, 2008). However, it is not clear how structural uncertainty can be represented and updated in a continuous-time continuous-state dynamical system model, a reasonable abstraction of neural computation. In this study, we investigate the Gradient Symbolic Computation (GSC) model (Smolensky et al., 2014) and show how it can dynamically encode and update structural uncertainty via the gradient activation of symbolic constituents. We claim that surprisal is closely related to the amount of change in the optimal activation state driven by a new word input. In a simulation study, we demonstrate that the GSC model implementing a simple probabilistic symbolic grammar can simulate the effect of surprisal on processing time. Our model provides a mechanistic account of the effect of surprisal, bridging between probabilistic symbolic models and subsymbolic connectionist models.

1 Introduction

A core computational problem in online language comprehension is to deal with local ambiguity, the one-to-many mapping from a unit symbol w_k (e.g., word) to symbol strings containing w at the k-th position $W_k^* = \cdots w_k \cdots$ and their interpretations S (e.g., sentences and their parses). Rational models of sentence comprehension solve this problem by computing $P(S|W_k)$, a conditional probability of interpretations given a partial string of symbols (henceforth, prefix) $W_k = w_1 \cdots w_k$, and updating it discretely for every new symbol input (Jurafsky, 1996; Hale, 2001; Levy, 2008). We will refer to this class of incremental processing models simply as (structural) probabilistic models.

The probabilistic model has drawn a lot of attention because it predicts processing difficulty in different regions of a sentence based on information-theoretic complexity metrics. The surprisal hypothesis (e.g., Hale, 2001; Levy, 2008) claims that reading time of w_k (as a measure of processing difficulty) is proportional to its surprisal, $-\log P(w_k|W_{k-1})$, or equivalently, the Kullback-Leibler (KL) divergence of $P(S|W_k)$ from $P(S|W_{k-1})$ (Levy, 2008). This hypothesis has been supported in many psycholinguistic experiments (e.g., Boston et al., 2008; Demberg and Keller, 2008; Smith and Levy, 2013).

In this study, our goal is to provide a neurally-plausible, mechanistic account of the relationship between surprisal and processing time. For our purpose, we need a model from which both kinds of information, $P(S|W_k)$ and processing times of w_k, can be collected directly without relying on stipulated linking hypotheses. Since the model is a dynamical system, processing time is directly modeled. To model the probability $P(S|W_k)$ relevant for rational analysis, we treat the model, primarily developed to study interpretation, as a *generator*: it is run to equilibrium with no input, producing a sentence parse as output. This is done repeatedly as the dynamical system is stochastic; this gives a probability distribution over generated parses we call ${}^*P(S)$: this we take to be the knowledge of sentence probabilities that is embodied in the model's dynamics. Then for any W_k, for rational analysis we compute ${}^*P(S|W_k)$ by conditioning ${}^*P(S)$ on W_k, i.e., ${}^*P(S|W_k)$ is the proportion of all generated parses that have prefix equal to W_k. We can then examine the extent to which the model, when serving as an incremental

parser, behaves in accord with rational inference given its knowledge.

The Gradient Symbolic Computation (GSC) framework (Smolensky et al., 2014) serves our goal. The GSC model is a continuous-time, continuous-state stochastic dynamical system model that computes the representation of a discrete structure gradually. This framework grew out of the Integrated Connectionist/Symbolic cognitive architecture (Smolensky and Legendre, 2006). GSC aims to provide an integrated account of the contribution of the continuous dynamics of cognitive processing and the discrete competence that characterizes our knowledge of language.

Cho et al. (2017) applied the framework to incremental processing problems focusing on transient dynamics during incremental processing and argued that the model can achieve two core computational goals in incremental processing: maintaining multiple context-appropriate and globally-coherent interpretations while rejecting interpretations that are context-inappropriate. The GSC parser meets these challenges by moving, during the processing of a word, to an intermediate activation state (a *blend state*) in which multiple symbolic constituents are simultaneously activated to varying partial degrees. From this state, the parser can reach all activation states representing context-appropriate and globally-coherent structures but does not move to activation states representing context-inappropriate structures (either grammatical or ungrammatical). The relation between intermediate activation states and probability distributions over discrete parses was briefly discussed but was not investigated systematically.

In this study, we propose a version of the GSC parser and show how it can be related to other probabilistic sentence-processing models. We argue that the parser's internal state – the activation values of multiple symbolic constituents along with control parameters of the parser – encodes a probability distribution over complete parses (Section 3). After encountering new input, the parser incrementally changes its internal state to encode a new probability distribution. The work the parser needs to do to shift this internal state is closely related to the KL divergence between the probability distributions, providing a link between processing time and surprisal (Section 4). In a simulation study (Section 5), we demonstrate that the GSC parser can approximate rational inference and report the correlation between processing time and surprisal in our model. In Section 6, we summarize our results and discuss some implications of our work.

2 Gradient Symbolic Computation

2.1 Representation

Consider a tree structure `S[1](A,B)`.[1] Let us assign a unique label for every position (called *role*) in the tree structure. For example, we assign labels `r`, `0`, `1` to the mother (root) and the left and right daughter nodes, respectively. Then, we can describe the tree as an unordered set of symbol/position (or *filler/role*) bindings: `S[1](A,B)` $\equiv \{$`B/1,S[1]/r,A/0`$\}$.

Let $\mathbf{f}$ and $\mathbf{r}$ be subsymbolic vector encodings of filler `f` and role `r`. The encoding of binding `f/r` is defined as the tensor product of the two vectors: `f/r` $\equiv \mathbf{f} \otimes \mathbf{r}$ whose (i,j)-th component is the product of the i-th component of $\mathbf{f}$ and the j-th component of $\mathbf{r}$. The encoding of a set of filler/role bindings is defined as the superposition (vector sum) of the encodings of component bindings: $\{\mathtt{f}_1/\mathtt{r}_1, \cdots, \mathtt{f}_k/\mathtt{r}_k\} \equiv \sum_k \mathbf{f_k} \otimes \mathbf{r_k}$. For example, `S[1](A,B)` $\equiv \mathbf{S[1]} \otimes \mathbf{r} + \mathbf{A} \otimes \mathbf{0} + \mathbf{B} \otimes \mathbf{1}$.

In this study, we used local representation (or one-hot encodings) of fillers and roles for facilitating computation. However, many equivalent models with distributed representations can be easily constructed by change of basis (Smolensky, 1986). The result will not change if the distributed representations of bindings remain orthonormal (Smolensky, 1990).

2.2 Constraints

The GSC model uses Harmonic Grammar (HG) (Hale and Smolensky, 2006) to specify grammars via *soft constraints* each of which imposes a reward (a 'positive constraint') or a penalty (a 'negative constraint') on the wellformedness or *Grammatical Harmony* of a gradient symbolic structure. The grammatical structures are those with maximal grammatical Harmony: these structures best satisfy the constraints of the grammar.

As an example, consider a rewrite rule: `S[1]` $\rightarrow$ `A B`. This rule defines a treelet `S[1](A,B)` as

[1] The motivation of using bracketed symbols (e.g., `S[1]`) is presented in Hale and Smolensky (2006). For our purpose, it suffices to say that a bracketed symbol can be considered as a different instance of the same class which has a unique pair of children.

grammatical. HG assigns a positive Harmony reward to any structure for every grammatical pair of bindings — e.g., (S[1]/r, A/0) — it contains. In a network implementation of this HG, these binary rules are implemented as positive weights on between-binding connections, so that whenever one binding is active, it sends positive activation to its grammatical parent and child binding(s).

In addition to these positive contributions from grammatical mother/daughter pairs, the Harmonic Grammar assigns a negative penalty $-b$ to every filler, where b is the number of edges that the filler must have in a grammatical structure. If all those edges are grammatically legal, they will produce positive binary rewards which by design exactly cancel the unary penalties, so that an illformed tree has negative Harmony but a wellformed tree has zero Harmony — the maximum value. The unary HG rules are implemented as negative weights on self-connections of binding units.

The Grammatical Harmony of a set of active filler/role bindings is simply the sum of the Harmony values assigned by all binary and unary HG rules. In the GSC implementation, Grammatical Harmony is defined as in Eq. 1.

$$H_G(\mathbf{a}; \mathbf{W}, \mathbf{ex}) = \frac{1}{2}\mathbf{a}^\top \mathbf{W}\mathbf{a} + \mathbf{ex}^\top \mathbf{a} \qquad (1)$$

where $\mathbf{a}$ is an activation state vector, $\mathbf{W}$ is a weight matrix implementing the grammatical constraints, and $\mathbf{ex}$ is an external input vector, stimulating the target terminal binding corresponding to the present input word. For example, suppose the model is given a second word 'B'. Because it is the second word of a sentence, it must occupy the second terminal role (in our case, 1).[2] Thus, the component of $\mathbf{ex}$ corresponding to binding B/1 has a positive value (a model parameter) and all the other components have a value of 0.

The goal of the GSC parser is to produce an output that represents a discrete tree (at least to a good approximation). This turns out to require further constraints which penalize representations that are not approximately discrete. The Harmony term in Eq. 2, in which f and r are filler and role indices, penalizes representations with multiple symbols filling the same role: it introduces competition among bindings in each role. It is called the *Competition Constraint*. The Harmony term in Eq. 3 penalizes every binding whose activation value is not close to either 0 or 1 — this is the crucial *Discreteness Constraint*, and H_Q is *Discreteness Harmony*. Note that the Competition and Discreteness Constraints in collaboration force the model to choose one filler, with activation 1, in each role. The representations of discrete trees satisfy both these constraints[3] and fall on what we call the *grid* of states: in these states, for each role, the bindings of that role to all symbols all have activation 0 except one, which has activation 1. The representation of the tree S[1] [A B] is on the grid, while an example non-grid state is the one encoding $0.3\,$S[1] [$(0.2\,$A $+\ 0.5\,$C$)$ $(0.4\,$B $-\ 0.1\,$D$)$]

Finally, to ensure the network state does not blow up, we also impose the Baseline Constraint (Eq. 4), which penalizes activation state distant from a baseline activation state $\mathbf{z}$.

$$H_C(\mathbf{a}) = -\sum_r (1 - \sum_f a_{f,r}^2)^2 \qquad (2)$$

$$H_Q(\mathbf{a}) = -\sum_r \sum_f (a_{f,r})^2 (1 - a_{f,r})^2 \qquad (3)$$

$$H_B(\mathbf{a}; \mathbf{z}) = -\frac{1}{2}\|\mathbf{a} - \mathbf{z}\|^2 \qquad (4)$$

The Total Harmony H is the weighted sum of the four Harmony values in Equations 1 – 4:

$$H(\mathbf{a}) = H_G(\mathbf{a}) + \beta H_B(\mathbf{a}) + cH_C(\mathbf{a}) + qH_Q(\mathbf{a})$$

where β, c, and q are the coefficients of non-grammatical constraints. While β and c are fixed, q changes in time, controlled by an external mechanism we do not model here.

The coefficient q governs the strength of the constraint to have discrete activation values (0 or 1) — that is, the strength of the requirement that the model *commit* to symbols being predicted to be present or absent. The Competition Constraint prohibits more than one symbol having activation 1 in any given role, so large q values force the model to *choose* among competitors. Hence we refer to q as the *commitment level*.

2.3 Processing dynamics

The model updates its activation state $\mathbf{a}$ as follows:

$$d\mathbf{a} = \nabla_{\mathbf{a}} H(\mathbf{a}; q(t))dt + \sqrt{2T}dW \qquad (5)$$

where W is the standard multidimensional Wiener process and T is the level of noise. $\nabla_{\mathbf{a}} H(\mathbf{a})$ is the

[2] In this study, we consider minimal tree structures so the three role labels r, 0, 1 will be enough. To deal with deep structures, a more elaborated role labeling system is required.

[3] There is a special Null Symbol "@" which is bound to every role that would otherwise be empty.

gradient of the total harmony evaluated at $\mathbf{a}$. The model optimizes the constraints by stochastically following the gradient, a Brownian motion with drift given by the gradient of Harmony hence, on average, increasing Harmony over time.

$q(t)$ is the commitment level at time t. For convenience, we assume that $q(0) = 0$ and q increases in time because the goal of computation (either in production or in comprehension) is to build a discrete symbolic structure. We will refer to how q changes in time as the *commitment policy* and discuss it in more detail in Section 3.

3 GSC parser as a probabilistic model

3.1 GSC parser

The GSC parser is an application of the GSC framework to incremental parsing. It processes a sentence word-by-word incrementally and passes through intermediate activation states (or blend states) to reach a grid point, the encoding of the parse of the sentence.

Let $\mathbf{ex}_k$, q_k, and $\mathbf{a}_k$ be the external input vector corresponding to w_k, the commitment level and the activation state vector after processing the k-th word. $\mathbf{a}_k$ is a local optimum if $T = 0$. For $T > 0$, we take $\mathbf{a}_k$ to be an approximation of the local optimum. Let $\mathbf{ex}_0(= \mathbf{0})$, $q_0(= 0)$, and $\mathbf{a}_0$ be the initial values of the variables before processing the first word of a sentence. As the parser processes a length-N sentence, its activation state changes from $\mathbf{a}_0$ through $\mathbf{a}_k$ to $\mathbf{a}_N$. Taking q_N to be large, $\mathbf{a}_N$ is close to a grid point and is classified into the nearby grid point by choosing the filler most strongly activated in each role (the *snap-to-the-grid* method). Word processing time for w_k is the time the parser takes to move from $\mathbf{a}_{k-1}$ to $\mathbf{a}_k$.

More specifically, the parser processes each word w_k in three phases. Let $\mathbf{a}_k^j$ be the activation state after phase j given word w_k; $\mathbf{a}_k = \mathbf{a}_k^3$.

- Phase 1a: Update $\mathbf{ex}$ from $\mathbf{ex}_{k-1}$ to $\mathbf{ex}_k$.
- Phase 1b: Update $\mathbf{a}$ from $\mathbf{a}_{k-1}(= \mathbf{a}_{k-1}^3)$ to $\mathbf{a}_k^1$, using $H(\mathbf{a}, q_{k-1})$, allowing settling to convergence.
- Phase 2: Update $\mathbf{a}$ from $\mathbf{a}_k^1$ to $\mathbf{a}_k^2$ by using $H(\mathbf{a}, q_{k-1}) \rightarrow H(\mathbf{a}, q_k)$, i.e., increasing from q_{k-1} to q_k at a constant rate $dq/dt = 1$.
- Phase 3: Update $\mathbf{a}$ from $\mathbf{a}_k^2$ to $\mathbf{a}_k^3(= \mathbf{a}_k)$, using $H(\mathbf{a}, q_k)$, allowing settling to convergence.[4]

The processing time of w_k is defined as the sum of the settling times in phase 1 and 3 and the duration of phase 2.

The parser, in phase 1, integrates a new word input with its internal language model (or structural prediction) and, in phase 2, updates the internal language model via the control of commitment level to make a new structural prediction. In the proposed model, the effect of instantaneous surprisal of w_k (phase 1) is conceptually distinguished from the effect of model update (phase 2) (c.f., O'Reilly et al., 2013).[5]

The role of phase 2 is to reduce the number of grid points reachable from the present activation state.[6] As q increases, the system passes through a series of *bifurcations*, the qualitative changes in the organization of the representation space. When q passes some critical values q_c, more local optima emerge. Each local optimum forms a local hump (*basin of attraction*) on the Harmony surface. Those local optima are separated by Harmony valleys that block transitions from one hump to another: the state seeks higher Harmony. Metaphorically, the paths to some futures (corresponding to different parses) are separated from the present state by these valleys. That is, some structural hypotheses are rejected (Cho and Smolensky, 2016).

Given a length-N sentence, we define a commitment policy π_N as a sequence of q values $(q_0, \cdots, q_k, \cdots, q_N)$ where q_k is the commitment level *after* processing the k-th word in a sentence.

[4] During phase 1 and phase 3, the model monitors convergence as follows. Let $H_{max}(t)$ be the maximum total harmony in a phase up through time t. If H_{max} has not been updated for a certain amount of time ($= 0.5$ in our simulation study; Section 5), the phase ends and the following phase begins. During phase 2, q increases at a constant rate $dq/dt = 1$ so the duration of phase 2 is simply $q_k - q_{k-1}$.

[5] Alternatively, we can consider a GSC parser with a discrete commitment policy. Given a new word input w_k, the model updates both q and $\mathbf{ex}$ discretely from q_{k-1} and $\mathbf{ex}_{k-1}$ to q_k and $\mathbf{ex}_k$. Note that the surprisal of w_k is computed given the updated internal model in this alternative model. Although this alternative parsed every sentence of a minimal grammar G (see Section 5) equally well, we prefer the proposed model to the alternative for the following reason. While $\mathbf{ex}_k$ is given from the environment, an optimal value of q_k given $\mathbf{ex}_k$ must be computed by the parser and the computation must take time.

[6] In terms of the number of reachable grid points, entropy is reduced during phase 2. Because the phase-2 duration is a monotonically increasing function of the amount of increase in q and q is associated with entropy (roughly speaking, the higher q, the smaller entropy), it is likely that a longer phase-2 duration is associated with a larger entropy reduction, which is consistent with the entropy reduction hypothesis (Hale, 2006), although the exact relation between q and entropy needs further investigation.

$q_0 = 0$ and q_N is set to q_{max}; in this setting, the model is guaranteed to reach a grid point after processing the whole sentence (to a close approximation; the higher q_{max}, the better the approximation).

3.2 GSC parser as a probabilistic model

The GSC parser can be related to a structural probabilistic model in the following way. Consider a prefix $W_k = w_1 \cdots w_k$ where w_k is not the final word of a sentence. The GSC parser processes the prefix under a policy $\pi_k = (q_0, \cdots, q_k)$. During processing w_k, the activation state changes from $\mathbf{a}_{k-1}$ to $\mathbf{a}_k$. If we set q_k to q_{max}, the parser will be forced to choose a grid point. If $T > 0$ and the same process is run multiple times, the parser will choose different grid points (encodings of S) in different frequencies. In this way, we can estimate a conditional probability that the parser reaches S if it starts from a tuple $(\mathbf{a}_{k-1}, q_{k-1})$ under $\mathbf{ex}_k$. Because $\mathbf{a}_{k-1}$ is reachable after the parser has processed W_{k-1} under the policy π_k, $P(S|\mathbf{a}_{k-1}, q_{k-1}, \mathbf{ex}_k) = P(S|W_k, \pi_k)$. In this way, we can map a tuple of the activation state and the control state $(\mathbf{a}, q)$ to a probability distribution over S under the constraint $\mathbf{ex}$. An important special case of this, with $k = 0$, allows us to estimate the unconditional distribution $P(S)$ by increasing q from 0 to q_{max} with $\mathbf{ex}_0 = \mathbf{0}$: this amounts to using the model as a *generator* as previewed in Section 1. This estimated distribution is ${}^*P(S)$.

3.3 Rational inference

Rational inference with w_k is defined as the update from ${}^*P(S|W_{k-1})$ to ${}^*P(S|W_k)$ given ${}^*P(S)$ where * indicates conditional probabilities computed by marginalizing ${}^*P(S)$ over cases where W_k were generated for the first k terminal roles.

The surprisal of w_k, $-\ln P(w_k|W_{k-1})$, equals the KL divergence between ${}^*P(S|W_{k-1})$ $(= P_{k-1})$ and ${}^*P(S|W_k)$ $(= P_k)$ (Levy, 2008), which is the expected value of $(\ln P_k - \ln P_{k-1})$.

3.4 Optimal commitment policy

We define a commitment policy π to be optimal if, for every W_k, it minimizes the KL divergence $D_k = D({}^*P(S|W_k) \| P(S|W_k, \pi_k))$. If the D_k are small, the parser approximates rational inference.

4 Surprisal as Harmony difference

The GSC parser processes a sentence word-by-word and processes every word in three phases. In this section, we argue that surprisal can be computed from the intermediate activation states directly and the value will be approximately proportional to the settling time in phase 1.

As the parser processes the k-th word in phase 1, the activation state changes from $\mathbf{a}^3_{k-1}$ to $\mathbf{a}^1_k$ under the influence of $\mathbf{ex}_k$. During this phase, q is fixed at q_{k-1}. When q and $\mathbf{ex}$ are fixed (all the other parameters are constant), the equilibrium probability density follows the Boltzmann distribution (Eq. 6) and the logarithm of the probability ratio of $P(\mathbf{a}^1_k)$ to $P(\mathbf{a}^3_{k-1})$ can be computed as in Eq. 7.

$$P(\mathbf{a}) = \frac{e^{H(\mathbf{a})/T}}{\int e^{H(\mathbf{a}')/T} d\mathbf{a}'} \tag{6}$$

$$\ln P(\mathbf{a}^1_k) - \ln P(\mathbf{a}^3_{k-1}) = \frac{1}{T}(H(\mathbf{a}^1_k) - H(\mathbf{a}^3_{k-1})) \tag{7}$$

where H is parameterized such that $q = q_{k-1}$ and $\mathbf{ex} = \mathbf{ex}_k$. Note that the LHS term of Eq. 7 corresponds to the KL divergence $D(P_k \| P_{k-1}) = E(\ln P_k - \ln P_{k-1})$ where $E(\cdot)$ is the expected value. Thus the surprisal at w_k is $E(\Delta H)/T$, with ΔH being the Harmony difference between the local optima before and after the input update.[7]

We can estimate the expected settling time t_c from the old to the new optimum by recalling that, on average, $d\mathbf{a}/dt = \nabla_{\mathbf{a}} H$, so:

$$\Delta H = \int_0^{t_c} \frac{dH(\mathbf{a})}{dt} dt = \int_0^{t_c} \nabla_{\mathbf{a}} H(\mathbf{a})^\top \frac{d\mathbf{a}}{dt} dt$$
$$\approx \int_0^{t_c} \|\nabla_{\mathbf{a}} H(\mathbf{a})\|^2 dt = t_c \cdot E(\|\nabla_{\mathbf{a}} H(\mathbf{a})\|^2)$$

where the approximation symbol indicates we ignore the stochastic term in Eq. 5. We approximate the average gradient with the average of the gradients at the initial and the final activation states $\mathbf{a}^3_{k-1}$ and $\mathbf{a}^1_k$. The gradient at $\mathbf{a}^1_k$ is $\mathbf{0}$ because $\mathbf{a}^1_k$ is the new optimum. The gradient at $\mathbf{a}^3_{k-1}$ can be calculated as follows: $\nabla_{\mathbf{a}} H(\mathbf{a}^3_{k-1}; q_{k-1}, \mathbf{ex}_k) = (\mathbf{ex}_k - \mathbf{ex}_{k-1}) + \nabla_{\mathbf{a}} H(\mathbf{a}^3_{k-1}; q_{k-1}, \mathbf{ex}_{k-1})$. Note that the last term is $\mathbf{0}$ because it was the optimum under $\mathbf{ex}_{k-1}$ (i.e., before the input word was updated) so the initial gradient is simply $(\mathbf{ex}_k - \mathbf{ex}_{k-1})$. It follows that the magnitude of the average of the initial and final harmony gradients in

[7] As the parser processes w_k, its state changes from $(\mathbf{a}^3_{k-1}, q_{k-1})$ through $(\mathbf{a}^1_k, q_{k-1})$ to $(\mathbf{a}^3_k, q_k)$, all of which have the same future under the influence of $\mathbf{ex}_k$. Thus, under an optimal commitment policy, $P_k = {}^*P(S|W_k) \approx P(S|\mathbf{a}^3_{k-1}, q_{k-1}, \mathbf{ex}_k) = P(S|\mathbf{a}^1_k, q_{k-1}, \mathbf{ex}_k)$. $P_{k-1} = {}^*P(S|W_{k-1}) \approx P(S|\mathbf{a}^3_{k-2}, q_{k-2}, \mathbf{ex}_{k-1}) = P(S|\mathbf{a}^3_{k-1}, q_{k-1}, \mathbf{ex}_{k-1})$.

phase 1 is constant for every w_k.[8] Thus, ΔH is approximately proportional to the settling time t_c.

In sum, surprisal of w_k, under an optimal commitment policy, is related to $\Delta H_k = H(\mathbf{a}_k^1; q_{k-1}, \mathbf{ex}_k) - H(\mathbf{a}_{k-1}^3; q_{k-1}, \mathbf{ex}_k)$ which in turn is proportional to settling time. In our model, surprisal has a geometrical meaning: it is the amount of hill climbing required to reach a new optimum due to the update of the word input.

5 Case study

We investigated a GSC model implementing a minimal probabilistic context-free grammar G = $\{p_1$ S[1] $\rightarrow$ A B, p_2 S[2] $\rightarrow$ A C, p_3 S[3] $\rightarrow$ D B, p_4 S[4] $\rightarrow$ D C$\}$ where p_k is the probability for the k-th sentence and $\sum_k p_k = 1$. Cho et al. (2017) used this minimal grammar (with p_1=p_2=p_3=p_4=0.25) to investigate whether and how the GSC model can deal with computational challenges arising from local ambiguity. They argued that this language creates the core computational problems of incremental processing in the purest form. For example, after processing 'A' as a first word, an ideal incremental processing system must reject `S[3](D,B)` and `S[4](D,C)`. At the same time, it must consider both `S[1](A,B)` and `S[2](A,C)` as candidate interpretations without choosing one over the other too early. They showed that the GSC model can achieve both computational goals by regulating commitment level q appropriately. When q increased too quickly or too slowly, the model respectively made "garden-path" errors (e.g., `S[2](A,C)` for an input sentence 'AB'; Bever, 1970; Frazier, 1987) or "local-coherence" errors (e.g., `S[3](D,B)` for an input sentence 'AB'; Tabor et al., 2004; Konieczny, 2005).

We investigated the same grammar G but we considered the cases where $p_1 \geq p_2$ because our interest is in the relationship between surprisal and processing times. To introduce a structural preference for `S[1]/(A,B)`, a small value $\Delta h \in \{0, 0.1, 0.2, 0.3\}$ was added to the Grammar Harmony of `S[1]`-bindings (see Table 1 in the Appendix). (The model parameter Δh must be distinguished from ΔH discussed above). p_k was empirically estimated by running the model as a generator (i.e., with no external input) 800 times.

[8] Because w_{k-1} and w_k are presented at two different positions in a sentence, $\mathbf{ex}_{k-1} \neq \mathbf{ex}_k$. In every $\mathbf{ex}_k$ (for $k > 0$), only one component has a non-zero value (+2 in the present study) and all the other components have a value of 0. Thus, $\|\mathbf{ex}_k - \mathbf{ex}_{k-1}\|$ is $2\sqrt{2}$ for every $k > 1$; it is 2 for $k = 1$.

5.1 Model

Figure 1 presents the GSC model implementing the grammar. Note that for a different choice of Δh, the parser implements a different PCFG. In addition to Δh, we manipulated T (see Eq. 5) in two levels (0.01 or 0.1) to see how the effect of Δh depends on T.

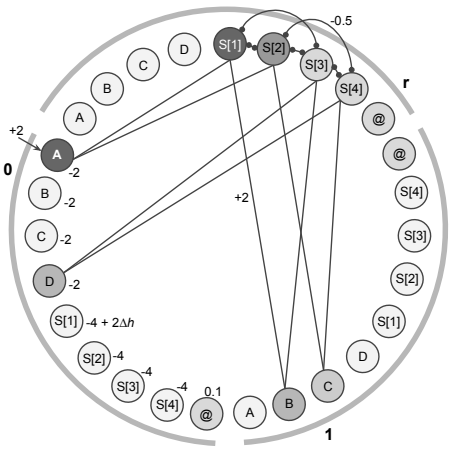

Figure 1: GSC implementation of grammar G via harmonic grammar rules. Only the implementation of grammatical constraints ($\mathbf{W}$ and $\mathbf{ex}$) are presented. The thick gray arcs show the grouping of bindings into different roles. Pairwise connections are bidirectional and implement binary HG rules. Every binding unit has a self-connection (implementing unary HG rules) and their values are presented near the binding units in role 0. The same fillers in other roles have the same negative self-connections as the filler in role 0. The arrow connecting to the binding A/0 indicates external input modeling the word input A as a first word. The colors of the binding units represent partial activation values (white=0, dark=1).

The GSC parser needs a commitment policy. Because every sentence of G is two words long, we considered a commitment policy $\pi = (q_0, q_1, q_2)$ where $q_0 = 0$, $q_2 = q_{max} = 15$, and q_1 was a free parameter.

5.2 Investigation of commitment policy

First, we investigated whether the GSC parser can approximate rational inference as introduced in Section 3. We considered 6 policies in which q_1 was set to one of the values (1, 3, 5, 7, 9, 11).

Every model with a unique combination of Δh, T, and q_1 processed each of four sentences

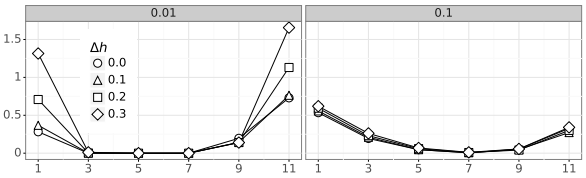

Figure 2: Plot of KL divergence of ${}^*P(S|W_2)$ from $P(S|W_2, \pi_2)$ against q_1 in $\pi_2 = (0, q_1, 15)$. Columns correspond to different T conditions.

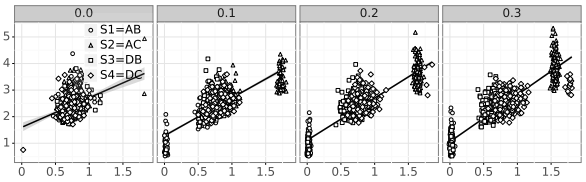

Figure 3: Scatterplot of w_2 processing time (phase-1 duration) against ΔH. Different panels correspond to different Δh conditions. A linear fit line is overlaid in each panel.

(S1=AB, S2=AC, S3=DB, S4=DC) word-by-word 200 times. By applying the algorithm introduced in Section 3, we estimated $P(S)$, $P(S|W_1, \pi_1)$, and $P(S|W_2, \pi_2)$. Because processing time was not of interest here, we excluded phase 1 and phase 3 as the parser processes each word. If dq/dt in phase 2 is small ($dq/dt = 1$ in the simulation), the omission of phase 1 and 3 does not change the result much. An optimal policy was defined as $(0, q_1, 15)$ that minimizes the divergence $D({}^*P(S|W_k) \| P(S|W_k, \pi_k))$ averaged over W_k.

Because π_1 was fixed to $(q_0, q_1) = (0, 15)$, commitment policy does not play any role for the estimation of $P(S|W_1)$. The mean KL divergence from $P(S|W_1)$ to ${}^*P(S|W_1)$ across different first words were small (range=[0.001, 0.021] when $T = 0.01$ and [0.001, 0.020] when $T = 0.1$), suggesting the GSC parser approximates ${}^*P(S|W_1)$.

For w_2, we estimated $P(S|W_2, \pi_2)$ under each of the 6 policies. Figure 2 presents the average KL divergences of ${}^*P(S|W_2)$ from $P(S|W_2, \pi_2)$ as a function of Δh and T. When $T = 0.01$, the divergence was 0 when q_1 is either 5 or 7 in every Δh condition, suggesting the model parsed each of the four sentences accurately. When $T = 0.1$, the divergence was minimal (< 0.017) when $q_1 = 7$ for every Δh condition.[9]

5.3 Investigation of processing times

To investigate the relationship among harmony difference, surprisal (assuming rational inference), and word processing time, we chose the best of the commitment policies $\pi = (0, 5, 15)$ for the condition $T = 0.01$. Each of four GSC parsers, implementing different PCFGs (due to the different Δh values), processed each of four sentences 200 times under the best policy. Because the goal now was to measure word processing time, all three phases were included in this simulation.

[9]See Figures 5 and 6 (Supplementary Material) for estimated probability distributions.

In Section 4, we argued that word processing time, more specifically, phase-1 settling time, must be must be proportional to HarmonyDifference $\Delta H_k = H(\mathbf{a}_k^1) - H(\mathbf{a}_{k-1}^3)$. Figure 3 presents w_2 phase-1 duration against ΔH_2, suggesting a linear trend.[10] In a regression analysis (Model 1A), we modeled w_2 phase-1 duration as a function of SentType (S1=AB, S2=AC, S3=DB, S4=DC to model processing of w_2 in context of w_1), NetID (a unique ID for each GSC parser with a unique Δh value), and HarmonyDifference. SentType and NetID were included to factor out manipulation-irrelevant variance so we do not report the estimates of their coefficients.[11] The coefficient of HarmonyDifference was significant: $b = 1.529$, $SE = 0.024$, $t = 64.919$, $p < .001$, supporting our claim. The adjusted R^2 statistic was 0.787 and $AIC = 3037$. We also tested whether $\ln(\Delta H)$ explains the phase-1 settling time well (Model 1B). The coefficient of log harmony difference was significant as well: $b = 0.445$, $SE = 0.008$, $t = 57.014$, $p < .001$. The adjusted R^2 stastistic was .755 and AIC was 3458, suggesting Model 1A explains processing time data slightly better.

In Section 3, we presented a method to derive a probability distribution over parses S from a tuple of an activation state and a control state q under $\mathbf{ex}$ and a commitment policy π. Based on this, we argued that the harmony difference (scaled by T), can be interpreted as the parser-specific surprisal

[10]The result was the same when total word processing time was used instead of phase-1 duration. This is because phase 2 has the same length for every sentence under the same policy and phase 3 settling time was not systematic in the current T setting. We present phase-1 duration data because it is theoretically related to harmony difference (Section 4).

[11]We did not include the interaction term between SentType and NetID because it covaried with harmony difference and surprisal. Recall that different levels of NetID are associated with different Δh values which in turn were used to create different surprisal values for different sentence types.

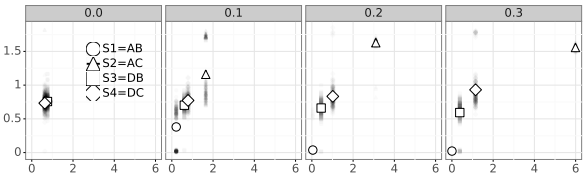

Figure 4: Scatterplot of ΔH_2 against surprisal of w_2 under rational inference. Different panels correspond to different Δh conditions.

$D(P(S|W_k, \pi_k)\|P(S|W_{k-1}, \pi_{k-1}))$, which will be similar to surprisal under rational inference, $D(^*P(S|W_k)\|^*P(S|W_{k-1}))$, under an optimal commitment policy. Thus, we predict harmony difference is a function of surprisal under rational inference under an optimal commitment policy.

Figure 4 presents harmony difference when the input word was updated from w_1 to w_2 against surprisal of w_2 under rational inference, suggesting a non-linear relationship between harmony difference and surprisal. In a regression analysis (Model 2A), we modeled harmony difference as a linear function of surprisal, controlling the effects of SentType and NetID. The coefficient of surprisal was significant: $b = 0.342$, $SE = 0.006$, $t = 53.933$, $p < .001$. The adjusted R^2 statistic was 0.786 and $AIC = -860.4$. In another regression analysis (Model 2B), we modeled harmony difference as a linear function of ln(surprisal). The coefficient of ln(surprisal) was significant: $b = 0.286$, $SE = 0.005$, $t = 60.984$, $p < .001$. The R^2 statistic was 0.811 and $AIC = -1259$, suggesting Model 2B better explains variance in ΔH.

We summarize the result in the following conceptual model: surprisal under rational inference $\rightarrow$ harmony difference (under an optimal commitment policy) $\rightarrow$ word processing time. In other words, harmony difference is the parser's actual surprisal under a commitment policy. The logarithm trend observed between surprisal and harmony difference needs further investigation but we consider two possibilities. First, the average magnitude of the actual gradient is systematically different depending on surprisal so our approximation introduces a bias. Second, although we chose the best commitment policy of 6 candidates, the chosen policy may not be optimal. Note that we used the same commitment policy for all four sentences. However, an optimal q_1 value may differ for the first word A and the first word D.

6 General Discussion

An important research question concerning online sentence processing is to understand the source of processing difficulty. The surprisal hypothesis (Hale, 2001; Levy, 2008) provides a simple, intuitive, and general explanation at a computational level: processing difficulty is proportional to surprisal. The underlying mechanism is still beyond our understanding but researchers have started developing mechanistic accounts of surprisal (e.g., Rasmussen and Schuler, 2017). In this study, we tried to contribute to this line of research by providing a mechanism that relates surprisal to processing time via a stochastic, wellformedness-optimizing mechanism.

Our effort can be summarized as follows. First, the GSC model encodes structural uncertainty in the gradient activation of constituent symbols. An activation state at a given commitment level is analogous to the state of a symbolic parser but contains uncertainty information. It corresponds to a probability distribution over parses in the following sense: if the system starts from the given activation state and the given commitment level and is forced to choose a parse, it will choose different parses (grid points) with different frequencies (see Section 3).

Second, the model updates uncertainty in two ways: in response to the update of external information and via the control of commitment level. On the one hand, external input update makes the previously optimal activation state suboptimal so drives the system to a new optimum. In Section 4, we claimed that the amount of change required to travel from the old to the new optimum, harmony difference, can be interpreted as surprisal. There we showed why the settling time is proportional to the harmony difference. On the other hand, the internal control of commitment level is critical in holding the amount of structural ambiguity at an optimal level; this is implied in Figure 6 (Supplementary Material) but was not the focus of this study. See Cho and Smolensky (2016) for the role of commitment policy.

Third, as we demonstrated in a simulation study (Section 5), the model can approximate rational inference under a good commitment policy and simulate the correlation between surprisal and processing time via harmony difference that is the parser's surprisal under the policy. There we reported the result that surprisal under rational in-

ference explains variance in harmony difference, which in turn explains variance in processing time. In other words, surprisal under rational inference $\rightarrow$ harmony difference (the parser's surprisal) under a commitment policy $\rightarrow$ processing time.

An implication of our work is that surprisal is not a function of linguistic environment only, which we assume the parser learned well. From the GSC point of view, both the linguistic environment and the parser's commitment policy determine surprisal of each word input. For optimal sentence processing, the model needs both types of knowledge.

A limitation of our work is the simplicity of the grammar we investigated. We are actively investigating (with promising preliminary results) the model's ability to process more complex cases. But we point out that finding a good parameter setting and a good commitment policy, which can be challenging, is a separate issue from understanding the relation between surprisal and processing time. The present study focuses on the latter and the claim we made is generalizable.

Probabilistic models (e.g., Hale, 2001; Levy, 2008) provide a computational account of why and what problems must be solved in online language comprehension. Dynamical connectionist models (e.g., Tabor and Hutchins, 2004; Vosse and Kempen, 2009) provide a mechanistic account of why some sentences (e.g., garden-path sentences) take longer to process than others. By proposing how structural uncertainty can be encoded and updated in a symbolically-interpretable dynamical system model, our work bridges between these two general approaches to modeling human sentence processing.

Acknowledgments

We thank Geraldine Legendre, Akira Omaki, Kyle Rawlins, Ben Van Durme, and Colin Wilson for their contributions to this work, and gratefully acknowledge the support of NSF INSPIRE grant BCS-1344269. We thank Paul Tupper for suggesting the form of the H_C and H_Q functions used in this work.

References

Thomas G. Bever. 1970. The cognitive basis for linguistic structures. In John R. Hayes, editor, *Cognition and the Development of Language*, John Wiley, New York, pages 279–362.

Marisa Ferrara Boston, John Hale, Reinhold Kliegl, Umesh Patil, and Shravan Vasishth. 2008. Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam Sentence Corpus. *Journal of Eye Movement Research* 2(1):1–12.

Pyeong Whan Cho, Matthew Goldrick, and Paul Smolensky. 2017. Incremental parsing in a continuous dynamical system: Sentence processing in Gradient Symbolic Computation. *Linguistics Vanguard* 3(1). https://doi.org/10.1515/lingvan-2016-0105.

Pyeong Whan Cho and Paul Smolensky. 2016. Bifurcation analysis of a Gradient Symbolic Computation model of incremental processing. In A. Papafragou, D. Grodner, D. Mirman, and J. C. Trueswell, editors, *Proceedings of the 38th Annual Conference of the Cognitive Science Society*. Cognitive Science Society, Austin, TX.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition* 109(2):193–210. https://doi.org/10.1016/j.cognition.2008.07.008.

Lyn Frazier. 1987. Sentence processing: A tutorial review. In M. Coltheart, editor, *Attention and Performance XII: The Psychology of Reading*, Lawrence Erlbaum Associates, pages 559–586.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL '01, pages 1–8. https://doi.org/10.3115/1073336.1073357.

John Hale. 2006. Uncertainty about the rest of the sentence. *Cognitive Science* 30(4):643–672.

John Hale and Paul Smolensky. 2006. Harmonic Grammars and harmonic parsers for formal languages. In Paul Smolensky and Géraldine Legendre, editors, *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. Volume I: Cognitive Architecture*, The MIT Press, pages 393–416.

Daniel Jurafsky. 1996. A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science* 20(2):137–194. https://doi.org/10.1016/S0364-0213(99)80005-6.

Lars Konieczny. 2005. The psychological reality of local coherences in sentence processing. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*. pages 1178–1183.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition* 106(3):1126–1177. https://doi.org/10.1016/j.cognition.2007.05.006.

Jill X. O'Reilly, Urs Schüffelgen, Steven F. Cuell, Timothy E. J. Behrens, Rogier B. Mars, and Matthew F. S. Rushworth. 2013. Dissociable effects of surprise and model update in parietal and anterior cingulate cortex. *Proceedings of the National Academy of Sciences* 110(38):E3660–E3669. https://doi.org/10.1073/pnas.1305373110.

Nathan E. Rasmussen and William Schuler. 2017. Left-Corner Parsing With Distributed Associative Memory Produces Surprisal and Locality Effects. *Cognitive Science* pages n/a–n/a. https://doi.org/10.1111/cogs.12511.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition* 128(3):302–319. https://doi.org/10.1016/j.cognition.2013.02.013.

Paul Smolensky. 1986. Neural and conceptual interpretation of PDP models. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2: Psychological and Biological Models*, MIT Press, Cambridge, MA, pages 390–431.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence* 46(1):159–216. https://doi.org/10.1016/0004-3702(90)90007-M.

Paul Smolensky, Matthew Goldrick, and Donald Mathis. 2014. Optimization and quantization in gradient symbol systems: A framework for integrating the continuous and the discrete in cognition. *Cognitive Science* 38(6):1102–1138.

Paul Smolensky and Géraldine Legendre, editors. 2006. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar. Volume 1: Cognitive Architecture*. The MIT Press, Cambridge, MA.

Whitney Tabor, Bruno Galantucci, and Daniel Richardson. 2004. Effects of merely local syntactic coherence on sentence processing. *Journal of Memory and Language* 50(4):355–370. https://doi.org/10.1016/j.jml.2004.01.001.

Whitney Tabor and Sean Hutchins. 2004. Evidence for self-organized sentence processing: Digging-in effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 30(2):431–450. https://doi.org/10.1037/0278-7393.30.2.431.

Theo Vosse and Gerard Kempen. 2009. The Unification Space implemented as a localist neural net: Predictions and error-tolerance in a constraint-based parser. *Cognitive Neurodynamics* 3(4):331–346. https://doi.org/10.1007/s11571-009-9094-0.

A Supplementary Material

A.1 Harmonic grammar rules

Table 1 presents a set of harmonic grammar rules for the grammar considered in the simulation study. An empty (or null) filler symbol @ was introduced to treat empty roles; in GSC, an empty role is represented as a binding of the role with the null filler. The null bindings were assigned a small positive harmony value (0.1), which we observed works well for more complex grammars. For the present case, this value can be set to 0 without changing the result much. Note that we assigned a negative harmony value (-0.5) between bindings that share the same daughter at the same role (competition rules). The motivation is to make the candidate parents of a binding compete with each other because every child must have a single mother. In the present study, the competition rules are not necessary because every binding competes with every other binding in each role due to the competition constraint (Eq 2). The competition rules are required when a symbol can be either the right child of a parent symbol or the left child of another parent symbol. In this case, the two parent symbols occur in two different roles. Because the competition constraint is applied to each role independently, we need the competition rules in this situation.

Binary rules	**Unary rules**
H(S[1]/r, A/0) = 2	H(S[1]/·) = $-2 + \Delta h$
H(S[1]/r, B/1) = 2	H(S[2]/·) = -2
H(S[2]/r, A/0) = 2	H(S[3]/·) = -2
H(S[2]/r, C/1) = 2	H(S[4]/·) = -2
H(S[3]/r, D/0) = 2	H(A/·) = -1
H(S[3]/r, B/1) = 2	H(B/·) = -1
H(S[4]/r, D/0) = 2	H(C/·) = -1
H(S[4]/r, C/1) = 2	H(D/·) = -1
H(S[1]/r, S[2]/r) = -0.5	H(@/·) = 0.1
H(S[1]/r, S[3]/r) = -0.5	
H(S[2]/r, S[4]/r) = -0.5	
H(S[3]/r, S[4]/r) = -0.5	

Table 1: Harmonic grammar rules of grammar G. The · symbol represents any arbitrary role symbol ($\in \{\mathtt{r}, \mathtt{0}, \mathtt{1}\}$) and @ indicates an empty symbol (or null filler).

A.2 Estimated probability distributions in the simulation study

Figure 5 presents the estimates of ${}^*P(S)$, ${}^*P(S|W_1)$, and $P(S|W_1, \pi_1)$ from the GSC parser used in the simulation study (Section 5). Overall, $P(S|W_1, \pi_1)$ approximates ${}^*P(S|W_1)$ well.

Figure 6 presents the estimates of $P(S|W_2, \pi_2)$, emphasizing the effect of commitment policy. We do not present ${}^*P(S|W_2)$; for each sentence input $W_2^{(i)}$ and its parse $S^{(i)}$, ${}^*P(S = S^{(i)}|W_2 = W_2^{(i)}) = 1$ and ${}^*P(S = S^{(j)}|W_2 = W_2^{(i)}) = 0$ for

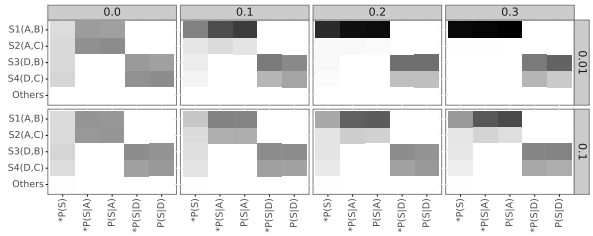

Figure 5: Heatmap of the estimated probability distribtuions given different W_0 and W_1 constraints (x-axis). The y-axis represents the chosen structures (grid points other than the four grammatical structures were collapsed into the "Others" class). Color (white=0, black=1) represents the relative frequencies of the chosen structures. Columns and rows correspond to different Δh and T values, respectively.

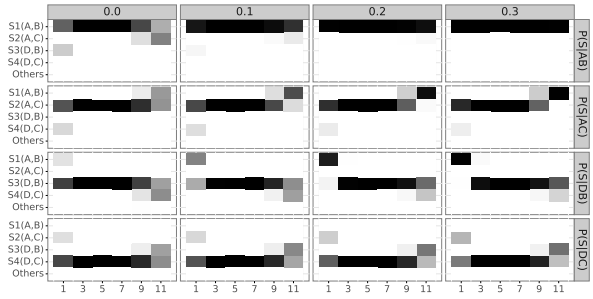

Figure 6: Heatmap of the estimated probability distributions given different W_2 constraints when $T = 0.01$. When $T = 0.1$, the overall pattern was similar but noisier. Columns correspond to different Δh values. In each panel, the x- and y-axes represent q_1 and the chosen structures and color represents proportion (white=0, black=1).

$j \neq i$. Under an optimal commitment policy, the model must be able to choose the target parse after processing a whole sentence. This was observed when q_1 was 5 or 7. When $q_1 = 3$, the overall parsing accuracy was very high but not perfect. Recall that we included the condition $\Delta h = 0$ to replicate Cho and Smolensky (2016) because we used a modified version of the GSC parser. As reported in their paper, the parser made local-coherence errors when q_1 was too low (e.g., `S[3](D,B)` for an input sentence S1=AB) and garden-path errors when q_1 was too high (e.g., `S[2](A,C)` for an input sentence S1=AB).