TOPICS
TOPICS IN COGNITIVE SCIENCE

Topics in Cognitive Science 15 (2023) 255–273
© 2022 The Authors. *Topics in Cognitive Science* published by Wiley Periodicals LLC on behalf of Cognitive Science Society
ISSN: 1756-8765 online
DOI: 10.1111/tops.12629

This article is part of the topic "Best of Papers from the 2022 Cognitive Science Society Conference," Andrea Bender (Topic Editor).

A Computational Complexity Perspective on Segmentation as a Cognitive Subcomputation

Federico Adolfi,[a,b] Todd Wareham,[c] Iris van Rooij[d,e,f]

[a]*Ernst Strüngmann Institute for Neuroscience in Cooperation with Max-Planck Society*
[b]*School of Psychological Science, University of Bristol*
[c]*Department of Computer Science, Memorial University of Newfoundland*
[d]*Donders Institute for Brain, Cognition, and Behaviour, Radboud University*
[e]*School of Artificial Intelligence, Radboud University*
[f]*Department of Linguistics, Cognitive Science, and Semiotics & Interacting Minds Centre, Aarhus University*

Received 12 September 2022; received in revised form 6 October 2022; accepted 7 October 2022

Abstract

Computational feasibility is a widespread concern that guides the framing and modeling of natural and artificial intelligence. The specification of cognitive system capacities is often shaped by unexamined intuitive assumptions about the search space and complexity of a subcomputation. However, a mistaken intuition might make such initial conceptualizations misleading for what empirical questions appear relevant later on. We undertake here computational-level modeling and complexity analyses of *segmentation* — a widely hypothesized subcomputation that plays a requisite role in explanations of capacities across domains, such as speech recognition, music cognition, active sensing, event memory, action parsing, and statistical learning — as a case study to show how crucial it is to formally assess these assumptions. We mathematically prove two sets of results regarding computational hardness and search space size that may run counter to intuition, and position their implications with respect to existing views on the subcapacity.

Keywords: Segmentation; Computational complexity; Tractability; Computational-level analysis; Modeling; Theory

Correspondence should be sent to Federico Adolfi, School of Psychological Science, University of Bristol, Beacon House, Queens Road, Bristol, BS8 1QU, UK. E-mail: fedeadolfi@gmail.com

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

...In [an input stream], however, as presented to a [cognizer],
such explicit segmentation cues are rarely to be found;
little pauses after every single [bit] might make things clearer,
but the input is continuous - a running stream [all around].
This implies that part of [cognizing] involves an operation
whereby input is segmented, to be processed [bit] by [bit],
for we cannot hold in memory each total [segmentation],
as most [such things] we come across [require a novel split].
— Cutler (1994)

1. Introduction

Cognitive scientists routinely invoke subcapacities in decompositional efforts to reverse-engineer fully fledged capacities of minds, brains, and machines (Cummins, 2000; Egan, 2017; Miłkowski, 2013). For instance, speech processing is presumed to decompose into, among other things, segmentation and decoding, and action understanding into parsing, predicting, and goal inference. These *subcomputations* are thought to tackle certain *problems* that the cognitive system faces to behave appropriately in the world.[1]

Problems that originally show up in one domain (e.g., speech processing) are subsequently encountered in other domains (e.g., action understanding), and so the conceptual apparatus naturally carries over. For example, cognitive scientists may come to view the problem of segmenting speech as analogous to the problem of parsing actions. Researchers can then transfer ideas across the domains, adopting and adapting similar subcomputations in their explanations of the different capacities. What is passed along, however, will include latent (and possibly mistaken) notions about the computational properties of these problems as well. For instance, if a cognitive scientist believes that the search space of speech segmentation is large (i.e., combinatorially complex) and that this makes the problem hard, then by analogy, the same could be inferred about the parsing problem in action understanding.

Once such initial framing of a cognitive (sub)capacity is adopted, it completely shapes the kinds of empirical questions that appear relevant and, in so doing, determines the course of research programs across disciplines and cognitive domains. Crucially, the assumptions that gave rise to the initial framing are seldom examined formally, and since they are taken for granted as background commitments, empirical tests are not designed to bear on them. These foundational oversights can sidetrack researchers into directions that will be largely immune to empirical corrective feedback later on.

To illustrate how crucial it is to formally assess the validity of intuitive assumptions about problem properties, and what can go astray if one fails to do so, we undertake here a formal examination of an example subcapacity. Our case study is *Segmentation*.[2] This subcapacity figures ubiquitously in explanations of real-world cognitive capacities, such as speech recognition, music perception, active sensing, event memory, temporal attention, action processing, and statistical learning. We focus on two classes of assumptions about its computational

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

properties: (1) the search space is excessively complex and (2) this makes the segmentation problem intrinsically hard. To formally assess the theoretical viability of these assumed properties, we develop a formalization of the (intuitive) segmentation problem at the computational level (cf. Bechtel & Shagrir, 2015; Marr, 1982), and we submit it to a mathematical analysis to assess the size of its search space, its computational hardness, and its possible sources of complexity using tools from computational complexity theory (Arora & Barak, 2009; Garey & Johnson, 1979; van Rooij, Blokpoel, Kwisthout, & Wareham, 2019).

The remainder of the paper is structured as follows. First, we introduce the problem of segmentation as it is conceptualized in the literature across cognitive domains. Next, we develop a formalization that abstracts from such specifications. We then survey and synthesize the core intuitive assumptions about the computational properties of the problem. Finally, we present proofs that speak to the validity of these assumptions, and discuss the implications for research on segmentation and subcomputations more broadly. As our results may run counter to intuition, we end with a word of caution regarding the general nonintuitiveness of the computational properties of hypothesized cognitive problems.

2. Conceptualization of segmentation

In order to rigorously examine computational assumptions, we need a mathematical formalization of the general problem that can be submitted to further analyses. This computational-level model, in turn, should capture key aspects of the theorized cognitive capacity. To that end, in this section we synthesize conceptualizations of the segmentation problem as it appears in various cognitive domains.

2.1. *Informal definitions: Segmentation as a fundamental subcomputation*

"How the brain processes sequences is a central question in cognitive science and neuroscience" (Jin, Lu, & Ding, 2020). A substantial amount of information available to the cognitive system is "continuous, dynamic and unsegmented" (Zacks et al., 2001). This implies that many cognitive processes must involve an operation whereby input is segmented (cf. Cutler, 1994), and hence, it also arises naturally in machine tasks, such as language translation (e.g., Kolokolova & Nizamee, 2014). The purpose of the segmentation process is, then, "to generate elementary units of the appropriate temporal granularity for subsequent processing" (Giraud & Poeppel, 2012). Succinctly, "[t]he central nervous system appears to 'chunk' time" (Poeppel, 2003), and "[this subcomputation] plays a fundamental role in the way we perceive and remember information in daily life" (Geerligs, van Gerven & Güçlü, 2021).

Several subfields of the cognitive and brain sciences have proposed segmentation as a key subcomputation. *Active listening* (cf. active sensing) casts it as "the selection of internal actions, corresponding to the placement of […] boundaries" (Friston et al., 2021), "to sample the environment" (Poeppel & Assaneo, 2020). *Event cognition* similarly defines it as "the process of identifying event boundaries […], a concomitant component of normal event perception" (Zacks et al., 2001). In *episodic memory*, it is "the process by which people parse the continuous stream of experience into events and sub-events [for] the formation of experience units" (Jeunehomme & D'Argembeau, 2018). Central to *music perception*, it features as determining the "perceptual boundaries of temporal gestalts" (Tenney & Polansky, 1980) and "entails the parsing into chunks" (Farbood, Rowland, Marcus, Ghitza, & Poeppel, 2015; Tillmann, 2012). The *speech recognition* literature describes it as the core process of "segmenting the continuous speech stream into units for further perceptual and linguistic analyses" (Teng, Cogan, & Poeppel, 2019), where it "allows the listener to transform [the] signal into segmented, discrete units, which form the input for subsequent decoding steps" (Poeppel & Assaneo, 2020). In *action processing*, "[o]ne initial step that aids in drawing [inferences regarding other people's goals and intentions based on observable action] is recognizing where action units begin and end within a stream of physically continuous motion" (Meyer, Baldwin, & Sage, 2011). Therefore, "[a] fundamental problem observers must solve [...] is segmentation [...] Identifying distinct acts within the dynamic flow of motion is a basic requirement for engaging in further appropriate processing" (Baldwin, Andersson, Saffran, & Meyer, 2008).

Such ubiquitousness across domains has been suggestive that the capacity "appeals to general principles the brain may use to solve a variety of problems" (Friston et al., 2021; Himberger, Chien, & Honey, 2018). "[M]any sequence-chunking tasks share common computational principles. [For example,] to find and encode the chunk boundaries" (Jin et al., 2020). Segmentation as a subcomputation appears across processing hierarchies as well, even when the world is relatively static: "[it] exists at multiple layers within a given problem" (Wyble & Bowman, 2019). The downstream operations on segments that partially determine optimal segmentation play similar roles but otherwise vary with cognitive domain and modeling framework.

Segmentation, concisely, is then a fundamental subcomputation whose requisite role across cognitive domains and processing hierarchies is to determine, given a sequence representation, the optimal boundary placement with respect to a downstream computation over segments.

3. Formalization of segmentation

A succinct, yet informal, definition of segmentation can be stated by verbally specifying the inputs and outputs of the conjectured subcomputation.

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

SEGMENTATION (INFORMAL)
Input: A sequence and a downstream process that, for any given segment of the sequence, can evaluate its quality relative to domain-specific criteria.
Output: The best[3] segmentation of the sequence with respect to criteria relevant for the downstream process.

With this sketch in mind (see Fig. 1 for a schematic), we develop the formal definition of the computational-level model.

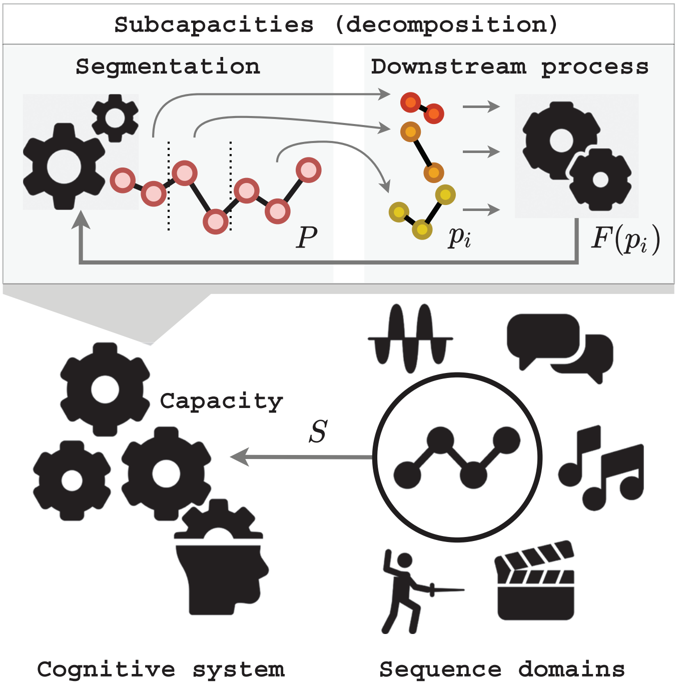

Fig. 1. Segmentation is a core subcomputation in domains including sound, speech, music, action, and event processing (bottom). Segmentation itself can be situated in a functionally decomposed capacity such that domain-specific downstream processes inform which possible segmentations of a given sequence S are best (top). Refer to the main text for definitions of S, P, F, and p_i.

We envision an input sequence $S = (s_1, s_2, \ldots, s_n)$ that captures the idea of a time-ordered representation the cognitive system must work with. Its origin could be sensory encoding at the periphery or deeper, more elaborate processes alike (e.g., an encoding of the acoustic envelope of speech or music, or a compressed representation of a visual scene). As instances of the segmentation problem appear throughout processing hierarchies, their inputs vary in origin and nature. We model the sequence with according generality. Next, we pin down the notion of a downstream cognitive process that computes over segments $p \in \mathscr{P}$ (e.g., a decoder that maps speech segments to phonemes, or a module that maps scene segments to

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

action meanings). Our formalization is agnostic as to what these domain-specific processes, and the theoretical frameworks used to model them, might be. We aim for generality and simply model, with a function $F : \mathscr{P} \mapsto \mathbb{Z}^+$ (over a possibly infinite domain) available at the input, the idea that the process is capable of guiding the placement of boundaries. This is achieved by reporting back some (discretized) aspect of its performance $F(p) \in \mathbb{Z}^+$ (e.g., label probability, likelihood with respect to a generative model, depending on framework). The desired output — a useful segmentation scheme — is modeled as a collection P of disjoint segments jointly making up the input sequence, whose overall appropriateness $V(P)$ with respect to the downstream process is optimal. These modeling choices yield the following formalization.[4]

SEGMENTATION (FORMAL)
Input: a finite sequence $S = (s_1, s_2, \ldots, s_N)$ of length $N \in \mathbb{N}$, with $s_i \in \mathbb{Z}$ and a scoring function $F : \mathscr{P} \mapsto \mathbb{Z}^+$ that maps contiguous subsequences $p = (s_i, s_{i+1}, \ldots, s_{i+q})$ to a positive value $F(p)$.
Output: a segmentation of S into contiguous subsequences, $P = ((s_1, s_2, \ldots), \ldots, (\ldots, s_{N-1}, s_N))$, where segments are disjoint, $\forall p_i, p_j \in P : p_i \cap p_j = \emptyset$, and span the original sequence, $\bigcup_{i=1}^{|P|} p_i = S$, such that its overall value $V(P) = \sum_{p \in P} F(p)$ is maximum.[5]

4. Assumptions about segmentation

So the picture that emerges is that [features] as exhibited
in [various forms of the input] can effectively predict
those procedures which, assuming that their use is not inhibited,
allow us to declare the segmentation problem licked.
— Cutler (1994).

To determine the course of our analyses, we survey views on the computational properties of the segmentation problem. We illustrate with examples and synthesize core intuitions.

4.1. Problem properties: Segmentation as a computational challenge

4.1.1. Hardness and complexity

Segmentation problems have been widely assumed to be computationally challenging. This is evidenced in explicit statements and in the "solutions" researchers propose after taking onboard certain beliefs about hardness. To illustrate: "Speech recognition is not a simple problem. The auditory system must parse a continuous signal into discrete words" (Friston et al., 2021). "It is hard for a brain, and very hard for a computer" (Poeppel, 2003). "[S]egmentation requires inference over the intractably large discrete combinatorial space of partitions" (Franklin, Norman, Ranganath, Zacks, and Gershman 2020).

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.1.2. Sources of complexity

As is evident in researchers' descriptions, the hardness is attributed to the (presumed) combinatorial explosion involved in the number of possible segmentation schemes — the size of the problem search space is informally taken as the source of computational complexity. Again, to illustrate: "Where should these candidate boundaries be placed? In an extreme case, we could place boundaries at every combination of time points [...] but that would be computationally inefficient given that we can reduce the scope of possibilities" (Friston et al., 2021). "The problem would be enormously complicated by the presence of so many candidates [...]" Brent (1999).

4.1.3. Solutions for complexity

Arguably as a consequence of coupling these intuitions with additional assumptions, the effectiveness of certain solutions has been taken for granted. "From the computational perspective, the aim of research in segmentation [...] is to identify mechanisms [that] reduce these computational burdens by reducing the number of candidate[s]" (Brent, 1999). This position has motivated the search for bottom-up segmentation cues or top-down biases (e.g., priors) that would achieve, among other things, such a narrowing down (e.g., Cutler, 1994; Friston et al., 2021; Teng et al., 2019). "We suggest a different role [of cues] in which they are part of the [segmentation] (rather than decoding) process" (Ghitza, 2012). For instance, researchers may observe environmental (Ding et al., 2017) and neural (Teng, Tian, Rowland, & Poeppel, 2017; Teng, Tian, Doelling, & Poeppel, 2017) regularities suggestive of segment-size constrained segmentation processes (Poeppel & Assaneo, 2020; Poeppel, 2003).

4.2. Core assumptions

This survey reveals a core set of intuition-based assumptions about the computational properties of segmentation:

- Real-world sequences (e.g., speech, music, scenes, and actions) and internal representations alike (e.g., memories of experiences) are "complex, continuous, dynamic flows."
- Cognitive systems need to make use of discrete representations of segments that are appropriate (size- and content-wise) for downstream tasks.
- The problem is "hard" — the obstacle being that there are "too many" possible segmentations of a given sequence.
- Cognitive systems must reduce the possibilities somehow, for example, via bottom-up cues and/or top-down biases.

5. Computational complexity of segmentation

It is generally nonobvious what problems are genuinely (as opposed to merely apparently) hard, which refinements will render a model tractable, or which restrictions will effectively reduce a search space. Intuitions about computational properties of problems are frequently mistaken, hence need to be validated against formal analyses (van Rooij, Evans, Müller, Gedge, & Wareham, 2008). We do so here through the lens of theoretical computer science. This section presents a complexity analysis in two parts according to the assumed properties they examine: search space size and problem hardness.

5.1. Search space of segmentation

We analyze the search space size as a possible source of hardness by envisioning a simple brute-force algorithm. If the number of candidate solutions grows polynomially (i.e., upper-bounded by N^c, where N is the sequence length and c is some constant), then such an algorithm would be tractable. It follows that the outcome of this thought experiment hangs entirely on what is revealed by the combinatorial structure of the search space. We describe the aforementioned growth through combinatorial analysis; first for the unconstrained problem and then including various theoretically motivated constraints.

5.1.1. Unbounded parts

When the size q of the segments is not constrained other than by the length N of the sequence, that is, $q \in [1, N]$, all boundary placements are possible. Notice there is a bijection between binary strings of length $N - 1$ and boundary placements in sequences of length N (Fig. 2). Since the number of possible binary strings of length k is given by 2^k, the number of possible segmentations that use unbounded parts grows as 2^{N-1} (i.e., exponentially).

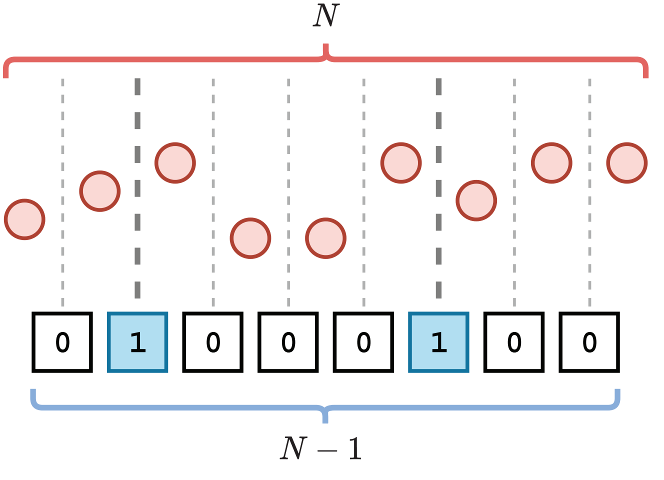

Fig. 2. Binary strings encode boundary placements. A sequence of length N (top) admits $N - 1$ choices of the presence/absence of boundaries (bottom). An example choice of boundaries is shown.

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

5.1.2. Segmentation as integer composition

In order to incorporate various constraints in combinatorial analyses, we draw an analogy between segmentation and *integer compositions* (Fig. 3). This enables us to take an analytic combinatorics approach (Flajolet & Sedgewick, 2009) to the latter and leverage the results to infer properties of the former.

Definition 1 (*Integer composition*). A composition of an integer N is an ordered list $C = (p_1, p_2, \dots, p_k)$ of positive integer parts $p_i \in \mathbb{N}^+$, such that $N = \sum_{p \in C} p$.

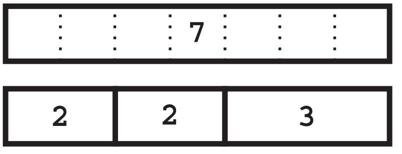

Fig. 3. Segmentation as integer composition. An integer (top), arbitrary parts, and their composition (bottom) stand for a sequence, segments, and a possible segmentation, respectively.

To obtain the growth rate for various restricted cases, we derive generating functions for each, whose coefficients count the number of compositions, and analyze them based on the following lemma.[6]

Lemma 1 (*Growth rate of the coefficients of a rational function*). *Let $S(x) = \sum_{n \geq 0} s_n x^n = \frac{P(x)}{Q(x)}$ be a rational function with $Q(0) \neq 0$ and assume P(x) and $Q(x)$ do not have any roots in common. The general form of the coefficients is $[x^N]S(x) = A^N \Phi(N)$, where A^N is the exponential and $\Phi(N)$ is the subexponential growth factor. Then, the exponential growth rate A of the sequence of coefficients (s_n) is equal to $|\frac{1}{\alpha}|$, where α is the root of $Q(x)$ of smallest modulus (for proof, see Bóna, 2016, Theorem 7.10).*

5.1.3. Lower-bounded parts

We consider integer compositions involving parts $p_i \in [a, N]$, with $1 < a < N$. This corresponds to segmentations whose segment sizes are bounded from below.

Theorem 1. *The number of $[a, N]$-restricted integer compositions of N grows exponentially with N.*

Proof (*sketch*). Through combinatorial arguments, it is possible to construct the generating function

$$G_{LB}(x) := \frac{x^a}{1 - x - x^a}$$

It can be shown, using the intermediate value theorem, that there always exists a root α of $Q(x) = 1 - x - x^a$ satisfying $0 < \alpha < 1$, for any a constrained as above. By Lemma 1, the

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

general form of the coefficients $[x^N]G_{LB}(x)$ has an exponential growth factor A^N, for some $A > 1$. □

5.1.4. Upper-bounded parts

We now consider integer compositions involving parts $p_i \in [1, b]$, with $1 < b < N$. This in turn corresponds to segmentations whose segment sizes are bounded from above.

Theorem 2. *The number of* $[1, b]$*-restricted integer compositions of* N *grows exponentially with* N.

Proof (*sketch*). Through combinatorial arguments, it is possible to construct the generating function

$$G_{UB}(x) := \frac{1 - x}{1 - 2x + x^{b+1}}$$

It can be shown, using the intermediate value theorem, that there always exists a root α of $Q(x) = 1 - 2x + x^{b+1}$ in the interval $(0, \frac{3}{4}]$, for any b constrained as above. By Lemma 1, the general form of the coefficients $[x^N]G_{UB}(x)$ has an exponential growth factor A^N, for some $A > 1$. □

5.1.5. Doubly bounded parts

Finally, we consider compositions involving parts $p_i \in [a, b]$, with $1 < a < b < N$. That is, in analogy to segmentations whose segment sizes are bounded both from above and below.

Theorem 3. *The number of* $[a, b]$*-restricted integer compositions of* N *grows exponentially with* N.

Proof (*sketch*). Through combinatorial arguments, we construct the generating function

$$G_{DB}(x) := \frac{x^a - x^{b+1}}{1 - x - x^a + x^{b+1}}$$

It can be shown, using the intermediate value theorem, that there always exists a root α of $Q(x) = 1 - x - x^a + x^{b+1}$ satisfying $0 < \alpha < 1$, for any a, b as above. By Lemma 1, the general form of the coefficients $[x^N]G_{DB}(x)$ has an exponential growth factor A^N, for some $A > 1$. □

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Taken together, the results proven in this section show that, contrary to common intuition, the vast number of possibilities implied by the segmentation problem is not easily tempered by bounding the size of the segments.

5.2. *Hardness of segmentation*

We showed that intuitive constraints do not render brute-force segmentation tractable. One may be tempted to conclude that this demonstrates the conjectured hardness of the segmentation problem. However, in this section, we present a theorem that contradicts this conclusion. The proof builds on the technique of (polynomial-time) reduction (Arora & Barak, 2009; Garey & Johnson, 1979; Karp, 1972; van Rooij et al., 2019).

Definition 2 (*Polynomial-time reducibility*). Let A and B be computational problems. We say A is *polynomial-time reducible* to B if there exists an algorithm (called *reduction*) to tractably transform instances of A into instances of B such that solutions for B can be easily transformed into solutions for A. This implies that if a tractable algorithm for B exists, it could be used to solve A tractably.

We present such a reduction from SEGMENTATION to a problem in graph theory. Along the way, we will introduce an alternative way of thinking about segmentation at the computational and algorithmic levels.

Theorem 4. SEGMENTATION *is tractable (polynomial-time computable) in the absence of constraints.*

Proof. We will show that, given an arbitrary instance of the segmentation problem, we can tractably construct an instance (with the correct associated output) of a target problem which is itself tractably computable. To begin, we introduce a class of graphs which we use as a stepping stone.

Definition 3 (*Interval graph*). An interval graph is an undirected graph $G = (V, E)$ built from a collection of intervals $\{p_i\} = \{\{x | a_i < x < b_i\}, \ldots\}$, here $x, a_i, b_i \in \mathbb{Z}$, by creating one vertex $v_i \in V$ for each interval p_i and an edge $\{v_i, v_j\}$ whenever the corresponding intervals have a nonempty intersection: $E = \{\{v_i, v_j\} \in V \times V | p_i \cap p_j \neq \emptyset\}$.

Consider an instance of SEGMENTATION. Given an input sequence, it is possible to construct an *interval graph* that satisfies Def. 3. Algorithm 1 demonstrates the procedure.

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Algorithm 1 — Construct segment graph from sequence.

Input:	*Output:*
$S = (s_1, s_2, s_3, ..., s_N),\ s_i \in \mathbb{Z}$	$V = \{(v_1, w_1), ..., (v_q, w_q)\}$
$F : \mathscr{P} \mapsto \mathbb{Z}^+$	$E = \{(v_i, v_j), ...\}$

```
 1: procedure BuildSegmentGraph(S, F)
 2:     P ← [ ]                                   ▷ legal segments
 3:     V ← {}                                    ▷ weighted vertex set
 4:     E ← {}                                    ▷ edge set
▷ construct weighted vertex set
 5:     for i ← 1 to |S| do
 6:         for j ← 0 to |S| − i do
 7:             segment ← S[i : i + j]
 8:             weight ← F(segment) × (−1)
 9:             P.append([i, ..., i + j])
10:             V.append(([i, ..., i + j], weight))
11:         end for
12:     end for
▷ construct edge set
13:     for i ← 1 to |P| − 1 do
14:         for j ← i + 1 to |P| do
15:             if P[i] ∩ P[j] ≠ ∅ then
16:                 E.append((P[i], P[j]))
17:             end if
18:         end for
19:     end for
20:     return (V, E)
21: end procedure
```

Remark 1. Algorithm 1 involves systematically generating all legal segments, computing and negating their weights, checking their pairwise overlap, and using this to construct a graph. We call this object a *segment graph*.

Consider the *time complexity* of Algorithm 1. The elementary instructions are the weight computation (line 8), appending (lines 9, 10, and 16), and set intersection (line 15); all of which are polynomial-time computable (F is assumed to be). We focus now on the number of implied iterations. The loops defined in lines 5 and 6 yield $N + (N - 1) + \cdots + 1$ iterations (the number of possible segments), given by a polynomial:

$$|P_N| = \sum_{k=1}^{N} k = \frac{N(N+1)}{2}$$

The loops defined in lines 13 and 14 yield a number of iterations equal to the number of segment pairs $(p_i, p_j) \in P_N^*$, given by the binomial coefficient $\binom{n}{k}$ with $n = |P_N|$ and $k = 2$,

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

which grows as a quadratic in $|P_N|$ (i.e. 4^{th}-degree polynomial in N),

$$|P_N^*| = \binom{|P_N|}{2} \sim O(N^4)$$

This algorithmic analysis demonstrates that BUILDSEGMENTGRAPH (Algorithm 1) is polynomial-time computable.

Consider next the *correctness* of Algorithm 1. We will show that a segment graph encodes the properties of candidate solutions to an instance of SEGMENTATION. For this, we need the following definitions.

Definition 4 (*Independent sets and maximality*). Let $G = (V, E)$ denote a graph. We call a vertex set $V^* \subseteq V$ an *independent set* if there exist no two vertices $u, v \in V^*$ such that $(u, v) \in E$. Such a set is said to be *maximal* if there exists no vertex $v \in V$ that can be added to V^* without breaking the independence.

Definition 5 (*Dominating sets and minimality*). Let $G = (V, E)$ denote a graph. We call a vertex set $V^* \subseteq V$ a *dominating set* if for all $v \in V$, either $v \in V^*$ or there is an edge $(v, u) \in E$ for some $u \in V^*$. Such a set is said to be *minimal* if there exists no vertex $v \in V^*$ that can be removed without breaking the dominance.

By construction, a legal segmentation is guaranteed to be represented within the segment graph as a subset of vertices with two properties:

- *maximal independence*: vertices are pairwise nonadjacent because segments in a segmentation should be disjoint; since the segments should span the sequence, adding any vertex breaks independence.
- *minimal dominance*: vertices in the graph are either in the subset or adjacent to one of its elements because once a segment subset spans the sequence, any other segment is guaranteed to overlap; since the segments should be disjoint, removing any vertex breaks dominance.

Remark 2. How segment graphs make the structure of the original sequence problem transparent is illustrated in Fig. 4.

A general feature of dominance and independence on arbitrary graphs is useful:

Lemma 2. *An independent vertex set in a graph is a dominating set if and only if it is a* maximal *independent set. Any such set is necessarily also a* minimal *dominating set. (cf. Berge, 1962; Goddard and Henning, 2013).*

It follows from the above and Lemma 2 that if a vertex subset in a segment graph is independent and dominant, then it is a candidate solution (i.e., valid segmentation). A feasible solution has, additionally, minimum weight among candidates: it is a *minimum-weight independent dominating set*. With this, we introduce the formal graph problem we reduce to.

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

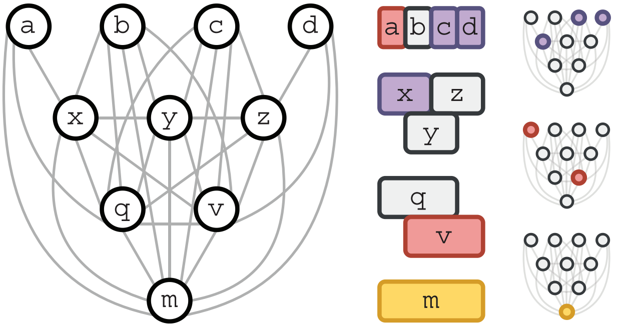

Fig. 4. Segment graph encoding (left) of a sequence of length 4. Nodes correspond to possible segments and edges represent pairwise overlap in the sequence. Possible segments are grouped according to length and overlap (middle). Candidate solutions are vertex subsets in the segment graph (right). Three example segmentations are color coded. (An animation of the BUILDSEGMENTGRAPH procedure may be found online in the Supporting Information section).

Definition 6. MINIMUM-WEIGHT INDEPENDENT DOMINATING SET
Input: A vertex-weighted graph $G = (V, E)$. For each $v \in V$, we have a weight $W(v) \in \mathbb{Z}$.
Output: An independent dominating set $V^* \subseteq V$ such that $Q(V^*) = \sum_{v \in V^*} W(v)$ is minimum.

So far, we have established that, given an instance I_{seq} of SEGMENTATION, we can construct, in polynomial time by Algorithm 1, call it $A(.)$, a corresponding instance $I_{graph} = A(I_{seq})$ of MINIMUM-WEIGHT INDEPENDENT DOMINATING SET. This demonstrates the validity of the reduction and we now finally consider the tractability of the problems.

Though the problem of finding minimum-weight independent dominating sets is *NP-hard* in general and remains so in several special cases (Garey & Johnson, 1979; Liu, Poon, & Lin, 2015), the following input restriction is relevant.

Lemma 3. MINIMUM-WEIGHT INDEPENDENT DOMINATING SET *is polynomial-time computable provided the input graph is an* interval graph *(for proof, see Chang, 1998, Theorem 2.4).*

Recall that the restriction required by Lemma 3 is guaranteed by our reduction. Hence, we conclude SEGMENTATION is tractably computable, which completes the proof. □

Note that the proof of Theorem 4 does not feature bounds on segment size. This result shows that, under plausible assumptions, the problem can be efficiently solved in the absence of such constraints. Our combinatorial and complexity analyses, therefore, suggest the need to rethink the specific conditions where such theoretically motivated constraints might hold, if at all. We discuss these implications next and situate our contributions in the broader landscape.

6. Discussion

Computational feasibility is a widespread concern that motivates choices in the framing and modeling of natural and artificial intelligence. While implicit or informal assumptions

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

abound, the reality may turn out to be counterintuitive as they are examined formally. Here, we undertook a formal examination of the existing computational assumptions about *Segmentation*. Using complexity-theoretic tools, we mathematically proved two sets of results that run counter to commonly held assumptions: (1) the search space is large but placing intuitive constraints does not alleviate the issue; and (2) a computational model that formalizes a cross-domain conceptualization of segmentation is tractably computable in the absence of widely adopted constraints to address the assumed hardness. In other words, to solve the segmentation problem, cognitive systems must find a needle in a vast but orderly haystack – the latter has enough structure that there exist efficient strategies for doing so in spite of its size.

Beyond our proofs, we set the groundwork for further refinements of segmentation theory and its computational analyses: (1) we contributed a formalization of the computation that satisfies a domain-agnostic specification; (2) we illustrated the relationship between segmentation and integer compositions, which makes the search space amenable to asymptotic analyses; and (3) we built a bridge from segmentation as originally defined on sequences to the mathematics of graphs, which opens up alternative formalisms to model the problem and to think about it algorithmically. A desirable consequence of translating problems between formal domains is that structure which was originally hidden from view may become visible. For instance, as a side-effect of encoding instances of the segmentation problem as segment graphs (Fig. 4; Algorithm 1), the growth of the number of possible segments is made explicit (i.e., a polynomial function of the input size). More generally, once segmentation is conceptualized as a graph problem, this opens up the area of parallel graph algorithms (e.g., Balayogan & Pandu Rangan, 1995; Bertossi & Bonuccelli, 1987) as a fruitful source of hypotheses for algorithmic-level explanations.

Our results challenge existing intuitions about hardness of the segmentation problem and its sources of complexity, and by extension question the motivation of proposed solutions and their associated empirical research foci. For instance, concerns about search space size and what mitigates it may be misplaced. The space of possible segments is not exponential to begin with; the space of segmentations is. However, the bounds on segment size we examined here are not a source of complexity: that is, even when introducing the presumed constraints, either individually or combined, the space remains infeasibly large (i.e., exponential). Left unexamined, this may still appear to support the conjectured hardness of the problem. But our tractability proof challenges this intuitive conclusion. It demonstrates that no assumptions about bottom-up segmentation cues or top-down biases on segment properties are necessary to make the formal problem tractable. These proofs run counter to the computational efficiency concerns that partially motivate segmentation theories. For instance, proposals that argue from minimal units of representation (cf. Pöppel, 1997), temporal integration limits of neuronal populations (cf. Overath, McDermott, Zarate, & Poeppel, 2015), intrinsic oscillatory timescales (cf. Ghitza, 2012; Wolff et al., 2022), bottom-up segmentation cues (e.g., Giraud & Poeppel, 2012), and top-down biases on candidate search (e.g., Friston et al., 2021), each to some extent build on the supposition of problem hardness, search space size, and various sources of complexity. Computational learning and simulations further lend some indirect support to the idea that segmentation might not represent a resource bottleneck in every case (Adolfi, Bowers, & Poeppel, 2022). This suggests that intractability concerns, if any, might be better placed, for instance, on the

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

domain-specific processes that compute over segments rather than the boundary placement itself.

Together, the results proven here caution against intuitive notions about the complexity properties of computational problems driving empirical programs, and demonstrate the need and benefits of critically assessing their soundness (something that is too rarely done explicitly; but see, e.g., van de Braak, de Haan, van Rooij, and Blokpoel, 2022; Woensdregt et al., 2021; van de Pol, van Rooij, and Szymanik, 2018; Rich, Blokpoel, de Haan, and van Rooij, 2020; Zeppi and Blokpoel, 2017 for notable exceptions). Whenever intuitions are challenged, this enables researchers to reevaluate the current meta-theoretical calculus (cf. Guest & Martin, 2021), and to redirect efforts as ideas shift regarding what evidence is relevant to collect. For instance, if researchers believe that a certain problem is computationally hard and that some set of neural and environmental regularities might speak to constraints that make it tractable, then they would be inclined to look for those regularities that satisfy such a requirement. If, however, the original belief is removed, the target regularities or the kinds of experiments that are adequate to test their putative role might be different.

We close with a similar word of caution about interpreting our results. These results are to some degree tied to the particular formalization we put forth. While modeling choices were motivated and they bear some generality, alternative theoretical commitments are conceivable. For instance, an extended model could allow for multiple unsegregated high-dimensional input streams with potentially overlapping output segments; it is an open question whether such a model would have different complexity properties. Going forward, it will be important to identify the precise conditions under which the segmentation problem crosses the intractability boundary (cf. Stege, 2012; van Rooij, 2015). We view our analyses, therefore, not as the last word on the computational complexity of segmentation but rather as initial words in a conversation with a sound formal basis.

Notes

1 We bear in mind various distinctions: (1) *problems* as they are intuited by researchers and *computational problems* as formalized in computer science; (2) hypothesized *capacities* of cognitive systems, the *real-world capacities* they allude to, and researchers' *explanations* of them which may include conjectured *(sub)computations*. We use computational-level theorizing and analysis as a way to explain cognitive capacities (cf. Marr, 1982), initially related to informal problems, through the formal definition of computational problems and associated computations. We only have access to the real-world capacities through this explanatory process.

2 Segmentation relates closely to computations whose names vary depending on time period, cognitive domain, and theoretical framework: chunking, sampling, discretization, integration, grouping, packaging, quantization, sequencing, segregation, parsing, temporal pooling, temporal gestalt, boundary placement, and temporal attention.

3 Without loss of generality, here "best" could be replaced by "good enough" and our formal results would still apply.

4 For succinctness, we omit the cumbersome set notation for sequences, $\{(i, s_i), \ldots, (n, s_n)\}$, and we slightly abuse set operation notation.

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

5 We model segmentation as optimization without loss of generality. Our results represent an upper bound on the complexity of the problem with respect to this modeling choice.

6 Here, we present proof sketches and direct the reader to https://arxiv.org/abs/2201.13106 for the full proofs.

Acknowledgments

We are grateful to the Cognitive Science Society for awarding the Computational Modeling of Higher-level Cognition prize and the Disciplinary Diversity & Integration prize based on this work. We thank the Computational Cognitive Science group at the Donders Institute for Brain, Cognition, and Behaviour for discussions, Nils Donselaar for invaluable feedback on a previous version that helped improve the manuscript, and Ronald de Haan for comments on future directions of this work. We thank four anonymous reviewers and one anonymous meta-reviewer for thoughtful comments, and Reviewer 1 in particular for a comprehensive, constructive, and educational review. FA thanks David Poeppel for support and discussions on auditory segmentation. TW was supported by NSERC Discovery Grant 228104-2015. This article builds on a conference paper by the same authors titled *Computational Complexity of Segmentation* that appeared in the Proceedings of the Annual Meeting of the Cognitive Science Society.

Open Access funding enabled and organized by Projekt DEAL.

References

Adolfi, F., Bowers, J. S., & Poeppel, D. (2022). Successes and critical failures of neural networks in capturing human-like speech recognition. *arXiv preprint arXiv:2204.03740*.

Arora, S., & Barak, B. (2009). *Computational complexity: A modern approach*. Cambridge; New York: Cambridge University Press.

Balayogan, V. B., & Pandu Rangan, C. (1995). Parallel algorithms on interval graphs. *RAIRO - Theoretical Informatics and Applications*, 451–470.

Baldwin, D., Andersson, A., Saffran, J., & Meyer, M. (2008). Segmenting dynamic human action via statistical structure. *Cognition*, *106*(3), 1382–1407.

Bechtel, W., & Shagrir, O. (2015). The non-redundant contributions of Marr's three levels of analysis for explaining information-processing mechanisms. *Topics in Cognitive Science*, *7*(2), 312–322.

Berge, C. (1962). *The theory of graphs and its applications*. Methuen.

Bertossi, A. A., & Bonuccelli, M. A. (1987). Some parallel algorithms on interval graphs. *Discrete Applied Mathematics*, 101–111.

Bóna, M. (2016). *Introduction to enumerative and analytic combinatorics* (2nd edition). Discrete Mathematics and Its Applications. Boca Raton, FL: CRC Press, Taylor & Francis Group.

Brent, M. R. (1999). Speech segmentation and word discovery: A computational perspective. *Trends in Cognitive Sciences*, *3*(8), 294–301.

Chang, M. S. (1998). Efficient algorithms for the domination problems on interval and circular-arc graphs. *SIAM Journal on Computing*, *27*(6), 1671–1694.

Cummins, R. (2000). How doesit work?" versus" what are the laws?": Two conceptions of psychological explanation. In F.C. Keil & R.A. Wilson (Eds.), *Explanation and cognition* (pp. 117–144).

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Cutler, A. (1994). The perception of rhythm in language. *Cognition*, 79–81.

Ding, N., Patel, A. D., Chen, L., Butler, H., Luo, C., & Poeppel, D. (2017). Temporal modulations in speech and music. *Neuroscience & Biobehavioral Reviews*, *81*, 181–187.

Egan, F. (2017). Function-theoretic explanation and the search for neural mechanisms. In D. M Kaplan (Ed.), *Explanation and integration in mind and brain science* (pp. 145–163). Oxford University Press.

Farbood, M. M., Rowland, J., Marcus, G., Ghitza, O., & Poeppel, D. (2015). Decoding time for the identification of musical key. *Attention, Perception, & Psychophysics*, *77*(1), 28–35.

Flajolet, P., & Sedgewick, R. (2009). *Analytic combinatorics*. Cambridge; New York: Cambridge University Press.

Franklin, N. T., Norman, K. A., Ranganath, C., Zacks, J. M., & Gershman, S. J. (2020). Structured Event Memory: A neuro-symbolic model of event cognition. *Psychological Review*, *127*(3), 327–361.

Friston, K. J., Sajid, N., Quiroga-Martinez, D. R., Parr, T., Price, C. J., & Holmes, E. (2021). Active listening. *Hearing Research*, *399*, 107998.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. A Series of Books in the Mathematical Sciences. New York: W. H. Freeman.

Geerligs, L., van Gerven, M., & Güçlü, U. (2021). Detecting neural state transitions underlying event segmentation. *NeuroImage*, 118085.

Ghitza, O. (2012). On the role of theta-driven syllabic p arsing in decoding speech: Intelligibility of speech with a manipulated modulation spectrum. *Frontiers in Psychology*, *3, 238*.

Giraud, A. L., & Poeppel, D. (2012). Cortical oscillations and speech processing: Emerging computational principles and operations. *Nature Neuroscience*, *15*(4), 511–517.

Goddard, W., & Henning, M. A. (2013). Independent domination in graphs: A survey and recent results. *Discrete Mathematics*, *313*(7), 839–854.

Guest, O., & Martin, A. E. (2021). On logical inference over brains, behaviour, and artificial neural networks. *PsyArXiv*, https://doi.org/10.31234/osf.io/tbmcg.

Himberger, K. D., Chien, H. Y., & Honey, C. J. (2018). Principles of temporal processing across the cortical hierarchy. *Neuroscience, 161-174*.

Jeunehomme, O., & D'Argembeau, A. (2018). Event segmentation and the temporal compression of experience in episodic memory. *Psychological Research, 481-490*.

Jin, P., Lu, Y., & Ding, N. (2020). Low-frequency neural activity reflects rule-based chunking during speech listening. *eLife*, *9*, e55613.

Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (Eds.), *Complexity of computer computations* (pp. 85–103). Springer.

Kolokolova, A., & Nizamee, R. (2014). Approximating solution structure of the Weighted Sentence Alignment problem.

Liu, C. H., Poon, S. H., & Lin, J. Y. (2015). Independent dominating set problem revisited. *Theoretical Computer Science*, *562*, 1–22.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. Cambridge, MA: MIT Press.

Meyer, M., Baldwin, D. A., & Sage, K. (2011). Assessing young children's hierarchical action segmentation. *Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 33*. *No*. 33.

Miłkowski, M. (2013). Reverse-engineering in Cognitive-Science. In K. Talmont-Kaminski & M. Miłkowski (Eds.), *Regarding the mind, naturally: Naturalist approaches to the sciences of the mental*. Cambridge Scholars Publishing.

Overath, T., McDermott, J. H., Zarate, J. M., & Poeppel, D. (2015). The cortical analysis of speech-specific temporal structure revealed by responses to sound quilts. *Nature Neuroscience*, *18*(6), 903–911.

Poeppel, D. (2003). The analysis of speech in different temporal integration windows: Cerebral lateralization as 'asymmetric sampling in time'. *Speech Communication*, *41*(1), 245–255.

Poeppel, D., & Assaneo, M. F. (2020). Speech rhythms and their neural foundations. *Nature Reviews Neuroscience*, *21*(6), 322–334.

Pöppel, E. (1997). A hierarchical model of temporal perception. *Trends in Cognitive Sciences*, *1*(2), 56–61.

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Rich, P., Blokpoel, M., de Haan, R., & van Rooij, I. (2020). How intractability spans the cognitive and evolutionary levels of explanation. *Topics in Cognitive Science*, 1382–1402.
Stege, U. (2012). The impact of parameterized complexity to interdisciplinary problem solving. In Bodlaender, H.L., Downey, R., Fomin, F.V., Marx, D.(Eds.), *The multivariate algorithmic revolution and beyond* (pp. 56–68). Springer.
Teng, X., Cogan, G. B., & Poeppel, D. (2019). Speech fine structure contains critical temporal cues to support speech segmentation. *NeuroImage*, *202*, 116152.
Teng, X., Tian, X., Doelling, K., & Poeppel, D. (2017). Theta band oscillations reflect more than entrainment: Behavioral and neural evidence demonstrates an active chunking process. *European Journal of Neuroscience, 2770-2782*.
Teng, X., Tian, X., Rowland, J., & Poeppel, D. (2017). Concurrent temporal channels for auditory processing: Oscillatory neural entrainment reveals segregation of function at different scales. *PLoS Biology*, *15*(11), e2000812.
Tenney, J., & Polansky, L. (1980). Temporal gestalt perception in music. *Journal of Music Theory*, *24*(2), 205.
Tillmann, B. (2012). Music and language perception: Expectations, structural integration, and cognitive sequencing. *Topics in Cognitive Science*, *4*(4), 568–584.
van de Braak, L. D., de Haan, R., van Rooij, I., & Blokpoel, M. (2022). Intractability of Bayesian belief-Updating during communication. *Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 44. No*. 44.
van de Pol, I., van Rooij, I., & Szymanik, J. (2018). Parameterized complexity of theory of mind reasoning in dynamic epistemic logic. *Journal of Logic, Language and Information*, 255–294.
van Rooij, I. (2015). How the curse of intractability can be cognitive science's blessing. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2.
van Rooij, I., Blokpoel, M., Kwisthout, J., & Wareham, T. (2019). *Cognition and intractability: A guide to classical and parameterized complexity analysis*. Cambridge University Press.
van Rooij, I., Evans, P., Müller, M., Gedge, J., & Wareham, T. (2008). Identifying sources of intractability in cognitive models: An illustration using analogical structure mapping. *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. *30*. No. 30.
Woensdregt, M. S., Spike, M., de Haan, R., Wareham, T., van Rooij, I., & Blokpoel, M. (2021). Why is scaling up models of language evolution hard? *Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 43. No*. 43.
Wolff, A., Berberian, N., Golesorkhi, M., Gomez-Pilar, J., Zilio, F., & Northoff, G. (2022). Intrinsic neural timescales: Temporal integration and segregation. *Trends in Cognitive Sciences*, *26*(2), 159–173.
Wyble, B., & Bowman, H. (2019). Temporal segmentation for faster and better learning. In *2019 Conference on Cognitive Computational Neuroscience*. Berlin, Germany. Cognitive Computational Neuroscience.
Zacks, J. M., Braver, T. S., Sheridan, M., Donaldson, D. I., Snyder, Z., Ollinger, J. M., Buckner, R. L., & Raichle, M. E. (2001). Human brain activity time-locked to perceptual event boundaries. *Nature Neuroscience*, *4*(6), 651–655.
Zeppi, A., & Blokpoel, M. (2017). Mindshaping the world can make mindreading tractable: Bridging the gap between philosophy and computational complexity analysis. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 6.

Supporting Information

Additional Supporting Information may be found online in the supporting information tab of this article:

17568765, 2023, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/tops.12629 by Cochrane Germany, Wiley Online Library on [15/05/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License