*From Dionysius Emerges Apollo*

# Learning Patterns and Abstractions from Perceptual Sequences

Dissertation

zur Erlangung des Grades eines
Doktors der Naturwissenschaften

der Mathematisch-Naturwissenschaftlichen Fakultät
und
der Medizinischen Fakultät
der Eberhard-Karls-Universität Tübingen

vorgelegt
von

Shuchen Wu
aus Jingdezhen, China

2025

# Declaration

Ich erkläre, dass ich die zur Promotion eingereichte Arbeit mit dem Titel: "Learning Patterns and Abstractions from Perceptual Sequences" selbständig verfasst, nur die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche gekennzeichnet habe. Ich versichere an Eides statt, dass diese Angaben wahr sind und dass ich nichts verschwiegen habe. Mir ist bekannt, dass die falsche Abgabe einer Versicherung an Eides statt mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft wird.

I hereby declare that I have produced the work entitled "Learning Patterns and Abstractions from Perceptual Sequences", submitted for the award of a doctorate, on my own (without external help), have used only the sources and aids indicated and have marked passages included from other works, whether verbatim or in content, as such. I swear upon oath that these statements are true and that I have not concealed anything. I am aware that making a false declaration under oath is punishable by a term of imprisonment of up to three years or by a fine.

*Tübingen, May 22nd 2025*

Shuchen Wu

| | |
|---|---|
| Tag der mündlichen Prüfung: | 22/05/2025 |
| | |
| Dekan der Math.-Nat. Fakultät: | Prof. Dr. Thilo Stehle |
| Dekan der Medizinischen Fakultät: | Prof. Dr. Bernd Pichler |
| | |
| 1. Berichterstatter: | Prof. Dr. Eric Schulz |
| 2. Berichterstatter: | Prof. Dr. Peter Dayan |
| | |
| Prüfungskommission: | Prof. Dr. Eric Schulz |
| | Prof. Dr. Peter Dayan |
| | Prof. Dr. Felix Wichmann |
| | Prof. Dr. Georg Martius |

*From Dionysius Emerges Apollo*

# Learning Patterns and Abstractions from Perceptual Sequences

Shuchen Wu

*May 22nd 2025*
Version: Final

*“Reality is merely an illusion, albeit a very persistent one.”*

— Albert Einstein

# Summary

Immersed in chaotic and noisy sensory streams, we perceive a structured world. From learning stimulus-response pairings to grouping nearby visual parts, and from grammar acquisition to learning recurring sequential patterns, cognition irresistibly and swiftly breaks a high-dimensional sensory stream into familiar parts and gradually unveils their relations. Why do structures emerge, and how do they help us learn, generalize, and predict? What underlying computational principles give rise to this fundamental aspect of perception and intelligent behavior?

A sensory stream — simplified to an extreme — is a one-dimensional discrete sequence. In the process of learning such sequences, we naturally segment them into familiar parts — a well-known phenomenon called chunking. In the first project, I investigated the factors that influence chunking behavior in a serial reaction time task. I showed that humans sensitively adapt to underlying chunks in sequences while exhibiting a resource-rational trade-off between speed and accuracy.

Taking a step further, I built models that capture the chunk learning process on a computational and algorithmic level. The model learns chunks and parses sequences one chunk after another. From a normative standpoint, I proposed that chunking can be a rational way for an intelligent agent to discover recurring patterns and nested hierarchies in sequences, and, in turn, factorize sequences more effectively. I showed that sequential chunks can be learned as readily accessible primitives, ready for reuse, transfer, composition, and mental stimulation. This consequently allows the model to build up a complex understanding of the world by seeing the new via composing the known. I demonstrated and generalized this model's ability to learn hierarchies in both single and multi-dimensional sequence domains, and showed its applicability as an unsupervised pattern discovery algorithm.

The second part of the investigation dives from the concrete domain to the abstract domain. I taxonomized two abstract sequence motifs and studied their implications for sequence memory recall. Behavioral evidence suggests that humans readily exploit redundancies in patterns in an abstract space for more efficient memory compression while transferring these motifs to novel sequences.

Taking a step further, I propose a non-parametric hierarchical variable learning model that combines abstraction with chunking—abstracting those chunks that appear in similar contexts as variables while learning chunks also on a symbolic variable level — gradually unearthing abstractions and discovering invariant patterns on a symbolic level. I demonstrate the algorithm's resemblance to human learning behavior and compare it with large language models.

Taken together, this thesis suggests that chunking and abstraction as simple computational principles give rise to structured knowledge acquisition in sequences with underlying hierarchical structure, from simple to complex, from concrete to abstract, enabling the recursive construction of the highly complex from the ground up. I demonstrated the models' resemblance to human behavior and their algorithmic applications to discover patterns.

# Zusammenfassung

Wenn wir in chaotische und laute Sinnesströme eintauchen, nehmen wir dennoch eine Welt mit fester Struktur wahr. Von der Zuordnung von Reiz-Reaktions-Paaren über das Gruppieren nahe beieinanderliegender visueller Elemente bis hin zum Grammatikerwerb und dem Erlernen wiederkehrender sequenzieller Muster: Die Kognition zerlegt unaufhaltsam und blitzschnell einen hochdimensionalen sensorischen Strom in vertraute Einheiten und deckt nach und nach deren Beziehungen auf. Aber warum entstehen diese Strukturen? Wie helfen sie uns beim Lernen, Verallgemeinern und Vorhersagen? Welches zugrundeliegende Berechnungsprinzip führt zu solch einer grundlegenden Komponente, die unser wertvolles Wahrnehmungs- und Intelligenzverhalten ermöglicht?

Ein sensorischer Strom kann stark vereinfacht als eindimensionale, diskrete Sequenz betrachtet werden. Beim Lernen solcher Sequenzen neigen wir dazu, sie in vertraute Teile zu unterteilen – eine tief verwurzelte Tendenz, die als Chunking bekannt ist.

Im ersten Projekt untersuchte ich, welche Faktoren das Chunking-Verhalten in einer seriellen Reaktionszeitaufgabe beeinflussen. Ich zeigte, dass Menschen sich flexibel an die zugrunde liegenden Chunks von Sequenzen anpassen und dabei einen ressourcenrationalen Kompromiss zwischen Geschwindigkeit und Genauigkeit eingehen.

In einem weiteren Schritt habe ich rechnerische und algorithmische Modelle entwickelt, die den Chunking-Prozess erfassen. Das Modell lernt Chunks und verarbeitet Sequenzen in diesen Einheiten. Aus normativer Perspektive schlug ich vor, dass Chunking für einen intelligenten Agenten rational ist, um wiederkehrende Muster und verschachtelte Hierarchien zu entdecken und so die Sequenzen effektiver zu faktorisieren. Ich zeigte, dass sequentielle Chunks als leicht zugängliche Primitive erlernt werden können, die zur Wiederverwendung, Übertragung, Komposition und mentalen Simulation bereitstehen. Dadurch kann das Modell ein tiefes Verständnis der Welt aufbauen, indem es das Neue durch die Kombination des Bekannten erschließt.

Ich demonstrierte die Fähigkeit des Modells, Hierarchien in ein- und mehrdimensionalen Sequenzen zu erlernen, und verallgemeinerte seine Anwendung als unüberwachter Algorithmus zur Mustererkennung.

Der zweite Teil der Untersuchung befasste sich mit dem Übergang von der konkreten zur abstrakten Domäne. Ich habe zwei abstrakte Sequenzmotive taxonomisch erfasst und ihre Auswirkungen auf das Gedächtnisretrieval untersucht. Das Verhalten zeigt, dass Menschen Redundanzen in Mustern im abstrakten Raum nutzen, um eine effizientere Gedächtniskompression zu erreichen, während sie diese abstrakten Motive auf neue Sequenzen übertragen.

Schließlich gehe ich noch einen Schritt weiter und stelle ein nicht-parametrisches, hierarchisches Modell zum Lernen von Variablen vor, das Chunking mit Abstraktion kombiniert. Hierbei werden Chunks, die in ähnlichen Kontexten wie Variablen auftreten, abstrahiert, während auch auf einer symbolischen Ebene gelernt wird, um schrittweise Abstraktionen und invariante Muster zu entdecken. Ich zeige, dass der Algorithmus dem menschlichen Lernverhalten ähnelt und vergleiche ihn mit großen Sprachmodellen.

Zusammenfassend zeigt diese Arbeit, dass Chunking und Abstraktion als grundlegende Rechenprinzipien zu einem strukturierten Wissenserwerb in Sequenzen mit zugrunde liegender hierarchischer Struktur führen. Vom Einfachen zum Komplexen, vom Konkreten zum Abstrakten wird so die rekursive Konstruktion hochkomplexer Strukturen von Grund auf ermöglicht. Ich habe nicht nur die Ähnlichkeit der Modelle mit dem menschlichen Verhalten aufgezeigt, sondern auch ihre algorithmischen Anwendungen zur Mustererkennung verdeutlicht.

# Acknowledgement

> *Every person is more than just oneself; s/he also represents the unique, the very special, and always significant and remarkable point at which the world's phenomena intersect, only once in this way, and never again.*
>
> **— Hermann Hesse**

I would like to express my deepest gratitude to the members of my supervisory committee, whose guidance and mentorship have shaped me into a better scientist throughout this journey. First and foremost, I would like to express my heartfelt gratitude to Eric, who has been a great mentor and offered unwavering support and advice throughout my PhD. He also demonstrated what it means to be a great PI and writer and taught me to see the bright sides. Eric's wonderful personality to see the best in people and to welcome new collaborators and ideas have attracted outstanding individuals to the lab. It is especially encouraging to witness how the lab has transformed from a group of three people to a research institute. I would also like to express my deep gratitude to Peter, whose diligence and sense of responsibility have set a role model for scientific excellence. He has fostered an inclusive environment at the institute for many through pandemic or political hardships. Peter's ability to attract brilliant minds to Tübingen has made precious opportunities to meet bright minds plentiful, which is instrumental to my growth. Felix, who has supported me since my master's years, has been a constant source of encouragement. Felix was the one who opened the door for me to come to Tübingen, and for that, I will always be profoundly grateful.

I am indebted to excellent scientists such as Mirko and Susanne, who showed me the meticulousness of a psychologist in scrutinizing every detail of experimental design. I am also thankful to Ishita Dasgupta and Noemi Éltetö for helping me start my first projects. I have worked with wonderful students, including Mehmet Yörüten and Atilla Schreiber, who have taught me a great deal. I am thankful for colleagues who

made the group a lively place of inspiring discussions and plenty of chuckles and ping-pong games, including Franziska Brändle, Akshay Jagadish, Tankred Saanum, Marcel Binz, Alex Kipnis, Tobias Ludwig, Julian Coda-Forno, Luca Schulze Bischoff, Kristin Witte, Xin Sui, Alicia Guzmán, and Can Demircan.

I am grateful for having met a network of wonderful colleagues and friends from the institute and beyond, including Kaidi Shao, Judith Borowski, Robert Geirhos, Ju-Young Lee, Georgy Antorov, Yan Ma, Weiyi Xiao, Tianyuan Teng, Surabhi Nath, Sebastian Burjins, Gabriele Belluci, Franziska Bröker, Hanqi Zhou, Chuyu Yang, Wenting Wang, Qi Wang, Rui Tian, Charline Tessereau, Oleg Solopchuk, Tingke Shen, Daniel Shani, Lion Schulz, Turan Orujlu, Azadeh Nazemorroaya, David Nagy, Kevin Lloyd, Ruiqi He, Junhao Liang, Chris Gagne, Sara Ershadmanesh, Florian Birk, Sahiti Chebolu, Stephan Bucher, Aenne Brielmann, Mihaly Banyai, Jiatong Liu, Jingyou Zou, Shervin Safarvi. I am thankful for the labs of Stanislas Dehaene, Chris Summerfield, Timothy Behrens, Nikos Logothetis, Liping Wang, Tianming Yang, Zhang Peng, Anli Liu, Zhe Chen, and Zeynep Akata for hosting my visit during the PhD, and for exemplifying to me scientific works of utmost excellence, from which I have been truly inspired. I am thankful for Fernand Gobet's advice and reassurances during the discussion at restaurant Dionysius in Cogsci.

I owe a debt to the support of MPI staff: Kathrin Prax, Bianca Gäßer, Susan Fischer, Blake Fitch, Haydar Martin, and Joachim Werner. Joachim always saved me when I forgot to bring my laptop battery or needed server PHP debugging. Bianca and Kathrin are lifesavers, helping me tackle urgent paperwork in time.

I am deeply grateful to my experimental participants, who have contributed their time to the experiments. Their insightful suggestions for improving experimental design and even their one-time help with debugging went beyond what I could have anticipated. I was especially heartened by the encouraging feedback I received from participants expressing how much they enjoyed the experiments. Their responses have deepened my appreciation for both the data and the value of thoughtful user interface design, reinforcing my commitment to make the most of their contributions.

I am thankful for the anonymous reviewers of our papers. Their comments have reassured the work's contribution, and their suggestions have significantly improved this work.

Words cannot express my appreciation for the community at 1-West and Fichtehaus. Never would I expect myself to find anywhere else so close to feeling at home. Paula and Linda advocated for my move, believing I needed connections in Tübingen. I am deeply grateful for my flatmates, who have been similar to an extension of my

family, including Linda, Ritika, Ruben, Lucia, Maxi, Jan, David, Charlotte, Juliane, Jo, Gregor, Kimo, Chiara, Luisa, Paula, Ximena, Malte, Flo, Chrisi, Steffi, Andy, Kristi, Christian, and Pascal, for the joys and sorrows that we shared, and the years that we grew together. Friends in the house, from whom I have been truly inspired through our discussions and experience organizing activities together, including Isa, Jacob, Anusha, Paul, Isabel, Nikos, Julius, Bea, Domi, Sophia, Pauline, Selena, Doro S., Doro R., Franziska, Sophie, Ruben, Felix, Roxana, Alexander, Jan-Lukas, Max, Lidewei, Praslav, and Parsival. I learned from human behavior and democratic decision-making to mastering Linsen Spätzle and hosting an unforgettable party. Thank you for making me feel constantly supported and acknowledged and for sharing a journey of growth with compassion.

I am especially grateful to Qinhan for always listening to me and my parents' support throughout the years. My mom taught me to appreciate beauty and art, while my dad exemplified a spirit of embracing challenges and constantly seeking self-improvement. My grandma taught me to love people and be optimistic, and my grandpa taught me to be curious and try new things. Mimi Wu's unconditional love and affection as a tabby cat have been a constant source of comfort.

I am lucky to have met great minds whose ideas had long-lasting inspiration for this work before the start of this PhD. Kevin Martin for his advice on mindfulness; Matthew Cook for models of computation; Marin Osaki and Lukas Vogelsang for many things, including floating along Aare, Chenxi Wu for variables; Lee Sharky for grounding startups; Aniruddh Galgali and Wenliang Li for London visits and Kai Sandbrink for being at almost every conference that I am also going to.

Finally, I am grateful to Ralf Haefner for introducing me to embark on this journey from my undergraduate years.

# Contents

**I Introduction** **1**

**1 Introduction** **3**

**II From simple to complex** **13**

**2 Chunking in serial reaction time tasks** **15**
2.1 The emergence of action primitives from sequences . . . . . 15
2.2 Summary of the article . . . . . 16
2.3 Discussion . . . . . 17
2.4 Limitation and future work . . . . . 19
2.5 Article Status . . . . . 20
2.6 Author Contributions . . . . . 20

**3 Chunking to compose — a source of infinite combination using finite means** **21**
3.1 Related Work . . . . . 23
3.2 Summary of the work . . . . . 24
3.3 Discussion . . . . . 27
3.4 Article Status . . . . . 30
3.5 Author Contributions . . . . . 30

**III From concrete to abstract** **31**

**4 Beyond concrete sequences — Formulating and testing two types of motif learning in sequence learning and transfer** **33**
4.1 Toward a Taxonomy of Abstract Motifs . . . . . 35
4.2 Summary of the article . . . . . 35
4.3 Discussion . . . . . 37
4.4 Article Status . . . . . 39
4.5 Author Contributions . . . . . 39

**5 The construction from the simply abstract to the complexly abstract, layer by layer** **41**
5.1 Related work on modeling abstractions . . . . . 42

5.2 The open question . . . 43
5.3 How abstract concepts may grow inside the mind . . . 43
5.4 Summary of the Article . . . 44
5.4.1 Sequences with nested abstract hierarchical structures . . . 45
5.4.2 Two ingredients of abstraction . . . 46
5.5 Discussion . . . 48
5.6 Article Status . . . 50
5.7 Author Contributions . . . 50

**IV Outlook** **51**

**6 Discussion** **53**

**7 Conclusion** **59**

**Bibliography** **61**

**8 Statement of Contribution** **79**

**V Manuscripts** **85**

**Chunking as a rational solution to the speed–accuracy trade-off in a serial reaction time task** **87**

**Learning Structure from the Ground-up—Hierarchical Representation Learning by Chunking** **113**

**Motif Learning Facilitates Sequence Memorization and Generalization** **151**

**Building, Reusing, and Generalizing Abstract Representations from Concrete Sequences** **169**

**9 Afterward: From Dionysius emerges Apollo** **193**

# Part I

## Introduction

# 1 Introduction

> *The propose of science is to find meaningful simplicity in the midst of disorderly complexity.*
>
> **— Herbert Simon**

We effortlessly perceive entities from an immense volume of an ever-changing sensory stream. Consider the process of walking through a bustling city: successive sensations arise from a continuous influx of perception. You notice the swift changes in traffic lights, hear the diverse sounds of car horns, feel the differences of the surfaces you are walking on, and smell the fleeting aroma of a nearby food stand. Amidst this overwhelming flood of sensation, cognition skillfully sifts through the sensory stream, identifying concrete entities such as "traffic signals", "road conditions", and "food stand". This remarkable capability allows us to swiftly navigate and interact with a complex environment.

Contrary to the ease of symbols emerging in our perception, machine learning systems still struggle to learn structured symbols and factorization from data. Nowadays, large-scale artificial neural networks are trained on hundreds of thousands of graphical computing units with various optimization goals to auto-regress data from the internet [127]. These models recently demonstrated remarkable performance in solving problems from recognizing speech and visual objects to playing chess [127, 24, 156, 202]. Despite the triumph on the surface level, it has been argued that the resulting models still do not understand and represent observations as humans do [191, 122]. Prominent models transform, predict, and generate sentences just by learning from lots of data and associating each part of the sentence with others [255], but this feature does not guarantee that entities of features shall reliably emerge within the model. Consequentially, large language models struggle to solve problems that involve symbolic manipulation, such as mathematical problems, which are never present in the training data [255, 181]; connectionist computer vision models still rely heavily on human-segmented scene data, hand denoted tags and human feedback to learn to segment images into visual entities [78, 94, 254, 18]. Modern connectionist AI systems still struggle with the ability to distill object entities from data, and understand object permanence and the interactions and relations amongst these recurring entities — an ability that simple animals are capable of [201, 19, 206].

The consequence of the gap between machine and human perception causes many problems. There are many problem domains that are trivial for humans but notoriously difficult for current connectionist machine-learning algorithms [211, 199]. The set of these problems includes — but is not limited to — continual learning [210, 199], disentangling representations [143], interpretability [102, 143], few-shot generalization [38, 102], and abstractions [200, 134]. These problems limit DNN's reliability in many safety-critical domains, such as the navigation of autonomous cars and medicine [77, 101]. These problems involve precisely the compact, efficient, and reusable representation that comes to our cognition and perception easily. On the other hand, these characteristics are the pronounced characteristics of symbolic systems. As soon as thought can be expressed in a symbolic form, interpretability, generalization, reuse, and transfer can also be expressed in clearly articulated forms.

Earlier approaches to studying artificial intelligence were predominantly symbolic. Herbert Simon, the father of modern artificial intelligence, posited that a system will only be capable of intelligent behavior if it operates and manipulates symbolic structures [203, 161]. AI in the last century was primarily about how precisely defined programs can manipulate symbols, such as automatically arranging and substituting mathematical or logical symbols to arrive at the next step of calculation or deduction [243, 136], to solve equations [146], to parse languages using syntactical trees [242, 23], or to partition images via visual grammar [144, 256]. Symbolic models of AI represent computing entities as symbols and define operations between them, such as the definitions of constants and variables in mathematics, classifications of words into word types and morphological rules in natural language processing, or categorizations of image grammar in computer vision. This approach allows for the flexible reuse and recombination of symbols, which can be clearly implemented in programming languages across different domains. However, symbolic AI faces a fundamental challenge: most real-world data cannot be easily broken down into discrete parts that interact in straightforward ways. As a result, symbolic models have limited applicability and heavily depend on the programmer's choice to design effective symbols and operational rules.

A similar issue arises in probabilistic symbolic AI, where observations are modeled with probabilistic entities to account for noise and uncertainty. When an agent observes multiple sensory inputs that contain noise and makes decisions based on them, it needs to estimate a high-dimensional distribution, often represented as $P(x_1, x_2, \ldots, x_n)$, i.e., a complex global function with many variables. The distribution, by its raw form, is complex and infeasible to compute. However, this computing challenge can be dramatically alleviated if there are groups of random variables that are statistically independent of each other. In this case, then the high dimensional distribution can be factorized into subsets of variables such

as $P(x_1, x_2)P(x_3, x_4)P(\ldots, x_n)$. The factorized form breaks the computation into smaller, independent parts, reducing both the time and complexity of calculations [119, 5]. This approach has been critical in probabilistic models like Markov random fields [129], Bayesian belief networks [169], message-passing algorithms [119], and it also applies to algorithms such as belief propagation [252], the Viterbi algorithm [229], the Kalman filter [113] and learning structural causal models [168]. In computational neuroscience, researchers propose that the brain may similarly factorize high-dimensional sensory information into several independent modular systems, each handling a specific part of the computation in parallel, making use of the advantage of parallel computing neural systems inside the brain [177]. However, probabilistic AI faces challenges similar to symbolic AI: it is notoriously difficult to determine which groups of variables are statistically independent, and can be considered as within a factor to estimate the full distribution [51, 207, 1].

How to come up with symbols and statistically independent groups of entities has been a challenge in symbolic and probabilistic AI. Symbolic AI systems depend heavily on human expertise to define the symbols and rules for manipulating them, while probabilistic AI struggles with the statistical complexity of determining the appropriate factorization without human input. At the same time, the missing perception of entities in connectionist AI contributes to the perception gap between humans and machines, causing their unreliability and lack of interpretability. These types of AI approaches do not have a clear answer to a fundamental feature that our cognitive system handles with ease: how do structured entities emerge in cognition?

Patterns are fundamental to both our perception and actions. We easily recognize concrete entities like “traffic signals”, “road conditions”, and “food stands” from a flood of sensory input, and we effortlessly perform action sequences, such as fetching a bottle, boiling water, and making tea. This question on where do cognitive entities come from — intersecting artificial intelligence and cognitive science — motivates the thesis to explore how cognitive entities emerge from perception, and their subsequent roles in factorization, interpretability, transfer and generalization, and how a computational model describing this process may bridge the gap between AI systems and human cognition.

The ability to perceive symbols and entities from an overwhelming and ever-changing sensory stream has historically been a topic of interest. William James famously described newborns’ experience as a “great blooming, buzzing confusion,” highlighting the immense challenge they face, immersing in noisy and fleeting sensory input to make sense of the world [108]. Over time, the infants all learn to recognize coherent objects and develop sequences of actions to interact between these objects.

Philosophers have long speculated how the mind extracts meaningful patterns from sensory input. The British empiricists, in particular, emphasized that entities, including symbols and ideas, emerge from accumulating experiences from the sensory streams. John Locke proposed that newborns begin with a blank state of mind (tabula rasa) and that perceptual categories and entities form through experience [131]. David Hume described this process as beginning with vivid 'impressions' from perception, which later evolve into more abstract and less intense 'ideas' that are readily retrieved by the thinking process. The empiricists suggest that the mind gathers information and constructs knowledge in an additive manner: as interactions with the world accumulate, more complex ideas develop from earlier experiences [103, 131, 100, 148, 149]. More complex mental structures can build on the previously learned ones [106].

This idea was later taken up by behaviorist psychologists. Through studying animals and their behaviors, they proposed that animals establish behavioral structure by learning associations between stimuli via repeated practice and reinforcement [167], [217, 215, 219, 218]. When stimuli repeatedly occur close together in time or space, the occurrence of one can evoke the memory of the other. Through practice, sequences of actions become associated and enforced [96, 216, 204], allowing animals to execute complex behavioral responses reliably.

Acquiring structured patterns from observation connects to our ability to transfer and generalize knowledge. Aristotle argued that structured perception is key to reasoning and inferring knowledge beyond our limited personal experience [9, 12, 8, 10, 11]. Similarly, Thorndike proposed that animals tend to reuse sequences of responses in new environments when these environments resemble situations they've previously encountered [216].

Before action and association, Gestalt psychologists proposed that the mind has an inherent tendency to organize perception into structure. In vision, for example, we tend to perceive nearby entities together as a whole, and farther entities as separate parts. Gestalt psychologists studied and characterized the tendency to organize complex sensory input into groups of parts, which are integrated into coherent wholes [236, 237]. Through this grouping process, the mind identifies patterns and regularities and simplifies complex images, allowing perception to identify entities that can be related to prior knowledge. Today, the Gestalt grouping laws are still used as guiding principles of visual design to convey messages from abstract logos[116].

While language is considered by some a uniquely human ability [49], patterns and recurring entities are prominent in both concrete and abstract language levels. Concrete patterns manifested in words and phrases appear at the explicit level, and

syntactical, morphological, and grammatical rules recur on an abstract level. Yet these rules still rigidly govern the composition of words into sentences. Understanding those rules is also critical to parse parsing sentences into coherent meanings. Mastering the application and usage of these patterns on concrete and abstract levels is critical for comprehending and using any language in flexible forms, allowing the infinite variations of meanings that language affords to convey [39, 42, 72].

Empiricist philosophers, behavioral and gestalt psychologists, and linguists have historically considered the importance of perceptual entities and have related this feature with the powerful ability of humans to generalize and transfer. In the context of this thesis, to build on the previous observations and use a modern, programmatic approach to study this question, I set up this problem by studying sequences. This is because sequences are the most simplified form of sensory experience. Any sensory signal can be distilled into a series of successive perceptions and actions across different modalities. Discrete sequences are the most abstract simplification of perceptual data, preserving the core aspect of this problem while throwing away the unnecessary complexities. Processing sequences — whether through perceiving, memorizing, or retrieving temporally ordered elements — is fundamental to nearly all human activities, from recalling events and generating actions to using language and enjoying music.

This setup of the problem casts the question into how cognitive entities emerge from perceiving discrete sequences. In cognitive science, this process is characterized by *chunking* [86, 85, 50, 97]. As we parse a sequence such as "DFJKJKJKDFDFJKJKDFDF", chunking refers to our tendency to segregate the sequence into a composition of recurring components, such as "DF"s and "JK"s — a behavior that even small babies and animals exert [172, 214].

Chunking relates to memory organization [28, 151]. When we need to remember a sequence, we tend to break the sequence into several chunks. These chunks are then memorized and recalled as separate entities [145, 152, 52, 26]. Chunks also serve as the units of our memory storage: we can hold between 4 to 7 chunks [152] in our short-term memory. Therefore, knowing longer chunks by heart helps us to remember longer sequences. Imagine having to remember the sequence "052917080461": taken on its own, this sequence might be difficult to recall. However, knowing that it contains both John F. Kennedy's and Barack Obama's date of birth will likely simplify this task. Chunks can be subject to composition; we memorize sequences more easily by organizing them into a nested hierarchy of smaller chunks embedded in bigger chunks [178].

Apart from cognitive processing and memory organization, chunking also helps the organization of action sequences. We build up complex action sequence executions by

piecing together familiar sequence chunks [227, 110, 186, 93, 67, 125], fundamental to the process of planning [222]. Chunking perception, memory, and action into entities as basic units of cognitive processing are critical for language acquisition and usage [172, 178, 114]. Apart from action execution, memory, and sequence parsing, chunking has been observed in other sensory domains and has been theoretically linked to mental compression and optimal pattern discovery [196, 238, 62, 91, 170, 165, 28, 115].

I decided to study chunking in simplified sequences to examine our cognitive capability to learn structured representation from sequential perpetual data. Formally, the thesis defines sequences as made of discrete elements coming from a set of distinct symbolic items representing all the perceptual possibilities $\mathbb{A}$, which can be related to the unique sensory experience that an agent ventures. An example of such a sequence could be $010021002112000\ldots$. If an agent experiences such sequence on and on, a reflection of the world through this one-dimensional perception, recurring chunks amongst this signal may convey underlying entities in the world. For instance, the agent may perceive some consecutive occurrence of space-time observation as entities, such as learning identities in the sequence: $\{0, 1, 21, 211, 12, 2112\}$, those identities will help the agent to parse the observation sequence one identity at a time. Hence, the model may use the biggest entity that it has learned about to partition the sequence: 0 1 0 0 21 0 0 2112 0 0 0.

This thesis constructs cognitive models based on previous knowledge about people's sequences' learning capabilities. Critically, two components suggested by previous literature are included as a part of all models developed in this thesis.

The first component assumes that the cognitive systems are capable of learning the associations between consecutively identified sequential units [137, 188, 250], rooting back to the behaviorists' proposal associative learning is critical for learning complex behavior. This learning characteristic is supported by a rich set of evidence from statistical and associative learning literature [44, 58, 88, 189, 89]. Examples include artificial grammar learning. Participants learn grammatical strings generated from a finite state language [44] specified by a transition matrix defined on a set of artificial vocabulary. After exposure to the sequence, participants can judge a set of test strings' consistency with the language with above-chance accuracy [58]. Models that learn the associative transition probabilities between the sequential units can reproduce participants' performance in this task [88, 189, 89]. This thesis develops models with components that represent the occurrence frequencies of sequential entities and the transition probabilities between them.

The second component assumes that people process sequence into disjunctive chunks [165, 152], resonating with proposals by Gestalt psychologists' that perceptual

"parts" are perceived together as "wholes". This learning characteristic is supported by the chunk learning literature [196, 238, 62, 91, 170, 165, 28, 115]. Examples include artificial grammar learning experiments, which showed that upon hearing continuous input streams made up of an artificial language containing underlying artificial words, children segmented the language stream into disjunctive recurring parts. This phenomenon can be explained by models that learn recurring disjunctive parts from sequences [172, 197]. This thesis develops models that are inherent in this property of segmenting sequences into disjunctive chunks.

The thesis proposes cognitive models on a computational and algorithmic level. Going beyond seeing chunking and statistical learning as heuristics or tendencies of human behavior, it takes a normative approach and hypothesizes that learning statistics and chunks are rational strategies adapted by a learning agent to discover underlying structures in sequences with embedded hierarchy. Combining the previous knowledge about human statistical learning and chunk learning, this thesis proposes that learning sequence statistics, when combined with learning chunks, become the seed for learning discrete segregated representations from discrete sequential data. In addition to learning cognitive entities, this thesis highlights a close relationship between the cognitive behavior of chunking and the topic of compositionality as a pressing problem puzzling both natural and artificial intelligence.

Specifically, this question is addressed in two parts. The first part studies the emergence of chunks as segregated sequential patterns, and the second part studies the emergence of abstract entities in relation to chunks. For both parts, the questions are approached in conjunction with conducting cognitive experiments while exploring the algorithmic properties of models that exert such properties of computing principles.

The chapters are divided to address the four submitted projects that have resulted from this PhD work:

**The emergence of chunks**

**Chapter 2** starts from conducting behavioral experiments to study how action sequences can be segregated into parts: we manipulated the underlying statistical structure of the sequences and instruction demands in a serial reaction time task. We discovered that humans adapt their behavior to the statistics of the sequence and learn longer chunks are adaptive to the underlying chunk length in the sequence. Meanwhile, instruction focusing on speed versus accuracy also modulates human chunking behavior. We developed a computational model that learns chunks and optimizes a utility function that trades off between speed and accuracy to explain the population behavior observed in this task.

- "Chunking as a rational solution to the speed-accuracy trade-off in a serial reaction time task" (Wu, Éltető, Dasgupta, & Schulz) [246] was published in *Nature Scientific Reports* 13, 7680 (2023), doi:10.1038/s41598-023-31500-3.

The second project (summarized in **Chapter 3**) delves into the algorithmic implications of chunking under a normative lens. It hypothesizes a rational justification for the cognitive tendency to chunk, which is to uncover the underlying hierarchical structure in sequences. It designs a generative model for sequences with a nested hierarchical structure and develops an inverse recognition model HCM (hierarchical chunking model), which recursively reuses the previously learned representations to construct new complex composites. This project demonstrates chunking as a mechanism to discover recurring sequential primitives as entities for sequence factorization subject to compositionality, and also generalizes chunking from a one-dimensional sequential domain to a visual and visual temporal domain and applies the model as a data-driven structural discovery algorithm. This work resulted in the following publication:

- "Learning Structure from the Ground up—Hierarchical Representation Learning by Chunking" (Wu, Elteto, Dasgupta, & Schulz) [245] was published in *Advances in Neural Information Processing Systems* 35, 36706 - 36721 (2022).

The second half of the PhD work studies chunking and its role in learning abstraction in sequences.

It starts with a sequence memory and recall experiment summarized in **Chapter 4**. The project taxonomizes two types of motifs and studies the influence of motifs in memorizing long sequences, in addition to transferring motif knowledge to novel sequences. The sequence recall experiment suggests that people exploit sequential patterns and redundancies not only on a concrete but also on an abstract level. Their learning and transfer behavior can be characterized by a motif learning model that chunks sequences on an abstract motif space.

- "Motif Learning Facilitates Sequence Memorization and Generalization" (Wu, Thalmann, & Schulz) has been submitted as a preprint on *PsyArXiv* [248] and has been accepted in *Nature Communications Psychology*, doi:10.31234/osf.io/2a49z.

The experimental work inspired further modeling of human abstraction learning and its connection with learning by association and chunking. The last project summarized in **Chapter 5** studies chunking in conjunction with abstraction under a normative perspective. A learning agent should learn abstract categories from observations because naturalistic sequences contain distinct objects of the same

category types. Such properties enable the learning agent to learn recurring chunks from sequential data, layer by layer, starting from concrete and building up to more abstract levels. This model relates abstraction with compression and generalization while distinguishing large language models' learning and transfer behavior from human learning.

- "Building, Reusing, and Generalizing Abstract Representations from Concrete Sequences" (Wu, Thalmann, Dayan, Akata, & Schulz) has been submitted as a preprint on *ArXiv* [247] and, at the time of submission, is under review at *International Conference on Learning Representations*, doi:10.48550/arXiv/2410.21332.

Some works are related to the thesis topic developed during the PhD but are not discussed in the thesis. This includes my work in the following two directions:

One is to test the applicability of chunking principles in biologically realistic simulated neural circuits. I worked with Atilla Schreiber and Chenxi Wu from Giacomo Indiveri's group to demonstrate a group of spiking neurons with biologically plausible principles of synaptic plasticity and homeostasis can learn structured patterns from sequences in mixed-signal neuromorphic hardware. The project suggests that biological neural circuits and computation structures can learn chunks in computationally and power-efficient ways.

- "Biologically-plausible hierarchical chunking on mixed-signal neuromorphic hardware" (Schreiber, Wu, Wu, Indiveri, & Schulz) was published in *Machine Learning with New Compute Paradigms* workshop at *Advances in Neural Information Processing Systems* 36 [193].

The other work studies the temporal dynamics of hierarchical visual grouping. I worked with Mehmet Yörüten to propose a psychophysics experiment studying people's recognition behavior upon seeing images tiled by amorphous sub-parts. We showed that the recognition difficulty of image parts can be described by a normalized min-cut algorithm that optimizes grouping by similarity under computational resource constraints.

- "Normalized Cuts Characterize Visual Recognition Difficulty of Amorphous Image Sub-parts" (Wu, Yörüten, Wichmann, & Schulz) was presented at the *Computational and Systems Neuroscience (COSYNE)* conference [244].

Together, this thesis proposes how structured representation in chunks and abstract rules arises from learning from discrete sequences. The means to do so is via

learning association to construct chunks and propose abstractions as cognitive entities, bringing forth computational efficiency and transferability.

# Part II

## From simple to complex

# 2 Chunking in serial reaction time tasks

## 2.1 The emergence of action primitives from sequences

A shared skill across learning, planning, problem-solving, and creativity is the capacity to break down complex tasks into manageable subcomponents, enabling humans to adapt and respond flexibly within complex, high-dimensional, and ever-changing environments [213].

One promising mechanism that underpins this ability is chunking. As a fundamental cognitive process, chunking facilitates the perception and execution of sequences. Lashley proposed that people organize complex actions by segmenting them into smaller, more manageable subsequences [125]. Through chunking, frequently occurring patterns within sequences are identified and grouped into discrete units that can be recognized as cohesive wholes [152, 120, 92, 198]. This chunking phenomenon extends beyond sequence learning, playing a role in domains including grammar acquisition, visual and working memory tasks, function learning, and chess [93, 86, 62, 67, 115, 36].

The ability to recognize and chunk recurring elements in sensory information allows for a compressed representation of long sequences [28]. These chunks can then be repurposed across different contexts, supporting the development of expertise as novices progress by building and recalling pattern-based chunks in long-term memory [160, 36, 84]. Evidence suggests that the foundational units in cognitive hierarchies emerge through the process of learning and organizing chunks in sequences.

In the first study, we conduct cognitive experiments to study chunking in sequence learning. Since movements are underlying the most fundamental aspects of chunk execution, this work started by studying the emergence of units in sequences of movement execution. We want to study how chunks emerge from repeated exposure to simple units, so we chose a sequence learning paradigm. A common paradigm to study sequence learning is the serial reaction time task (SRT) [162, 239, 185, 115]. In SRTs, participants are instructed to press a number of keys that map to the

displayed instruction cues. During the experiment, sequences of instruction cues appear consecutively on the screen; upon the occurrence of each instruction cue, participants react by pressing the corresponding key.

In our experiments, participants were instructed to press four keys, denoted as A, B, C, and D, on the keyboard, which mapped to four instructions that appeared on the screen. If particular patterns, for example, ABC, keep repeating, then grouping repeated chunks as a unit should facilitate the prediction of the upcoming keys. Specifically, detecting a chunk's beginning, in this case, A, implies that the within-chunk items B and C will follow. This anticipation of the following elements of a given chunk can allow participants to anticipate what is coming next and thereby react faster [115, 160].

To study whether participants' chunking behavior adapts to task demands in an SRT task, we manipulated sequence statistics and instructions to participants during the training blocks to examine the behavior change from the baseline to the test block. By default, instruction sequences were generated from a non-deterministic, first-order Markovian transition matrix between the four instruction keys. Out of all 16 transitions specified between the four keys, the transitions from A to B and C to D were highly probable ($P = 0.9$), and the transitions from B to C and from D to A were medium probable ($P = 0.7$). In this way, participants often observed reoccurring sequence segments such as AB and CD and could possibly perceive them as "illusory" chunks, even though the generative model was, as mentioned, nondeterministic first-order Markovian. In practice, the instruction keys were randomly mapped to D, F, J, and K for each individual to randomize the key correspondence to the keyboard placement of the fingers.

## 2.2 Summary of the article

We propose a normative rational chunking model that learns chunks, taking two components into account. One is sequential statistics, i.e., sequences with distinct underlying chunks should result in different learned representations. By merging previously learned chunks, the model finds the best set of chunks to be learned when the entire sequence is considered, thereby segmenting the stream of symbols into compact units of chunks. The model learns patterns as chunks when there are underlying recurring patterns in the sequence. When reaction speed is preferred over accuracy, the model learns longer chunks while tolerating more mistakes. We test these predictions in two experiments, separately manipulating sequence statistics and instructions given.

The first experiment examined how underlying chunks affect sequence learning, controlling for instructions. We gave participants first-order Markovian instruction during the baseline and the test block, sandwiching a training block, which we manipulated. During the training block, we trained three groups of participants separately on sequences containing no chunks (independent), chunk AB (size 2 chunk), or chunk ABC (size 3 chunk), and the elements that do not belong to the chunks occur randomly in the sequence. Consistent with the model's prediction, participants' reaction time data (comparing the test block with the baseline block to evaluate the influence of the training block) suggest that participants learn chunks when underlying chunks are in the sequence.

The second experiment tested the prediction of the influence of instruction on chunking while controlling the sequences in all baseline, training, and test blocks to be the default Markovian sequences. In the training block, one group of participants was instructed and rewarded to perform the task as fast as possible, and the other group was as accurate as possible. The results of this second experiment suggest that the group focusing on speed chunked more than the group focusing on accuracy despite making more mistakes. The analysis result aligns with the model prediction that participants shall adapt their chunking behavior to optimize the trade-off between accuracy and speed.

Our results shed new light on the benefits of chunking under specific task instructions and pave the way for future studies on structural inference in statistical learning domains.

## 2.3 Discussion

Our work can be related to several lines of previous research on chunking. Firstly, Servan-Schreiber and Anderson [197] studied how chunking facilitates memory by examining subjects' memorization for artificially produced grammatical sentences. They proposed that a hierarchy of chunks forms as subjects remember sentences. They instructed subjects to memorize sentences chunked by distinct hierarchy levels (e.g., word level vs. phrase level) and examined subjects' judgment of grammaticality afterwards. Their result suggested that the hierarchy of chunks influenced participants' grammaticality judgments. Additionally, subjects overtly chunked the training sentences even when they were presented in an unstructured manner. These findings are similar to our current model, which also predicts chunking will appear in unstructured data but does so via a trade-off between accuracy and speed. The competitive chunking model provides a modeling framework consistent with our model but does not explain the processes that give rise to chunks' construction and

hierarchy. The mechanism of recombining previously acquired chunks in our model can fill this blank.

Another related model is PARSER. Proposed by Perruchet and Vinter [172]. PARSER can produce artificial language stream segmentations of continuous input streams without any episodic cues such as pauses [189]. PARSER randomly samples the size of the next chunk of syllables and parses the sequence by disjunctive chunks. Each chunk the model learns is associated with a weight, which increments with observational frequency and decrements via a forgetting mechanism. Since both PARSER and our model evaluate chunks based on their occurrence statistics (PARSER approximates the chunk frequency online, whereas the rational chunking model evaluates the joint probability empirically), the simulation results produced by PARSER on syllable parsing can –in theory– be reproduced by our model. Distinct from PARSER and unique to our model is the mechanism of conjoining acquired chunks to construct new chunks and relating the general chunking mechanism to a rational form of utility maximization.

Other methods use neural networks to learn chunks in sequences. Wang et al. trained a self-organized recurrent spiking neural network with spike-timing-dependent plasticity and homeostatic plasticity on sequences like the ones commonly used in SRTs and showed that it could reproduce several sequence learning effects, in particular, transfer effects [233]. It is, however, unclear whether the network learned explicit chunks that enabled this transfer because it is generally difficult to interpret the learned representations of such models. Compared to this approach, our model can serve as an interpretable computational level model because one can directly assess which chunks the model has learned.

Another modeling approach to study how structure emerges from learning is to use variants of the Bayesian ideal observer framework [165, 87, 163]. These models are also rational because their inference is evaluated on the observational instances. The difference between these models and our model mainly lies in the context window and their structural assumptions. For example, with the hierarchical Dirichlet process model [68], the maximal size of the context window, for recognition convenience, is pre-determined to evaluate the prediction of the next element given the previous context. In contrast, our model adapts its context length based on the previously acquired subsequences of chunks. Therefore, we think that these models are very similar to the accuracy part of our rational model and –in the limit– might even make the same predictions for bigrammatic chunks. Apart from that, the rational chunking model accounts for the speed-accuracy trade-off, which is harder to realize and implement in a purely context-dependent Bayesian ideal observer framework relevant to the serial reaction time task.

Finally, relying on the trade-off between speed and accuracy is one way that chunking benefits performance. Other mechanisms have also been proposed, such as minimizing memory or action complexity [81]. Extending our current model to other domains using these additional complexity measures will make scalable predictions in memory, reinforcement learning, and planning.

## 2.4 Limitation and future work

This work has limitations. One is that chunk boundaries are inferred and non-explicit from reaction time speed up. We cannot say for sure when a participant formed a chunk that has been established in mind from analyzing reaction time alone. Future work can integrate the methods we used with other sources of information, such as eye movement data or measurements of brain activities, to cross-validate the estimation of human chunk boundaries. Alternative paradigms, such as asking participants to recall sequences freely, can also help to elucidate the demarcation of chunk boundaries in human behavior.

Our experiments examined learning from sequences with simple underlying chunks; future work may study learning sequences with more complex compositional structures within, which may adapt to participants' learning progress. For example, a model may infer participants' learning progress on the go and introduce novel chunk combinations, i.e., a concatenation of participants' previously learned chunks, up until a point when the participant has shown indicators of sufficient knowledge (presumably using the staircase method [46]) in the two basic chunks to be composed. Such tasks may lead to an adaptive instruction sequence tailored to participants' idiosyncratic learning progress affected by individual tendencies of chunk building, consolidating, and concatenation. Modeling work may examine the influence of different chunk-building parameters on the dynamics of this adaptive learning process. Relating to real-life experience, this process may affect the progression of acquiring composable skills. It may explain phenomena such as the deeper participants are into a book, the faster the reading speed becomes, and the greater the size of a sentence being parsed [183, 13, 30, 192]. In chess-playing, the model can simulate the progressive memory complexity reduction of strategic chess board configurations as a player advances from novice to expert level [84]. The implications of such findings may inform educational curriculum design to adapt to the learning progression of individuals.

Finally, we have examined chunk learning in a simplified experimental setup in this project. Literature has suggested that such simple chunks can greatly benefit the composition of more complex action sequences [195, 194, 222], pointing to the

direction that one primary consequence of chunk learning is to construct simple primitives which can be concatenated to complex composites [212]. In the next project, we further dive into an algorithmic formulation of how simple chunks can recursively merge to create complex composites and a hypothesis on why chunk learning is rational for an agent to build up a better understanding of perceptual sequences via reusing the previously learned representations.

## 2.5 Article Status

"Chunking as a rational solution to the speed-accuracy trade-off in a serial reaction time task" (Wu, Éltető, Dasgupta, & Schulz) [246] was published in *Nature Scientific Reports* 13, 7680 (2023), doi:10.1038/s41598-023-31500-3.

## 2.6 Author Contributions

**Conceptualization:** Shuchen Wu, Noémi Éltető, Ishita Dasgupta, Eric Schulz.
**Formal analysis:** Shuchen Wu, Eric Schulz.
**Software:** Shuchen Wu.
**Visualization:** Shuchen Wu, Noémi Éltető.
**Writing – original draft:** Shuchen Wu, Eric Schulz.
**Writing – review & editing:** Shuchen Wu, Noémi Éltető, Ishita Dasgupta, Eric Schulz.

# 3

# Chunking to compose — a source of infinite combination using finite means

> *Why does each new year seem to pass faster than the one before?*
>
> ...
> *We essentially conduct a lifelong process of chunking — taking small concepts and putting them together into bigger and bigger ones — recursively building up a giant repertoire of concepts in mind.*
>
> **— Douglas Hofstader**

Primary to Hofstadter's speculation is a fundamental feature of chunking, which combines simpler components into more extensive and complex components that explain ever-larger recurring experiences in life. In the former project, the experiment on serial reaction time tasks suggests that humans adapt chunking behavior to the underlying regularities in the sequences. In this subsequent paper, I investigate the relation between chunking with hierarchical sequence structure, compositionality, and factorization.

Cognitive scientists have suggested the vital role of chunking as a way to circumvent our inherent mental limitations. About half a century ago, Miller reported that our short-term memory is limited to holding 4 to 7 *chunks* [152]. Once a chunk has been learned, it is memorized, identified, and parsed as a whole [62, 125, 227]. This discovery has led to ample subsequent work suggesting chunks as the primary information processing unit and serving a role in decomposing complex sequences into familiar parts. To illustrate, consider remembering a sequence like "schwarzwälderkirschtorte" — a challenging task on its own — a sequence with 26 items. However, knowing that it is a concatenation of the German words "Schwarzwälder" (Black Forest) and "Kirschtorte" (cherry cake) simplifies the task, as the sequence is decomposed into several familiar parts. Recognizing familiar subsequences aids in remembering more complex sequences.

The process of breaking perception into several entities goes beyond sequences to the visual domain: Gestalt psychologists have observed and developed the notion of 'Prägnanz': upon processing a complex and chaotic visual scene, people tend to organize and group visual perceptual units together into coherent wholes [236, 237]. A primary tendency to group perceptual units as a whole is by proximity: entities close to each other tend to be perceived together to form a group [236, 237]. Interestingly, the grouping by proximity in vision resembles the grouping by chunks in sequences, suggesting the chunking principle as a candidate to break complex observation into parts in both visual and sequential domains.

If chunking can be a candidate of a cognitive principle that underlies many domains, what could be a normative reason to justify the rationality of learning chunks? From a computational standpoint, upon processing streams of perceptual sequences, a learning agent faces an inherent challenge in learning structure from the sequence for better predictability, memorization, and recall upon task demand. From this point of view, chunking implies several attractive algorithmic features.

The first is that chunks can serve as computational processing entities to explain the emergence of symbols. This points to a potential answer to the unresolved problem in symbolic AI. Symbolic AI systems study the consequence of intelligent behavior via operating with symbols but do not address where symbols come from (sometimes, they resort to some innate explanation that circumvents this problem). Because symbolic AI relied on this fundamental assumption, their approach suffered difficulty in scaling up to higher dimensions, as going into any more complex data domains will reveal the problem of finding primitive symbols to parse the data reasonably. The formation of chunks through learning offers a potential answer to how symbols and distinct entities emerge from experience. By enabling us to segment observations into discrete components, chunks serve as foundational units that transform input sequences into recognizable parts.

The second is that chunks can become independent entities to factorize a probabilistic estimation of the observational sequence. An observational sequence with $n$ entities can be described as distributed in a high dimensional probability distribution $P(x_1, x_2, \ldots, x_n)$. Chunks become a unit of sequence parsing, and thereby, a sequence spanning in many observational units can be partitioned by grouping observational units as chunks, each chunk occurring independently from the occurrence of other chunks, and therefore factorizing the observational sequence distribution by chunks of consecutively occurring stimuli: $P(x_1, x_2)P(x_3, x_4)P(\ldots, x_n)$. This suggests that the mechanism of learning chunks may help computer scientists find ways of circumventing the computational complexity of factorizing high-dimensional distribution of observational sequences and can also serve as a way of learning a generative model of the sequences.

What follows as the third attractive algorithmic property is compositionality: the previously learned chunks can be composed into more complex chunk combinations. Previous work using symbolic systems to study human behavior has suggested that the composition of primitive operations explains both human behaviors of learning a hierarchical organization of the primitives [123], which helps people to make fast structure learning and generalization [195]. Additionally, the components that are composed may contribute to the processes of generalization and transfer between separately learned sequences [160]. Chunk learning and chunk composition may also lead to chunk transfer.

## 3.1 Related Work

Several approaches to model chunk learning from sequences exist in the literature. One is a process type of cognitive model including PARSER [172], CCN [198], and others [186]. These models use heuristics to illustrate how chunks can arise from data, usually from jointly frequently occurring items that contain associative relationships. While PARSER compared simulated chunking with baby sequence segmentation when learning an artificial stream of language, CCN related the process of chunking with compositionality by organizing chunks in a hierarchical way. These process-level models are limited in their heuristics and lack a normative account of why chunking can be a rational behavior for the learning agents.

In contrast to the process models, normative statistical models describe ideal observers' behavior to explain why chunking is rational, usually using variants of the Bayesian ideal observer framework [87]. For example, given a linguistic corpus, these models infer a segmentation with the highest probability from a set of chunks following the minimal description length principle. These models are rational as the inference is evaluated on observational instances. However, the inference process of these models suffers from combinatorial explosion with increasing sequence length. Presumably, the normative chunking models do not relate chunking with compositionality because of the computing complexity.

Other chunk learning models are connectionist in nature. Approaches include using an artificial neural network to learn sequence segmentation or simulating spiking neural networks that are similar to our biological neural construct and translating the chunk learning problem into a loss function for network parameter optimization [75, 45, 233]. Usually, these models generate behavioral consequences of chunk learning, but they are opaque to interpret due to their connectionist setup.

This project implements computational cognitive models that connect chunk learning in cognitive science with the algorithmic advantages it provides. It also proposes a normative explanation, hypothesizing that chunking is a rational strategy. Building on previous experiments, which show that humans learn underlying chunks in sequences, this project suggests that humans behave like rational agents, uncovering patterns in their observations. Thus, chunking may serve as a rational method for identifying underlying relationships and regularities within observed sequences.

## 3.2 Summary of the work

To study the normative explanation of the chunking mechanism's rationality while exploring the algorithmic advantage that such a mechanism brings forth, the project started by investigating the relation between hierarchical structure and chunking. To preciously control the structure of the sequence, we design a generative model that randomly comes up with an underlying nested hierarchical structure. The generative model starts with an inventory of initialized unique chunks of atomic units. In a number of iterations, existing chunks in the inventory randomly concatenate together to form chunk composites. Each chunk in the inventory is assigned an independent occurrence probability. The sequence is generated by consecutively sampling chunks from the generative model. This approach to developing a generative model captures the essence of compositionality at the sequence level: simpler concatenated chunks form foundational units that recur and combine to create increasingly complex structures within the hierarchy.

Observing such non-iid sequences with recurring chunks sampled from the hierarchical generative model, this project proposes that chunking can become a means to uncover the underlying structures in sequences. It proposes a simple chunk learning model that contains three components that can be plausibly implemented by cognition.

The first component is parsing, i.e., the model parses and identifies chunks together as a unit from the sequences [161, 86, 171, 27]. The second component is learning the associative statistics of consecutively parsed chunks, a notion that inherits the legacy of behaviorists' proposal (that animals learn to associate between events) while also being affirmed by the statistical learning literature (human learners are sensitive to the transition statics and the occurrence probability between consecutively observed entities in sequences [88, 189, 89, 188]). Meanwhile, a forgetting component multiplies the count of parsed chunks by a discounting factor, a common practice that models forgetting [61, 159, 158, 180, 184, 7].

Connecting the three components, this paper proposes the hierarchical chunking model, which builds up a nested hierarchical structure from sequences. HCM starts out learning about the minimally sufficient atomic sequential units as initial chunks to parse the sequence and combines chunks which has a correlated consecutive occurrence as indicated by the information provided by the associative statistics into more complex chunks to add to the dictionary. The simple merging process allows the model to learn more complex representations by reusing the previously learned chunks. In this way, a long and complex sequence can be learned as one entity in the dictionary by reusing and concatenating the existing chunks in the memory dictionary. The model learns a dynamical graph that is a trace of the evolving representation in the dictionary.

HCM brings the feature of compositionality in the sequence learning domain. As the previously learned chunks are used as basic components to parse the sequence and as candidates to compose into new chunks. The compositionality process contains a normative aspect guided by the recorded sequence parsing statistics, reducing the space of compositionality to those chunks that contain a correlated consecutive occurrence relationship and thereby also circumventing the vast search space as encountered by alternative formulations. Apart from that, the chunks become entities to factorize the high dimensional sequence distribution. One can evaluate the probability of a sequence $S\ (x_1, x_2, \ldots, x_n)$ occurring $P(S)\ P(x_1, x_2, \ldots, x_n)$ by the probability of chunks that constitute the sequence parsed by the model: $P(x_1, x_2)P(x_3, x_4)P(\ldots, x_n)$ (each group is the elements inside a chunk). Alternatively, the resulting representation can also be used as a generative model to produce imaginary sequences composed of sampling the learned chunks adhering to their occurrence probability by the model.

HCM is formulated as a normative chunk discovery algorithm, i.e., chunking is rational for the model to uncover the underlying nested hierarchical structure in the sequence. This project includes learning guarantees of a rational HCM on an idealized generative model and demonstrated its convergence. Apart from formulating chunking as a normative representation discovery process, the project also shows several learning advantages this algorithm affords.

The first one is data efficiency. On sequences with embedded hierarchies produced by the generative model, I compared HCM with RNNs learning from the same amount of sequential data. Given the same length of sequences, HCM could adaptively build its nested hierarchical representation by detecting correlation violations until no correlation can be detected. In contrast, neural networks are much slower at adapting their representation. Given the same amount of training data, HCM learned a better representation of the sequence than an RNN. We also observed that the

advantage in HCM's data efficiency becomes more pronounced as the hierarchy depth of the generative model increases.

As HCM learns interpretable chunks, we also looked at the implication of transfer when the model adapts its previously learned representation to novel sequences generated by alternative hierarchical structures. The model's interpretability informs positive/negative transfer to learn representations in a novel environment where sequences come from a generative model with overlapping/complementary chunks. Since the previously learned chunks can be reused, the transparency of all existing chunks acquired by the model shows whether the new chunks need to be learned additionally. With full knowledge of the transfer sequences in relation to the learned representation from the model, positive or negative transfer can be reliably predicted.

Many natural sequences may contain a hierarchical component similar to the generative model; as a testing ground, HCM was applied to learn structures from sequences from the book *The Hunger Games*. From text sequences, HCM learns nested hierarchically embedded chunks reflecting the hierarchical organization structure of language. This includes the step-wise emergence of word parts such as common prefixes and suffixes in a word, commonly used verbs, and nouns, and later more complex phrases also emerge, including phrases such as "it is not just", "in the school", "our district," and "cause of the", similar to how we parse sentences through successive units of words and phrases when reading instead of letter-by-letter [166].

This project also delves into the algorithmic consequence of chunking as a representation discovery mechanism in discovering interpretable compositional relationships from and beyond one-dimensional sequences, and into higher-dimensional visual and visual temporal sequences. I extended HCM to learn visual temporal chunks via proximal grouping and demonstrated that the model could learn frequently occurring visual-temporal parts to aid in breaking down a complex sequence of images into chunks of visual temporal wholes via combining their corresponding parts. Consequently, the complexity of the visual-temporal sequence reduces as learning progresses, and the model learns recurring visual-temporal movements. The model's behavior suggests that chunk learning can also capture the correlation in both spatial and temporal dimensions, hence may explain cognitive phenomenon beyond one-dimensional sequences to higher dimensions such as visual or proprioceptive sequences [54].

The ability to discover recurring temporal-spatial patterns and their sub-recurring patterns as chunks organized in a nested hierarchical graph makes HCM a method for extracting patterns in an unsupervised manner. One candidate data type hypothesized to contain a hierarchical structure is neural activities [6]. We demonstrated

that HCM can be used to learn recurring activations of functional brain regions on a resting-state fMRI data set. The interpretability of chunks allows matching the occurrence of chunks and stimulus onset to be compared with the known network connectivities in the brain. Via this method, we found nested network structures such as functional regions responsible for affect processing [209, 232], visual attention control [228], or theory of mind processes [15]. On a population level, we also observed a correlation between the average chunk size per participant and age - implying the aging brain possesses a more modularized activity signature.

Together, we propose a model that processes a stream of perceptual sequence into chunks. HCM learns chunks as the basic unit of cognitive processing, allowing for the composition of chunks, factorization of sequences, and transferability to novel sequences. We demonstrate the application of this model to discover recurring patterns in an unsupervised manner, from one-dimensional sequences to multiple-dimensional visual-temporal sequences. This work suggests that chunking is a universal computational principle used to acquire parsable entities from sequences, resonating with previous discoveries in fields from sequence learning and memory to gestalt psychology and the arrival of visual entities.

## 3.3 Discussion

The algorithmic design of HCM favors simplicity as a proof of concept to demonstrate the power of chunking. This simplification comes with limitations. One is the greediness in parsing; the model finds the biggest chunk in the learned dictionary in its volume to parse the sequence despite their low occurrence frequency as a heuristic. This choice may hinder the model from discovering the most plausible underlying chunks in the sequence, but rather in favor of expanding the inventory of its dictionary. Depending on the application, the future may extend the parsing process also to consider a model that infers the observation of chunk online [165], thereby making the algorithm adhere to the updated probability of chunk parsing probability.

Many pattern detection algorithms, including deep learning approaches, are not robust to independent noise in data. This poses big application problems and does not relate to the noisy biological substrate of the neural system, and hence is also difficult to implement in hardware systems that are also prone to noise or computation corruption [90, 127, 98].A feature of HCM is that chunk discovery is little affected by independent noise in sequences or when the occurrence instance of some underlying chunk is only partially observable by occasional signal corruption. This is because independent noise does not affect the statistical correlation among

consecutively parsed chunks and, hence, does not influence the process of chunk proposal. What noise affects is the parsing process, i.e., smaller chunks consistent with the noise-corrupted sequences will be chosen as candidates for sequence parsing. As noise robustness is not the focus of this project, future work may exploit this feature advantage of the algorithm by simply making the parsing process more noise robust via introducing a similarity comparison mechanism matching the previously learned chunks and the sequence observed or adapting to some probabilistic variant, such as computing the chunk that leads to the maximal a posteri given a noisy observation sequence.

Another limitation of this work is the computational efficiency in chunk searching. During each parsing step, the number of search steps to identify the biggest chunk that matches the sequence scales with the inventory size. Future work can improve the efficiency of this parsing step by either organizing chunks in a prefix tree-like structure to shorten the search step to scale with the depth of the prefix tree. Alternatively, chunk parsing can be implemented in parallel computing systems that contain independent components checking the consistency of individual chunks in the inventory simultaneously, with the biggest chunk matching in size winning the competition. Even better will be the combination of both types.

One innovation of this work is to relate chunking to compositionality in sequence learning, suggesting that chunking can be a means to compose the more complex from the simple in any problem domain that can be formulated as symbolic sequences. This formulation informs and differentiates from other approaches to model compositionality.

The most prominent models that explain human compositional behavior are program induction models. They have been mostly applied to behavior domains with partial observability, such as explaining the composition of handwritten digits or coming up with programming steps that generate the output of a transformation from one base word form to its morphological variant [121, 66, 65].

Program induction models capture humans' capability to compose more complex cognitive representations from simpler ones by formulating mental computation as a combination of programs. Problem-solving involves searching for the combination of programs with the highest probability of explaining the observed data and updating its posterior distribution over the programs with more observation instances.

However, two problems hinder the program induction approach from being integrated to explain more human behavior or data domains with higher dimensionality. The first problem is that these models are domain-specific and assume knowing a set of computational primitives. However, these mental primitives are not known in

most domains. Hence, such models have been limited to simple domains that are feasible for experts to hypothesize computational primitives, including hand-digit writing [121], geometric hand-drawing [66], or simple programming [187]. The other problem is the vast program search space. Finding the right program combination that takes an input and produces an output is an exhaustive process and subject to combinatorial explosion. Improvements to program induction methods include using neural networks to guide the search process [66], but only partially alleviates this issue.

By formulating learning in the sequential domain, our model of learning to compose entities to parse sequences captures the essence of compositionality while resolving both limitations of program induction approaches. Since the model starts learning with an empty inventory, primitives can be learned without the necessity to impose a library, and the model does not differentiate between the basic primitives or the frequently used composite chunks. Studying chunking in sequence learning also circumvents the huge computational complexity of finding combinations of primitive functions that explain an output observation, as combinations are acquired via observing sequences.

While not every human behavioral aspect can be framed as inducing programs, most human behavior can be described by sequences of action and perception. Chunking, as a proposal for the emergence and reuse of subsequences as cognitive units, offers an explanation for a part of behavior in all domains that contain a sequential component. In future work, this approach can be integrated with the program induction methods by applying chunk learning to the sequences of program execution traces or other data domains where composing programs plays a role.

Future work may also relate this model with cognitive experiments to test the implication of chunking during learning. If chunking as a tendency helps learning and reuse to acquire complex structures, then the curriculum should affect the learned representation critically. In particular, training needs to include enough repetition sequences to allow sufficient time for the learning agent to pick up the invariant patterns and rules. Indeed, studies have suggested that training regimes that fix invariant rules (blocked training) facilitate humans in learning about the underlying rules compared to regimes where the rule keeps changing (interleaved training) [73] - a feature that also distinguishes humans from artificial neural networks. Future studies can investigate the contribution of chunking to the curriculum effect observed in these experiments.

Apart from the curriculum effect, the correlation detection feature of HCM can be applied to propose variables and generate hypothetical causal graphical models from sequential observations, subject to further model selections or interventions.

Furthermore, applying chunking principles in the visual domain could explain the diverse Gestalt grouping laws. For instance, the tendency to group by similarity or common fate may be due to the correlated occurrence of similar objects in the visual field. Future studies may apply chunking principles to natural video data to test the hypothesis that some gestalt rules reflect the correlation of recurring patterns in natural image sequences. This could lead to the understanding that particular grouping principles result from an agent's identification of familiar visual relationships to reduce the perceptual complexity of observations, which is a rational strategy to break complex perception into parts.

## 3.4 Article Status

"Learning Structure from the Ground up—Hierarchical Representation Learning by Chunking" (Wu, Elteto, Dasgupta, & Schulz) [245] was published in *Advances in Neural Information Processing Systems* 35, 36706 - 36721 (2022).

## 3.5 Author Contributions

**Conceptualization:** Shuchen Wu.
**Experiments:** Shuchen Wu, Ishita Dasgupta, Eric Schulz.
**Software:** Shuchen Wu.
**Visualization:** Shuchen Wu, Noémi Éltető.
**Writing – original draft:** Shuchen Wu, Eric Schulz.
**Writing – review & editing:** Shuchen Wu, Noémi Éltető, Ishita Dasgupta, Eric Schulz.

# Part III

## From concrete to abstract

# 4

# Beyond concrete sequences — Formulating and testing two types of motif learning in sequence learning and transfer

In the past chapters, we have investigated chunking experimentally and theoretically. From sequence learning experiments, we observed that humans adapt their chunk learning strategy to the underlying recurring chunks in sequences, suggesting that a normative pattern discovery principle can explain the chunk learning process. We delved into this question further in the second project, looking at sequences that contain a nested hierarchical structure. We proposed a model that adheres to the cognitive ability of humans to uncover chunks as entities of recurring patterns from sequences with a nested hierarchical structure. The model learns interpretable chunks that are also transferrable to novel sequences.

Chunking has limitations in capturing the structural learning abilities of humans. It is a fascinating aspect of learning and cognition that we not only learn recurring patterns in their concrete form, but we are equally good at dismissing irrelevant details to learn abstract recurring patterns. Psychologists and neuroscientists related abstracting sensory experiences into concepts as preconditions of forming episodic memory during development. Concepts can be bound with context and chained into a memory of episodes, facilitating a recollection of concept sequences surrounding a personal event [82, 224]. Abstract concepts and categories become the seed of the thinking process, not only helpful for our memory but also helpful with logical induction and deductions of conclusions that lie outside of our personal experience [9]. This chapter studies patterns in their abstract forms and their relation to transfer and generalization.

Many daily examples suggest our instinctive attraction to motifs from sequences. Music, for instance, contains abundant melodic motifs invariant amongst varying note specifications. Beethoven's Fifth immediately comes to mind when the iconic sequences of notes strike: GGGE, FFFD. Within the symphony, the note sequence progresses to GGGB or GGGC, with variations in forms and voices, one at each step. The two examples point to two abstract sequence motif types that people are

sensitive to, as the literature suggests. The first type relates to the initial definition of Gestalt ('form' in German) back in 1890: Von Ehrenfelds observed that a melody is recognized when played with different keys [63]. Later, this form of motif sensitivity was also suggested by the language learning literature. [139] exposed seven-month-old infants to short sequences with simple grammar patterns such as ABA, QWQ, and EFE. This exposure made the infants sensitive to non-explicit sequence patterns, leading them to direct their gaze toward novel sequences sharing the same structure, such as KTK, rather than different structures, such as DDF. However, studies on these motifs usually look at the effect of motifs in very short sequences and do not describe how the abstract motifs may build up from the learning process.

The second motif type, expecting that the progression of the music will vary on a position following the iconic three strikes GGG, relates closely to the linguist's hypothesis on the acquisition of language, specifically grammar structure acquisition [25, 142]. This type of motif is speculative to exist as a precondition of grammar structure acquisition. It also relates to Chomsky's hypothesis that statistical learning between words cannot explain the infinity of language utterances — a symbolic acquisition of language structure is necessary for people to judge the grammaticality of unseen sentences [40, 141, 250]. Learning abstract patterns at the symbolic level, such as the category of a noun, allows learning the abstract pattern of grammar, such as noun phrases typically consisting of a determiner followed by a noun. Developmental linguists suggest that children, after exposure to their native language, learn about the abstract category of nouns and verbs and are capable of applying their knowledge about nouns to novel phrases that have been seen to belong to the noun category. Sometimes, they overgeneralize the syntactical structure of nouns that demand an exceptional case [142]. This capability also demands the acquisition of language structure at the symbolic and abstract levels. However, such types of learning abstract structures in sequences have not been examined in an experimental setting.

In this work, we expand on the previous literature with new additions. As the first kind of motif has only been tested in short sequences, we want to examine how people learn about these motifs in long sequences that are cognitively challenging. We look at the progress of how people acquire sequence motifs and how the two types of motifs affect the learning and memorization of novel sequences outside of participants' training experience.

To do that, we first defined the two types of motifs in a sequence learning setting, especially the second type, as it has primarily been discussed in the language learning setting. We then formalize these motifs in a sequence learning setting to help study such motifs in a domain-general way. Following our definition, we conduct a sequence memory and recall experiment to test the effect of motif learning and

motif transfer when people have been exposed to the two motif types and their ability to transfer to novel sequences that share the same motif type. The cognitively demanding task of remembering long sequences necessitates gradually building knowledge of the motif during the learning period.

Although literature have suggested that human cognitive capability is sensitive to abstract motifs in sequences, there have been no explanations for an underlying reason why people should learn motifs from sequences. Human motif learning has been discussed on an observational level, lacking a normative account. The modeling work we propose provides such a normative account: we suggest that learning abstract motifs can be closely related to learning chunks, and the process can be described by chunking on an abstract motif level. In this way, learning motifs can be explained by participants finding invariant structures in sequences for efficient compression. We build a model that integrates abstraction learning and chunking into the same program to discover sequence motifs manifested as abstract chunks in sequences.

## 4.1 Toward a Taxonomy of Abstract Motifs

We define two types of motifs: projectional motifs and variable motifs.

A projectional motif is a pattern in a projected space shared among distinct sequences. A transformation function maps the superficial content to this projected space. For example, GGGE and FFFD's music phrases share the projectional motif XXXY.

A variable motif is a pattern with invariant and variant parts. In a sequence with a variable motif, a variable symbol represents a quantity that can change. These sequences share a structure with a varying entity at the "X" position and constant entities elsewhere. For instance, the music phrases GGGEZ, GGGB, and GGGC share a variable motif GGGX, with X taking the value of Z, or B, or C.

## 4.2 Summary of the article

This article explores how abstraction aids in memorizing sequences and transferring abstract knowledge from one sequence to another in recall experiments.

We tested this hypothesis in two serial recall experiments: participants were instructed to memorize 12 consecutively displayed colors and then recall the sequence

by pressing corresponding keys, with recall accuracy recorded as the primary measure for analysis.

Experiment 1 examined how projectional motifs aid memorization and transfer. Sequences consisted of two variables, X and Y, each appearing 6 times per sequence presentation. Participants were divided into two motif groups (Motif 1; Motif 2) and a control group (Independent). A motif was consistent across training trials in the motif groups, while in the Independent group, X and Y were permuted in each trial. X and Y were mapped to distinct colors. Participants underwent 40 training trials followed by three transfer blocks, each consisting of 8 trials, testing motifs of each type. In the transfer blocks, sequence colors from training did not reappear. Experiment 2 tested the learning and transfer of variable motifs. Participants were divided into a variable motif group (motif) and a fixed group (control). The variable motif group memorized sequences like B X D F, with X varying (A, C, E), while the fixed group memorized sequences like B A D F, with no variation. Participants underwent 40 training trials followed by 24 transfer trials. In the test block, both groups memorized new sequences with variable X in the same positions as training but with changed fixed parts. Analysis of the human recall accuracy data suggested that participants effectively learned and transferred both motif types. Training with variables and projectional motifs improved recall accuracy, especially on transfer sequences.

We propose a model that differs one step from the hierarchical chunking model, integrating learning transition statistics, learning chunks, and pattern discovery on a motif level. To simulate sequence motif learning of the first type, we simulated the model by learning chunks in the abstract projectional motif space. For the motif of the second type, we integrated a component that proposes an abstract variable entity based on preadjacency and postadjaency transition statistics between the parsed chunks, thereby discovering recurring chunks in sequences that contain variables. Together, such a model simulates a progressive build-up of sequence motifs via discovering recurring patterns in the motif space and progressively concatenating the previously learned abstract chunks into bigger chunks, reusing the knowledge of sequence motifs to novel sequences. Simulation of the model in the two experiments learning identical sequential instructions to the participants suggested that the motif learning models progressively learn motif chunks, which help the model to transfer and generalize. The model sequence generation accuracy correlates with participants' sequence recall accuracy during the progression of the experiment. A detailed model comparison separately, including each component that consists of the motif learning model, suggested that motif learning and transfer cannot be explained by chunk learning or associative learning alone. Expanding chunking from concrete sequences to abstract representations was crucial for capturing the learning and transfer effects in this set of experiments.

Our findings suggest that human participants use both motifs to facilitate sequence memorization and generalization to novel, unseen sequences. This learning and transferring process in this experiment can be captured by chunk learning in the two types of abstract motif space. Discovering recurring patterns in sequences helps people memorize and transfer sequences with abstract motifs. Our work paves the way for a better understanding of how abstract motifs emerge progressively from sequences and their implications in generalization.

## 4.3 Discussion

It is a fascinating aspect of learning and cognition that we not only learn recurring patterns in their concrete form, but we are equally good at dismissing irrelevant details to learn abstract patterns. Our work hypothesized two sequence motif types that humans could learn and generalize, tested these hypotheses in sequence memorization and recall experiments, and proposed a model that progressively builds up a complete sequence motif via chunking in abstract space. Our work advances our understanding of how people construct abstract representations from observational sequences for efficient compression and generalization.

This work paves the way for future work to expand into the characteristics of learning abstractions in sequences. Our experiment tested abstract motif learning in a restricted number of sequence types: projection motif that spans the entire sequence and variable motifs that contain one variable at a specific ordinal position of the sequence; future work may expand upon the variability of this experimental paradigm and design experiments to study and test more flexible motif learning. For example, one can have, in the experimental sequence, multiple variable entities, X and Y, each having distinct entailment, located at different positions of the sequence, and look at how the learning of a sequence that contains Xs and Ys, helps to transfer to novel sequences, where the location of Xs and Ys may also swap.

Additionally, future work may test hierarchies that span multiple abstraction layers, such as projectional motifs embedded in variable motifs, i.e., variables that represent several possible projectional motifs or vice versa, and how learning adapts to various motifs. Modeling-wise, this may correspond to the discovery of a abstract structure that helps a learning agent compress sequences based on the previously learned motifs. Future work can test the interaction between learning this motif structure and participants' performance in transfer and the individual variabilities in their sensitivities to either motif type in sequences.

We studied motif learning in the sequence learning domain; future work may relate this work further to the general domain of learning abstractions. Many of these works on the role of abstraction and generalization formulate their problem in a non-sequential domain. For example, previous work on abstraction has studied our tendency to understand abstract concepts via metaphor, such as understanding the concept of an 'argument' in terms of 'war' and thereby transferring the feature of war to the concept of 'argument' [126, 124]. Another example is in problem-solving, where people tend to find a solution to a new problem based on their knowledge of a familiar problem that resembles the new problem in abstract ways [59]. Additionally, acquiring reasoning rules from experience has been proposed to build the foundation for logical deduction and reasoning [37]. Many of these works argue that finding commonalities among conceptual space manifested as abstract rules is fundamental to human intelligence. The property of abstraction has also helped to advance multiple fields. In math, abstraction empowers mapping deducted theorems from one axiomatic system to another [154]. In computer science, abstracting computing steps into functions and classes allows the reduction of the computational complexity of programs [3].

Future work may connect the sequential aspect of this model with the progressive acquisition of concept relational graphs or show how abstractions described by this abstraction literature can arrive from perceiving data sequentially. In particular, it can be interesting to adapt the model to describe a process of how abstraction structure can be built up progressively from learning: for example, the model can describe the process of realizing a solution to a simple problem via learning sequences that underlie the program traces of the search steps. In the meantime, arriving at sequences of simple abstract execution steps may help participants learn to piece together knowledge in the abstract space to reach higher-order abstraction manipulation. Models of such flavor can also be used to simulate and measure the learning difficulty of abstract problems, or the learning and developmental stages necessary as a precondition to understanding abstract concepts or transferring metaphorical understandings. Generally, the model may illuminate how simple mechanisms of chunking in an abstract space may help cognition find a common pattern beyond seemingly distinct observations and build up layers of cognitive sophistication to construct and extrapolate concepts outside our finite sequential experience.

## 4.4 Article Status

"Motif Learning Facilitates Sequence Memorization and Generalization" (Wu, Thalmann, & Schulz) has been submitted as a preprint on *PsyArXiv* [248] and has been accepted in *Nature Communications Psychology* doi:10.31234/osf.io/2a49z.

## 4.5 Author Contributions

**Conceptualization:** Shuchen Wu, Mirko Thalmann, Eric Schulz.
**Experiments:** Shuchen Wu, Mirko Thalmann, Eric Schulz.
**Software:** Shuchen Wu.
**Analysis:** Shuchen Wu, Mirko Thalmann.
**Writing – original draft:** Shuchen Wu.
**Writing – review & editing:** Shuchen Wu, Mirko Thalmann, Eric Schulz.

# 5

# The construction from the simply abstract to the complexly abstract, layer by layer

> *Within great truth lies great simplicity.*
>
> **— Lao Tzsu**
> Tao Te Ching

In the sequence recall experiments, we verified firsthand that two types of motifs help people memorize and transfer their knowledge to novel sequences. Our experiments suggested that people learn patterns not only on the sequence surface level but also on an abstract level. In the following work, I delve further into how chunking on an abstract level may help uncover layers of abstraction in data and how the mechanism of such a model relates to abstraction in general.

The ability to form task-specific abstract representations has been suggested to be fundamental for our intelligence [117, 16, 173, 49]. People are equipped with the ability to abstract. As we learn a new language, we also learn the salient patterns underlying grammatical forms without explicitly being told about the rules. For example, after learning German for a while, you will expect a verb at the end of a subordinate clause. This verb can mean "kick", "support," "drink,"... etc. But you develop a sensitivity to the functionality of the last word. Children acquire grammar structure when learning a language; they learn the rules, such as determiners precede nouns, and generalize the rules [50]. Infants as early as 23 months old can learn the category of nouns [221, 21, 147], expect the syntactic category of the next word in a sentence, and use their knowledge about nouns in argumentative roles that they have not experienced in the past. Denoting unknown entities in symbolic abstract form was fundamental to the development of mathematics. It is easier to arrive at a solution of an algebraic equation such as "x + 5 = 10" by assuming 'x' as a symbolic, unknown entity to find out about "x = 5". A proper abstract description may help an agent discover the underlying relation that governs the otherwise highly complex and variable observations. Consider Newton's law in physics or Maxwell's equation describing electromagnetic waves: abstracting unknown entities in symbolic forms has been civilization's workforce to discover the invariant laws in nature.

## 5.1 Related work on modeling abstractions

Knowing the theoretical and pragmatic implications of studying abstraction, researchers have attempted to model abstraction learning using different approaches.

One approach to model abstraction builds explicit discrete conceptual relational systems manifested in graphical structures. This sort of model has been applied to explain human behavior, including understanding abstract concepts via linguistic metaphor [126, 124]: such as grounding the concept of an 'argument' in terms of the definition of 'war' and thereby transferring the characteristics of war to the understanding of 'argument'. Models with this flavor have also been applied to explain people's transfer behavior in problem-solving, how solving a new problem becomes much easier when knowing the solution to a familiar problem that resembles the new problem on an abstract level [59]. These models usually represent knowledge or conceptual understanding in discrete forms, manifested in conceptual relational networks in which nodes are ideas or concepts, and edges denote the relation between the ideas. Modern adaptations of such approaches include the Probabilistic Analogical Mapping (PAM) model, which uses word embeddings created by neural network systems like BART (which maps concepts to vector embedding) to construct such a conceptual relational network [107]. The task of finding abstraction amongst the source and target analogical concepts or using the solution of a previous problem to solve a new problem can be translated into finding and applying graphical commonalities between the two discrete graphical structures [132, 190]. A limitation of this approach is that these models assume a conceptual relational structure or acquire them from connectionist systems and do not explain how learners build up the discrete conceptual relational graph from experience. In a similar vein, Kemp and Tenenbaum [213] use a Hierarchical Bayesian model defined over a set of graph primitives and grammars to combine the primitive to illustrate how complex graphical structures can be acquired by combining simpler ones. However, the model relies on assuming a library of primitive abstraction relations and does not explain how the abstraction primitives may arise from data. Hence, these explicit discrete abstraction models have been primarily applied to restricted problems or data domains.

Another approach to model abstraction circumvents the problem of finding an explicit representation via training connectionist neural networks through a variety of datasets that demand abstraction learning and transfer. Such approaches include meta-learning [70, 205, 95]: training a neural network on a distribution of tasks that are in different domains but share some underlying properties [22, 104]. In these cases, neural networks can do one-shot or few-shot learning in novel tasks that share the underlying property with similar tasks in training data. These connectionist models contain implicit abstract representations in the learned weights [231, 251],

which are opaque to interpret what explicit rules or commonalities may have arisen from learning. Hence, understanding and interpreting these models is an area under active research [251, 56, 60, 57]. Without knowing the explicit abstractions acquired by neural networks, it is even harder to compare with the type of abstraction and transfer ability that humans are using. Therefore, datasets or tasks that demand models to exert human-like abstractions and reasoning abilities are still challenging for the best connectionist models today [38, 155, 140].

## 5.2 The open question

When studying the growth of abstraction from perceptual data in the cognitive system, it is crucial to develop computational principles that yield interpretable structured representations. These principles should be capable of learning structure from experience while maintaining interpretability. To achieve this, we can draw insights from the literature on developmental psychology.

## 5.3 How abstract concepts may grow inside the mind

Literature suggests that abstract conceptual symbols originate in perceptual experience and arise from superficial sensory experiences. Indeed, evidence suggests a close relation between neural activities representing concepts and neural activities representing experiences. There are no specific neural substrates dedicated only to representing abstract concepts. Instead, during sensory-motor perceptual experiences, association areas in the brain capture bottom-up activation patterns in sensory-motor regions. Later, perceptual symbols activate the association areas, which in turn reactivate sensory-motor areas [17, 83, 47].

The developmental literature suggests several key features of abstraction. The first feature abstraction is **commonality**. Classical conceptions suggest that abstraction arises from generalizing common features of experience. For example, the abstract concept of 'swans are white' arises after observing many instances of white swans [253].

The second feature is **discretization**: a continuous information stream is divided into concrete, recurring, and symbolic units. There is an end, a beginning, and a range of values that an abstract concept assumes, such as the abstract concept of a noun includes cat, dog, box, and other words that belong to the noun category

[164]. The abstract categories that point to the words are articulated when parsing a string, such as *The cat jumped out of the box* to check the grammatical validity.

The third feature of abstraction is **information reduction**, i.e., throwing away information. An abstract concept is less specific than the concrete concept that it entails. The word *cat* is more specific than the category *noun*. Throwing away information has been suggested to be critical for people to learn higher-order statistical relationships that govern observation [135].

The fourth feature relates abstraction tightly to **generalization and transfer**. As in perceptual systems, there will never be an exact reoccurrence of the same data point. Abstracting from past experiences helps to develop concepts that can be reused to facilitate performance on a never-encountered task.

Finally, abstract ideas can also be assembled. More complex abstract structures can be constructed via operating on existing abstractions, also referred to as coordination (of schemas) [164, 174].

While previous modeling work captures some of the abstraction features in developmental psychology, no model that captures all of them exists. This raises the question of what a minimal model can be that captures all of the abstraction features as described by psychologists. And what basic computational principle allows a learning agent to abstract while exhibiting the aforementioned features? Additionally, how can more complex, abstract structures be constructed by combining components from more superficial abstract structures in a way that is consistent and similar to the aforementioned developmental trajectories?

I explore this question by controlling the generative model of sequences that favor abstraction and studying how the underlying structure can be uncovered by a learning agent. I propose to combine chunking and abstraction learning, being previously discussed in isolation, as the core mechanisms of this model. I argue that chunking — in conjunction with learning abstractions — can give rise to the ability to learn both concrete and abstract patterns while giving the power to assemble the complex from simpler parts.

## 5.4 Summary of the Article

### 5.4.1 Sequences with nested abstract hierarchical structures

This project starts by studying one-dimensional discrete sequences as an extremely simplified version of perception. The perceptual sequence may reflect the underlying environment's inherent nested structures and regularities. To start with studying the necessity of abstraction, we develop a generative model to produce sequences that mimic the emergence of nested hierarchy in natural systems, taking references from the properties of self-diversifying systems [105, 4].

Specifically, this theory hypothesized that a set of simple principles constitutes the diverse observation in the natural world. Such systems 'make infinite use of finite media' whose 'synthesis creates something not present in any of the associated constituents'. Within a self-diversifying system, a set of existing stable objects form stable combinations with one another to form more complex objects. This automatically leads to a variation and oversupply of created objects while producing stable combinations that share similar properties. Examples of such systems include the diverse chemicals constituted by atomic units, the diverse organisms constituted by a combination of genes, and the infinite possibilities constituted by finite means present in the human language.

To capture this feature in the sequence subject to study, we simulated generative models that operate in one-dimensional sequences to simulate perceptual observations. The key assumptions of this model are:

- All objects are made out of a finite combination of atomic elements.
- The observation sequence is sampled from the created stable objects in the existing inventory of the world, where each object in the inventory occurs with a certain probability.
- Existing objects may concatenate and combine into composites, forming new 'things' with similar properties that interact analogously.
- Some created objects share similar properties and belong to one category. This property will make them interact with other things in the world similarly, producing composites that only differ among objects within the same category.

The generative model begins with an inventory of basic atomic units. This inventory expands through the formation of novel stable combinations by concatenating existing objects or categories. Initially, atomic elements randomly combine to

form objects, thereby expanding the inventory. Some of these objects become new categories, representing groups of objects that share similar interaction properties. These categories then serve as additional components, combining with other objects or categories to form new stable combinations to expand the inventory further. The agent observes random samples of objects from the inventory within the artificial world created by the generative model.

### 5.4.2 Two ingredients of abstraction

A learning agent perceives sequences that reflect the underlying nested structure created by the generative model and uses two types of abstraction, along with learning chunks, to uncover what are the unvarying patterns that occur in its perceptual stream.

We introduce two implementations of abstraction notions in HVM that make the algorithm more effective while expanding its capability to uncover a hierarchy of variables.

**Abstraction as organizing chunks via common subparts** The first type of abstraction is finding common parts between the learned chunks. Upon parsing the observational sequence, the model needs to search among its existing learned chunks to retrieve one consistent with the sequence. The number of search steps to allocate the biggest matching item in the parsing tree grows with the size of the dictionary.

Abstraction, as finding common parts between learned chunks, helps the model organize its memory more effectively for information retrieval. The memory of the learned chunks is implemented in a Trie structure: each ancestor node is the common prefix of its children, connecting the longer chunks with the shorter and more frequent chunks in a hierarchical memory recall graph. During parsing, the search starts from the root of the parsing tree, following each leaf node consistent with the sequence, and terminates at the deepest leaf node. Identifying the final node consistent with the upcoming part of the sequence is guaranteed to be the deepest chunk in the tree. Connecting chunks from their common prefixes reduces the search step to the depth of the tree.

**Abstraction as inventing symbols that represent variables** As perceptual sequences contain categories that similarly interact with other objects, uncovering these abstract concepts as categories helps the learning agent acquire higher-order patterns and relations that explain more observations.

The second characteristic of abstraction is to replace the occurrence of distinct chunks using a symbol. The symbol is meant to denote categories of chunks sharing similar interaction properties. The symbol is identified when any chunk that this symbol represents is identified, which helps the agent to identify an underlying pattern in varying observations. This abstraction feature enables more abstract concepts to emerge from concrete patterns in a graded fashion, layer by layer, and the more abstract patterns detected based on the description of the previously acquired symbolic observation description, analogous to human abstraction concept formation during development.

We propose a hierarchical variable learning model (HVM) as an extension of the hierarchical chunking model that combines chunking with the two types of abstractions proposed above. HVM abstracts commonalities among its learned chunks to organize memory in a Trie structure to enhance retrieval efficiency. Furthermore, HVM proposes symbols to represent abstract categories of chunks with similar interaction properties. The model learns chunks on the description of previously learned symbolic patterns. This dual approach discovers abstractions by proposing variables to capture sequence variability and uses chunking operations on an expanding symbol inventory, mirroring concept discovery during cognitive development.

We first show that abstraction via extracting commonalities among chunks reduces parsing search steps. Additionally, proposing symbols to capture categories helps the model learn unvarying patterns that explain a larger part of the sequence. The models that exploit hierarchical structures compress the sequence more effectively than traditional compression methods.

Next, we showed the relation between compression efficiency, abstraction, and generalization. We showed that as the layer of abstraction increases, more abstract, symbolic chunks are learned by the hierarchical variable model, which comes with more distortion in pure symbolic representations and higher sequence parsing likelihood in novel transfer sequences. A more symbolic description of patterns in sequences helps the model to parse novel sequences with less surprise.

Relating the model to human sequence learning, we used the same sequence recall experiment to instruct the model to remember sequences and compared the model's negative log-likelihood to participant sequence recall times. We discovered that the model's negative log likelihood correlates with human sequence recall more strongly than alternative models that do not learn and transfer chunks or variables. We further compared several large language models' (LLMs) negative log-likelihood in the memory experiment. In comparison to the cognitive models, we found that LLMs do not abstract.

## 5.5 Discussion

In this work, we propose a model that learns chunks from the specific to the abstract levels. Chunking endows the learning agent with the ability to generate structural primitives as recurring patterns in sequences. Abstraction enables the agent to symbolize and organize akin patterns into categories. The interplay of the two forces enables learning unvarying patterns, from simple to complex, from concrete to abstract, growing with practice. The compositional nature of sequences encourages recursive reuse, building up the intricate wholes from intermediate parts.

Our work goes beyond previous work in two aspects: the first is that assumptions on primitive abstraction functions are lifted and can be learned from data; the second is that an explicit abstraction can be acquired instead of relying on connectionist systems learning implicit abstractions. This work lays a foundation for a series of further investigations to elucidate the emergence of abstract structures from learning.

Future work can relate this model to more cognitive phenomena or dive into the implementation level to understand the emergence of abstract structures in artificial/biological neural networks. One direction is to relate the parsing graph with memory retrieval. Using commonalities shared among memory items to organize memory implies that giving longer retrieval cues shall help allocate the retrieval and identification of a long memory faster and more accurately than shorter retrieval cues. Other experiments may relate the model's parsing steps with behavioral recognition time. A tree-structured parsing graph implies a logarithmic search time that grows with the number of stored memory items. Future work may test whether the memory retrieval time grows with the memory size or the logarithm of the memory size. Another direction is to relate chunking in the abstract space with the merge operation that allows a step-wise combination of corpus units to generate grammatically intact utterances [176, 43, 41], due to their close resemblance. Additionally, the model predicts the existence of a particular type of memory error: items that appear in similar variable categories are likely to be confused during recall compared to memory items from different categories. Existing evidence suggests this to be the case [34, 76]. Further application of the model may include the explicit grounding of novel abstract concepts based on existing chunks to emulate our cognitive tendency to conceptualize the nonphysical in terms of the physical or the less clearly delineated in terms of the more clearly delineated.

Abstraction also necessarily happens in large AI models used these days. Throwing away information and obtaining vital information is necessary to perform reasonably in any classification task and beyond. Previous work showed that the level of

abstraction increases with the neural network's processing layer [118], and so does the level of information comprehension. However, how abstraction arises and its influence on transfer is still unknown, especially for modern AI systems. For substantially abstract tasks such as arithmetic or algorithmic binary operations, it has been observed that although some neural network models learn and predict very well on the training set after training, good performance on the test set does not emerge until training for an excessive amount [179, 153]. Similarly, intriguing phenomena have been observed, such as a leap of reasoning ability emerging after excessive data [235]. Other works on meta-learning suggested that training neural networks to perform multiple tasks helps the network to transfer skills and solve problems in novel situations. Our work demonstrates a tight relationship between abstraction and transfer. Our result differentiating models that learn chunks and abstractions on transfer sequences suggest that the acquisition of abstraction does not improve performance on training sets that representations on a superficial level can solve, but only on tasks that demand the usage of sufficiently abstract representation. LLMs' inability to exploit variable structure from the training sequence to the transfer sequences suggests the absence of a particular abstract representation that overlaps both the training and the test set. It relates to this question of how such an abstract structure may emerge at one point during the excessive training process. Our work urges future studies to study the emergence of abstraction at different learning stages by probing the model's behavior on tasks that demand different levels of abstraction and inform hypotheses such as are the lower, specific levels of abstraction necessary for the network to realize and come up with higher levels of abstraction in tolerance with higher variability in data.

This algorithm can also be adapted for flexible applications. Future work can further improve computational efficiency adapting to the application in need. For example, one can implement a hash table on each branch of the chunk parsing graph to further reduce the computational steps of chunk parsing. Alternatively, the stored memory chunks can be organized via a semantic relational network or other structures to group similar memory items closer to each other in retrieval or storage space. In parallel computing systems, the time needed to retrieve and identify the correct stored chunk for parsing can be further reduced by harnessing neurally plausible such as a winner-take-all architecture to efficiently trigger the activation of neural populations representing the storage of chunks.

Finally, the abstraction and chunking considered in the context of this work are in the domain of perception. Future work may integrate this modeling framework with action. Compositionality, transfer, and reuse models have also been proposed in classical hierarchical reinforcement learning and resemble human behavior [249]. Abstraction in state space can alleviate the combinatorial explosion that plagues planning: by transforming the state space of a ground Markov Decision Process

to that of an abstract one, task complexity can be reduced, paying a small loss of optimality. Approximate state abstraction condenses prohibitively large task representations into essential information and allows solutions to be tractably computable [2]. Future work could explore connections between this model and theories of hierarchical processing that integrate action and perception [182, 109, 71]. For instance, action could be incorporated as a mechanism to selectively focus on and verify information, aligning it with perceptual expectations.

## 5.6 Article Status

"Building, Reusing, and Generalizing Abstract Representations from Concrete Sequences" (Wu, Thalmann, Dayan, Akata, & Schulz) is a submitted manuscript [247] and, at the time of submission, is under review at *International Conference on Learning Representations*, doi:10.48550/arXiv/2410.21332.

## 5.7 Author Contributions

**Conceptualization:** Shuchen Wu.
**Experiments:** Shuchen Wu, Peter Dayan, Eric Schulz.
**Software:** Shuchen Wu.
**Analysis:** Shuchen Wu, Mirko Thalmann.
**Writing – original draft:** Shuchen Wu.
**Writing – review & editing:** Shuchen Wu, Mirko Thalmann, Peter Dayan, Zeynep Akata, Eric Schulz.

# Part IV

## Outlook

# 6 Discussion

In this thesis, we have proposed computational models that learn concrete and abstract chunks from the ground up, uncovering and factorizing sequences with a nested hierarchical structure. The work also included human behavioral experiments, linking these computational models to human behavior. The models were applied to learn both abstract and concrete patterns in the general domain of sequence learning. Together, this thesis outlines a proposal for how structured representations may emerge from data, inspired by the cognitive mechanism of chunking. It also explored how previously learned chunks can facilitate the composition and reuse of knowledge, enabling the learning of more complex chunks at both concrete and abstract levels.

Like all research, the work presented in this thesis has limitations. One key limitation of the proposed models is their reliance on a greedy parsing strategy. For simplicity, the models always select the longest chunk from the learned dictionary that aligns with the incoming sequence as the basis for parsing. While this heuristic encourages the learning of more complex chunks, it can also lead to rigidity. In some cases, selecting a longer chunk may not be the optimal choice if it has a lower likelihood of occurrence. This approach risks introducing dogmatism, where previously learned chunks overly influence how new chunks are discovered, limiting the model's flexibility in learning alternative sequence fragments that could lead to a more diverse set of chunk entities.

Future work could, therefore, enhance both the rationality and flexibility of the parsing process. One potential improvement would be integrating sampling methods into the parsing algorithm, which could introduce more variability and adaptability in chunk discovery. Alternatively, the parsing strategy could be adapted to account for partial observability in sequences where chunks might be incompletely visible. This could be achieved by sampling the chunk that most closely aligns with the observed sequence based on both prior knowledge and their occurrence likelihood. Incorporating Bayesian inference into the parsing process would allow the model to infer the most likely chunks given the sequences observed so far, improving the model's capacity to learn more flexible and diverse dictionaries.

Another limitation of this work lies in the potential mismatch between our generative model and the perceptual reality we aim to describe. In this thesis, we related chunk learning to a rational process of discovering underlying patterns in sequences, leading us to propose a generative model that produces sequences with a nested hierarchical structure. The chunk-learning models we developed served as recognition models that approximate the inverse of this postulated generative process. However, the assumption of a hierarchical structure may not hold in all domains of sequential data. Some sequences might have an entirely different structure or lack any hierarchy, such as patterns that do not exhibit spatial or temporal continuity.

In such cases, the chunk-learning models proposed here may generate an excessive number of chunks, diverging from an optimally succinct representation of the underlying sequence. For example, consider a sequence where each number is always a multiple of the previous one — neither the HVM nor HCM models would capture this particular rule. Thus, in domains where the underlying structure deviates significantly from the hierarchical assumption, the models may fail to learn useful data representations. In this case, the data would need to be decomposed or transformed into an alternative space that contains a recurring structure to enjoy the advantage of this type of model. This potential deviation highlights the need for future research to explore sequence structures and quantify such mismatches. Since the true generative processes behind sensory data are unknown, studies should aim to align generative models with the specific types of sequences being analyzed. Different real-world sequences may have distinct structural properties. Future work could bridge the gap between the generative model and actual sequence structures by identifying statistical properties, such as power-law coefficients [175, 69], or using alternative measures of compositionality [157, 138] to characterize the deviation between the generative model and the structure of data at hand. In sequences generated by processes that do not guarantee spatial or temporal continuity, it could be interesting to test if the tendency to chunk may mislead humans to learn inefficient or faulty patterns, a seemingly irrational behavior caused by a system evolutionarily adapted to a particular data type.

Despite the limitations in some domains, this hierarchical assumption may underlie many sequential data types, including language or visual-temporal sequences. Applying chunking models to such data can provide valuable insights for practitioners looking to extract structured representations. One particularly exciting direction is behavioral data. From nematodes to fruit flies, from mice to humans, neural ethologists have observed that animals exert complex behavioral repertoires by recursively combining behavioral movement primitives, and has long been postulated that behavior is fundamentally organized by a hierarchical structure [14, 220, 20, 208, 241, 150]. However, this hypothesis has been difficult to test on movement recordings due to a lack of methods to extract behavioral hierarchy in an unsupervised manner

[20, 111, 48]. Future work could apply and extend the models proposed in this thesis to automatically extract animals' "behavioral syllables," allowing an automatic decomposition of complex behavior into modularized hierarchical structures. Decomposing behavior into hierarchies will allow future scientists to study and test hypotheses that have been primarily approached in a qualitative way, bringing an understanding of movement organization into a quantitative domain. Example questions include how behavioral dictionaries are constructed, the organization of movement motifs, how animals chain these motifs to form complex sequences, and how the psychological or energetic state influences the composition of movement motifs. Relating the movement motifs to animal's neural activities may also provide insights into the neural basis of sequence chunking.

Beyond the limitations of the greedy algorithm and assumptions about the generative model, this thesis addresses chunk learning at the computational and algorithmic levels of Marr's [144]. It explores chunking from a computational principle point of view and proposes that the goal of chunking is to find what are the underlying invariant entities in observational sequences. The thesis then proposes an algorithm with minimal but cognitively plausible components to learn chunks. Future work shall build on top of this framework and study chunking on Marr's implementation level — how chunk learning can be implemented by a neural system within a biological substrate. This could involve exploring the biological mechanisms that give rise to sequence chunking and hierarchical compositionality in behavior, and identifying neural interactions that might give rise to equivalent computation of associative and chunk learning as included by the cognitive models in this thesis.

In the past, I have explored this implementation-level question by asking how chunking can be implemented in neuromorphic circuits. In mixed-signal neuromorphic hardware that emulates the firing activities of simple biological neural circuits, we have demonstrated a group of spiking neurons with synaptic plasticity and homeostasis can efficiently parse chunks and learn nested patterns from sequences. This work suggests that the parallel computation of the neural system is naturally efficient for learning and retrieving a successive activation of neural sequences in a computationally efficient way [193]. Additionally, chunk learning can be an algorithm that is especially efficient for a parallel computational system to learn structure via interacting with the environment, a property that the brain exhibits.

Another way to investigate this implementation-level question is to study the neural correlates of chunk learning directly. Literature suggests neural substrates of chunk learning in the human brain, such as neural oscillation frequencies, reflecting the nested structure in linguistic sequences on an organization level of syllables, words, and phrases [114]. These findings resonate with our discovery of rich nested structures in fMRI data [245]. Future work may adapt this model to identify

groups of coordinating functional brain regions or neural population activities while participants are listening to or reading linguistic sequences and use this method to look at how the neural circuits can be held accountable for the emergence of pattern identifications across a variety of linguistic organizational levels.

Alternatively, future work may also explore the neural basis of recurring action sequence patterns directly in biological neural activities. Previous research has observed transient, sequential neural firing activities across species and multiple brain areas, which have been linked to cognitive processes including animal spatial navigation [226], learning [74], sleep [128, 240], planning [53], and the encoding and switching of abstract rules [230]. These firing sequences often exhibit a hierarchical structure [29, 99, 133]. In humans, similar sequences of neural activity have been implicated in cognitive operations like memory retrieval [226], consolidation [80, 35, 64], planning [226], and creative thought [31, 112, 32].

The models presented in this thesis could be adapted to identify and separate recurring patterns of transient neural firing among neural population recordings. These neural sequences could then be interpreted as entities that correspond to specific behavioral patterns. Advancing in this direction could lead to a deeper understanding of how biological neural systems learn, adapt to, and process recurring patterns in sensory sequences. Additionally, hypotheses such as whether the hierarchical organization of behavior is driven by a hierarchical organization of neural activities, which may originate from the structure of naturalistic data, could be tested. This knowledge could, for example, inform the development of improved learning curricula.

Finally, the algorithm proposed here operates primarily on one-dimensional discrete sequences. HCM is extended to learn chunks in higher-dimensional data but limited to 625 dimensions; future work can extend the algorithm further to learn structured representations in higher-dimensional sequential data, such as visual-temporal, proprioceptive, or frequency/sound domain.

Taking sequential data in the visual-temporal domain, for example, symbolic computer vision models have seen the visual perception process as an inverse recognition process to discover unvarying entities in the visual data. Exemplified by works by Zhu et al., computer vision researchers have tried to come up with a set of image primitives and image grammar in order to parse objects, scenes, and events as entities that are interpretable and robust under occlusion and signal perturbation [258, 257, 256].

Although limited in its scalability, this previous approach informs future work to integrate chunk learning with connectionist systems to arrive at structured parsing

of image/video data. Instead of learning directly from the full image space, which usually contains high dimensionality and variability, future work can take the embedding layer of pre-trained neural network models on vision data such as visual transformers and their variants [234, 33, 130, 223, 55, 225], and directly look for recurring entities in the embedding layers, as training on a downstream task shall force the connectionist system to learn compressed representations. Other dimensionality reduction methods, such as vector quantized variational autoencoders, may also facilitate the reduction of embedding space to a manageable lower dimensional space. Applying chunking models to a low-dimensional embedding space or subsequent processing by some similarity-matching algorithms could extract interpretable symbolic entities from implicit intermediate network layers and reveal recurring entities in neural activity patterns, potentially corresponding to recurring entities in the input data. These entities could be perturbed to influence the behavior of downstream neural networks. Furthermore, manipulating this structured representation, such as combining these chunks, may create an explicit compositional structure within the embedding space, which could help to disentangle factors that independently influence observation data, allowing the downstream decoder network to generate data that adheres to the compositional structure. This approach might also offer insights into the texture bias in computer vision models compared to the shape bias in human vision, possibly due to the different chunk content learned by these two systems [79].

More significantly, applying variants of HCM and HVM to the embedding spaces of connectionist models that process high-dimensional data could bridge the gap between discrete entities identified by perception and the high-dimensional continuous representations typical of connectionist systems. This may enable models to form object-based representations and learn relations defined at a symbolic level, thereby improving the reliability of connectionist models while enabling the learning and transfer of previously learned entities, potentially reconciling the gap between symbolic and connectionist approaches to artificial intelligence and arriving at a modern solution to an ancient problem.

# 7 Conclusion

Navigating a bustling city may seem effortless, but beneath this fluid interaction lies a remarkable cognitive feat: the ability to continuously process a torrent of sensory data, distinguishing relevant entities—such as traffic signals, road conditions, and food stands—from the chaotic stream of perception. This seamless parsing of the world into discrete units is fundamental to human cognition, yet both connectionist and symbolic AI models have struggled to explain how entities emerge from perception in such a natural, efficient manner. For centuries, philosophers, psychologists, and cognitive scientists have debated this core process, and this thesis focuses on chunking as a plausible cognitive mechanism that bridges this gap.

The thesis begins by experimentally probing human chunking behavior in a serial reaction time task. The results of two experiments suggest that people adapt their chunking strategies dynamically, attuned to both the statistical properties of sequences and the demands of the task at hand. The observed behaviors align with a rational model of chunk learning that strategically balances the trade-off between speed and accuracy—a principled computational account of efficient segmentation and organization.

Building on these insights, the second project explores chunking from a computational and normative standpoint. A generative model is proposed that unearths hierarchical patterns within sequences, recursively uncovering nested structures that serve as the fundamental building blocks of perception. These chunks become entities, not merely for understanding one-dimensional sequences, but as reusable and composable components in more complex, multidimensional environments. The proposed model is extended to learn part-whole structures from visual-temporal data, uncovering meaningful patterns in brain function and demonstrating the versatility of chunking as a learning mechanism.

In the third project, chunking moves beyond concrete sequence elements to capture more abstract motifs. Participants in two additional experiments progressively learn and transfer these motifs across cognitively demanding serial recall tasks, suggesting a deep, transferable chunking process. The corresponding model captures similar transfer behaviors, distinguishing between superficial chunk associations and deeper, abstract motifs. This reveals chunking's potential as a mechanism for abstract pattern

recognition, with implications for how humans generalize from past experiences to novel situations.

The final project pushes the boundaries of chunking further into the realm of abstraction, proposing a hierarchical variable model (HVM) that layers abstraction over concrete chunking. This model not only compresses and organizes information through chunking but also generalizes these chunks into abstract variables, yielding a structured, flexible memory organization. The model's striking alignment with human behavior demonstrates its potential to outperform large language models by learning contextually relevant, nested categories, emphasizing both memory efficiency and generalization.

Altogether, this thesis argues that chunking—a simple computational mechanism for associating nearby chunks in temporal and spatial proximity—serves as a fundamental process for singling out stable entities from the noisy perceptual stream. Through computational modeling and empirical evidence, the thesis shows how chunking, starting from scratch, can scaffold an agent's understanding of the world, forming the basis for a structured world model learned through experience.

This work advances our understanding of chunking beyond traditional sequence learning. It illuminates how chunking compresses and organizes perceptual data, supports the learning of part-whole hierarchies, and facilitates compositionality and transfer across abstract domains. The thesis underscores chunking's centrality in human cognition, intersecting areas such as associative learning, Gestalt theory, and grammar acquisition. It calls for future investigations into chunking's role as a discovery mechanism across cognitive domains, inviting further exploration of its neural basis, behavioral relevance, and potential as a method for uncovering hierarchical regularities in complex data.

# Bibliography


[1] Pieter Abbeel, Daphne Koller, and Andrew Y. Ng. "Learning Factor Graphs in Polynomial Time & Sample Complexity". In: *CoRR* abs/1207.1366 (2012). arXiv: 1207.1366 (cit. on p. 5).

[2] David Abel, D Ellis Hershkowitz, and Michael L Littman. "Near Optimal Behavior via Approximate State Abstraction". en. In: (), p. 9 (cit. on p. 50).

[3] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. 2nd ed. Archived from the original on 26 February 2009. Retrieved 22 June 2012. MIT Press, 1996 (cit. on p. 38).

[4] William L. Abler. "On the particulate principle of self-diversifying systems". en. In: *Journal of Social and Biological Structures* 12.1 (Jan. 1989), pp. 1–13 (cit. on p. 45).

[5] Srinivas M Aji and Robert J Mceliece. "The Generalized Distributive Law". In: 46.2 (2000), pp. 325–343 (cit. on p. 5).

[6] Pedro Alves, Chris Foulon, Vyacheslav Karolis, et al. "An improved neuroanatomical model of the default-mode network reconciles previous neuroimaging and neuropathological findings". In: *Communications Biology* 2 (Oct. 2019), pp. 1–14 (cit. on p. 26).

[7] John R. Anderson. *How can the human mind occur in the physical universe?* Oxford University Press, 2007 (cit. on p. 24).

[8] Aristotle. *Categories*. Translated by E. M. Edghill (cit. on p. 6).

[9] Aristotle. *Categories; On Interpretation; Prior Analytics*. Ed. by H. P. Cooke and Hugh Tredennick. Loeb Classical Library 325. Cambridge, MA: Harvard University Press, 1938 (cit. on pp. 6, 33).

[10] Aristotle. *On Interpretation*. Translated by E. M. Edghill (cit. on p. 6).

[11] Aristotle. *Posterior Analytics*. Translated by G. R. G. Mure (cit. on p. 6).

[12] Aristotle. *Prior Analytics*. Translated by A. J. Jenkinson (cit. on p. 6).

[13] Jane Ashby and Keith Rayner. "Eye movements during reading: Eye guidance in reading and scene perception". In: *Current Directions in Psychological Science* 14.5 (2005), pp. 218–221 (cit. on p. 19).

[14] Gerard P. Baerends. "The functional organization of behaviour". In: *Animal Behaviour* 24 (1976), pp. 726–738 (cit. on p. 54).

[15] Richard P. Bagozzi, Willem J. M. I. Verbeke, Roeland C. Dietvorst, et al. “Theory of Mind and Empathic Explanations of Machiavellianism: A Neuroscience Perspective”. In: *Journal of Management* 39.7 (2013), pp. 1760–1798. eprint: https://doi.org/10.1177/0149206312471393 (cit. on p. 27).

[16] Lawrence W. Barsalou. “Perceptual symbol systems”. In: *Behavioral and Brain Sciences* 22.4 (1999), pp. 577–660 (cit. on p. 41).

[17] Lawrence W. Barsalou. “Perceptual symbol systems”. en. In: *Behavioral and Brain Sciences* 22.4 (Aug. 1999). Publisher: Cambridge University Press, pp. 577–660 (cit. on p. 43).

[18] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, et al. “Relational inductive biases, deep learning, and graph networks”. In: *arXiv preprint arXiv:1806.01261* (2018) (cit. on p. 3).

[19] Michael J Beran and Audrey E Parrish. “Capuchin monkeys (Cebus apella) treat hidden objects as "persistent": An examination of object permanence and representation”. In: *Journal of Comparative Psychology* 128.1 (2014), p. 77 (cit. on p. 3).

[20] Gordon Berman, William Bialek, and Joshua Shaevitz. “Predictability and hierarchy in Drosophila behavior”. In: *Proceedings of the National Academy of Sciences* 113 (Oct. 2016) (cit. on pp. 54, 55).

[21] Savita Bernal, Ghislaine Dehaene-Lambertz, Séverine Millotte, and Anne Christophe. “Two-year-olds compute syntactic structure on-line”. In: *Dev Sci* 13.1 (Jan. 2010), pp. 69–76 (cit. on p. 41).

[22] Marvin Binz, Ishita Dasgupta, Ashok Jagadish, et al. “Meta-learned models of cognition”. In: *Behavioral and Brain Sciences* 28 (2023) (cit. on p. 42).

[23] Daniel G. Bobrow. “Natural language input for a computer problem solving system”. In: *Proceedings of the International Conference on Computational Linguistics*. 1964, pp. 146–154 (cit. on p. 4).

[24] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, et al. *End to End Learning for Self-Driving Cars*. 2016. arXiv: 1604.07316 [cs.CV] (cit. on p. 3).

[25] George Boole. *The Laws of Thought (1854)*. London, The Open court publishing company, 1854 (cit. on p. 34).

[26] Daniel Bor, John Duncan, Richard J. Wiseman, and Adrian M. Owen. “Encoding strategies dissociate prefrontal activity from working memory demand”. In: *Neuron* 37.2 (2003) (cit. on p. 7).

[27] Matthew M. Botvinick. “Hierarchical cognitive control and the hierarchical organization of behavior”. In: *Neuron* 73.5 (2012), pp. 971–990 (cit. on p. 24).

[28] Timothy F. Brady, Talia Konkle, and George A. Alvarez. “Compression in Visual Working Memory: Using Statistical Regularities to Form More Efficient Memory Representations”. In: *Journal of Experimental Psychology: General* 138.4 (2009) (cit. on pp. 7–9, 15).

[29] Steven L. Bressler and Emmanuelle Tognoli. “Operational principles of neurocognitive networks”. In: *International Journal of Psychophysiology* 60.2 (2006), pp. 139–148 (cit. on p. 56).

[30] Marc Brysbaert and Boris New. “Faster syntactic parsing in native-language reading: What makes it possible?” In: *Trends in Cognitive Sciences* 9.6 (2005), pp. 283–286 (cit. on p. 19).

[31] György Buzsáki. “Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning”. In: *Hippocampus* 25.10 (Oct. 2015), pp. 1073–1188 (cit. on p. 56).

[32] György Buzsáki. “Neural syntax: cell assemblies, synapsembles and readers”. In: *Neuron* 68.3 (Nov. 2010), pp. 362–385 (cit. on p. 56).

[33] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, et al. “End-to-End Object Detection with Transformers”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 213–229 (cit. on p. 57).

[34] Cameron M. Carpenter, Charlene E. Webb, Alexandria A. Overman, and Nancy A. Dennis. “Within-category similarity negatively affects associative memory performance in both younger and older adults”. In: *Memory* 31.1 (2023), pp. 77–91 (cit. on p. 48).

[35] Matthew F. Carr, Shantanu Jadhav, and Loren M. Frank. “Hippocampal replay in the awake state: A potential substrate for memory consolidation and retrieval”. In: *Nature Neuroscience* 14 (2011), pp. 147–153 (cit. on p. 56).

[36] W. G. Chase and H. A. Simon. “Perception in chess”. In: *Cognitive Psychology* 4(1).55-81 (1973) (cit. on p. 15).

[37] Patricia W Cheng and Keith J Holyoak. “Pragmatic reasoning schemas”. In: *Cognitive Psychology* 17.4 (1985), pp. 391–416 (cit. on p. 38).

[38] Francois Chollet. “On the measure of intelligence”. In: *arXiv preprint arXiv:1911.01547*. 2019 (cit. on pp. 4, 43).

[39] Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, 1965 (cit. on p. 7).

[40] Noam Chomsky. *Aspects of the Theory of Syntax*. 11. MIT press, 2014 (cit. on p. 34).

[41] Noam Chomsky. “Problems of projection”. In: *Lingua* 130 (2013). SI: Syntax and cognition: core ideas and results in syntax, pp. 33–49 (cit. on p. 48).

[42] Noam Chomsky. *Rules and Representations*. Columbia University Press, 1980 (cit. on p. 7).

[43] Noam Chomsky. *The Minimalist Program*. MIT Press, 1995 (cit. on p. 48).

[44] Noam Chomsky and George A. Miller. “Finite State Languages”. In: *Information and Control* 1.2 (1958), pp. 91–112 (cit. on p. 8).

[45] Axel Cleeremans, David Servan-Schreiber, and James L. McClelland. “Finite state automata and simple recurrent networks”. In: *Neural Computation* 1 (1989). Place: US Publisher: MIT Press, pp. 372–381 (cit. on p. 23).

[46] Tom N. Cornsweet. “The Staircase-Method in Psychophysics”. In: *The American Journal of Psychology* 75.3 (1962), pp. 485–491 (cit. on p. 19).

[47] Valentina Cuccio and Vittorio Gallese. “A Peircean account of concepts: grounding abstraction in phylogeny through a comparative neuroscientific perspective”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 373.1752 (June 2018). Publisher: Royal Society, p. 20170128 (cit. on p. 43).

[48] Sandeep Robert Datta, David J. Anderson, Kristin Branson, Pietro Perona, and Andrew Leifer. “Computational Neuroethology: A Call to Action”. In: *Neuron* 104.1 (2019), pp. 11–24 (cit. on p. 55).

[49] Stanislas Dehaene, Fosca Al Roumi, Yair Lakretz, Samuel Planton, and Mathias Sablé-Meyer. “Symbols and mental programs: a hypothesis about human singularity”. en. In: *Trends in Cognitive Sciences* 26.9 (Sept. 2022), pp. 751–766 (cit. on pp. 6, 41).

[50] Stanislas Dehaene, Florent Meyniel, Catherine Wacongne, Liping Wang, and Christophe Pallier. “The Neural Representation of Sequences: From Transition Probabilities to Algebraic Patterns and Linguistic Trees”. In: 88.1 (2015) (cit. on pp. 7, 41).

[51] S. Della Pietra, V. Della Pietra, and J. Lafferty. “Inducing features of random fields”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.4 (1997), pp. 380–393 (cit. on p. 5).

[52] Frank N. Dempster. “Memory span: Sources of individual and developmental differences”. In: *Psychological Bulletin* 89.1 (1981) (cit. on p. 7).

[53] Kamran Diba and György Buzsáki. “Forward and reverse hippocampal place-cell sequences during ripples”. In: *Nature Neuroscience* 10 (2007), pp. 1241–1242 (cit. on p. 56).

[54] Don C Donderi. “Visual complexity: a review.” In: *Psychological bulletin* 132.1 (2006), p. 73 (cit. on p. 26).

[55] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *arXiv preprint arXiv:2010.11929* (2020) (cit. on p. 57).

[56] Lindsay N. Driscoll, Louisa Duncker, and Christopher D. Harvey. “Representational drift: Emerging theories for continual learning and experimental future directions”. In: *Current Opinion in Neurobiology* 76 (2022), p. 102609 (cit. on p. 43).

[57] Lindsay N. Driscoll, Krishna V. Shenoy, and David Sussillo. “Flexible multitask computation in recurrent networks utilizes shared dynamical motifs”. In: *Nature Neuroscience* (2023), pp. 1–15 (cit. on p. 43).

[58] D. E. Dulany, R. A. Carlson, and G. I. Dewey. “A case of syntactical learning and judgment: How conscious and how abstract?” In: *Journal of Experimental Psychology: General* 113.4 (1984), pp. 541–555 (cit. on p. 8).

[59] Karl Duncker. “On problem-solving.” en. Trans. by Lynne S. Lees. In: *Psychological Monographs* 58.5 (1945), pp. i–113 (cit. on pp. 38, 42).

[60] Louisa* Duncker, Lindsay N.* Driscoll, Krishna V. Shenoy, Maneesh Sahani, and David Sussillo. “Organizing recurrent network dynamics by task-computation to enable continual learning”. In: *Advances in Neural Information Processing Systems*. Vol. 33. NeurIPS. 2020, p. 78 (cit. on p. 43).

[61] Hermann Ebbinghaus. *Memory: A contribution to experimental psychology*. Teachers College, Columbia University, 1913 (cit. on p. 24).

[62] Dennis E. Egan and Barry J. Schwartz. “Chunking in recall of symbolic drawings”. In: *Memory & Cognition* 7.2 (1979) (cit. on pp. 8, 9, 15, 21).

[63] Christian von Ehrenfels. “Über "Gestaltqualitäten"”. In: *Vierteljahrsschrift für wissenschaftliche Philosophie* 14 (1890), pp. 249–292 (cit. on p. 34).

[64] Howard Eichenbaum. “Memory on time”. In: *Trends in Cognitive Sciences* 17.2 (2013), pp. 81–88 (cit. on p. 56).

[65] Kevin Ellis, Adam Albright, Armando Solar-Lezama, Joshua B. Tenenbaum, and Timothy J. O’Donnell. “Synthesizing theories of human language with Bayesian program induction”. en. In: *Nature Communications* 13.1 (Aug. 2022), p. 5024 (cit. on p. 28).

[66] Kevin Ellis, Catherine Wong, Maxwell Nye, et al. *DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning*. arXiv:2006.08381 [cs]. June 2020 (cit. on pp. 28, 29).

[67] Nick C. Ellis. “Sequencing in SLA: Phonological memory, chunking, and points of order”. In: *Studies in Second Language Acquisition* 18.1 (1996) (cit. on pp. 8, 15).

[68] Nóra Éltető, Dezső Nemeth, Karolina Janacsek, and Peter Dayan. “Tracking human skill learning with a hierarchical Bayesian sequence model”. In: *PLoS Comput Biol* 18.11 (Nov. 2022), e1009866 (cit. on p. 18).

[69] Ramon Ferrer-i-Cancho and Ricard V. Sole. “Theories of linguistic complexity”. In: *Proceedings of the National Academy of Sciences* 98.22 (2001), pp. 13475–13480 (cit. on p. 54).

[70] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Vol. 70. PMLR. 2017, pp. 1126–1135 (cit. on p. 42).

[71] A Fisher and RPN Rao. “Recursive neural programs: A differentiable framework for learning compositional part-whole hierarchies and image grammars”. In: *PNAS Nexus* 2.11 (2023), pgad337 (cit. on p. 50).

[72] W. Tecumseh Fitch and Angela D. Friederici. “Artificial grammar learning meets formal language theory: an overview”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 367.1598 (July 2012). Publisher: Royal Society, pp. 1933–1955 (cit. on p. 7).

[73] Timo Flesch, Jan Balaguer, Ronald Dekker, Hamed Nili, and Christopher Summerfield. “Comparing Continual Task Learning in Minds and Machines”. In: *Proceedings of the National Academy of Sciences* 115.44 (2018), E10313–E10322 (cit. on p. 29).

[74] David J. Foster and Matthew A. Wilson. “Reverse replay of behavioural sequences in hippocampal place cells during the awake state”. In: *Nature* 440 (2006), pp. 680–683 (cit. on p. 56).

[75] Robert M. French, Caspar Addyman, and Denis Mareschal. “TRACX: A recognition-based connectionist framework for sequence segmentation and chunk extraction.” en. In: *Psychological Review* 118.4 (Oct. 2011), pp. 614–636 (cit. on p. 23).

[76] David A. Gallo and Henry L. Roediger. “The effects of associations and aging on illusory recollection”. In: *Memory & Cognition* 31.7 (2003), pp. 1036–1044 (cit. on p. 48).

[77] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, et al. “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673 (cit. on p. 4).

[78]Robert Geirhos, Patricia Rubisch, Claudio Michaelis, et al. “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness”. In: *arXiv preprint arXiv:1811.12231* (2018) (cit. on p. 3).

[79]Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, et al. “Generalisation in humans and deep neural networks”. In: (2018) (cit. on p. 57).

[80]Hagar Gelbard-Sagiv, Ramon Mukamel, Michal Harel, Rafael Malach, and Itzhak Fried. “Internally generated reactivation of single neurons in human hippocampus during free recall”. In: *Science* 322.5898 (2008), pp. 96–101 (cit. on p. 56).

[81]Samuel J. Gershman. “Origin of perseveration in the trade-off between reward and complexity”. In: *Cognition* 204 (2020), p. 104394 (cit. on p. 19).

[82]Simona Ghetti and Joshua Lee. “Children’s episodic memory”. In: *Wiley Interdisciplinary Reviews: Cognitive Science* 2.4 (2011), pp. 365–373 (cit. on p. 33).

[83]Robert Glaser. “Expert knowledge and processes of thinking”. In: *Enhancing thinking skills in the sciences and mathematics*. Ed. by Diane F. Halpern. Lawrence Erlbaum Associates, Inc., 1992, pp. 63–75 (cit. on p. 43).

[84]F. Gobet and H. A. Simon. “Expert chess memory: Revisiting the chunking hypothesis”. In: *Memory* 6.255-255 (1998) (cit. on pp. 15, 19).

[85]Fernand Gobet and Gary Clarkson. “Chunks in expert memory: Evidence for the magical number four...or is it two?” In: *Memory* 12.6 (2004) (cit. on p. 7).

[86]Fernand Gobet, Peter C.R. Lane, Steve Croker, et al. “Chunking mechanisms in human learning”. In: *Trends in Cognitive Sciences* 5.6 (2001) (cit. on pp. 7, 15, 24).

[87]Sharon Goldwater, Thomas Griffiths, and Mark Johnson. “A Bayesian Framework for Word Segmentation: Exploring the Effects of Context”. In: *Cognition* 112 (Apr. 2009), pp. 21–54 (cit. on pp. 18, 23).

[88]Rebecca L Gomez and LouAnn Gerken. “Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge”. en. In: *Cognition* 70.2 (Mar. 1999), pp. 109–135 (cit. on pp. 8, 24).

[89]Rebecca L. Gómez. “Variability and Detection of Invariant Structure”. en. In: *Psychological Science* 13.5 (Sept. 2002). Publisher: SAGE Publications Inc, pp. 431–436 (cit. on pp. 8, 24).

[90]Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2015) (cit. on p. 27).

[91]Jonathan Grainger and Johannes C. Ziegler. “A dual-route approach to orthographic processing”. In: *Frontiers in Psychology* 2.APR (2011) (cit. on pp. 8, 9).

[92]Ann M Graybiel. “The basal ganglia and chunking of action repertoires”. In: *Neurobiology of learning and memory* 70.1-2 (1998), pp. 119–136 (cit. on p. 15).

[93]Ann M. Graybiel. “The basal ganglia and chunking of action repertoires”. In: *Neurobiology of Learning and Memory*. Vol. 70. 1998, pp. 1–2 (cit. on pp. 8, 15).

[94]Klaus Greff, Sjoerd Van Steenkiste, and Juergen Schmidhuber. “Binding problem in artificial neural networks”. In: *arXiv preprint arXiv:2004.13665* (2020) (cit. on p. 3).

[95] Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. "Meta-learning for low-resource neural machine translation". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2018, pp. 3622–3631 (cit. on p. 42).

[96] E. R. Guthrie. "Conditioning as a principle of learning". In: *Psychological Review* 37.5 (1930), pp. 412–428 (cit. on p. 6).

[97] G. S. Halford, W. H. Wilson, and S. Phillips. "Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology". In: *Behavioral and Brain Sciences* 21.6 (1998) (cit. on p. 7).

[98] Moritz Hardt, Benjamin Recht, and Yoram Singer. "Train faster, generalize better: Stability of stochastic gradient descent". In: *arXiv preprint arXiv:1509.01240* (2016) (cit. on p. 27).

[99] Kenneth D. Harris. "Neural signatures of cell assembly organization". In: *Nature Reviews Neuroscience* 6 (2005), pp. 399–407 (cit. on p. 56).

[100] Thomas Hartley. *Observations on Man, His Frame, His Duty, and His Expectations*. London: Samuel Richardson, 1749 (cit. on p. 6).

[101] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. "Neuroscience-inspired artificial intelligence". In: *Neuron* 95.2 (2017), pp. 245–258 (cit. on p. 4).

[102] Felix Hill, Andrew Lampinen, Robin Schneider, Stephen Clark, and Matthew Botvinick. "Environmental drivers of systematicity and generalization in a situated agent". In: *arXiv preprint arXiv:1910.00571* (2019) (cit. on p. 4).

[103] Thomas Hobbes. *Leviathan*. London: Andrew Crooke, 1651 (cit. on p. 6).

[104] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. "Meta-learning in neural networks: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2021), pp. 5149–5169 (cit. on p. 42).

[105] W. von Humboldt and M. Losonsky. *Humboldt: 'On Language': On the Diversity of Human Language Construction and Its Influence on the Mental Development of the Human Species*. Cambridge Texts in the History of Philosophy. Cambridge University Press, 1999 (cit. on p. 45).

[106] David Hume. *An Enquiry Concerning Human Understanding*. [1748] Archived from the original on 10 July 2018. Retrieved 14 March 2015. London: A. Millar, 1777 (cit. on p. 6).

[107] Nicholas Ichien, Hongjing Lu, and Keith J. Holyoak. "Predicting patterns of similarity among abstract semantic relations." en. In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 48.1 (Jan. 2022), pp. 108–121 (cit. on p. 42).

[108] William James. *The Principles of Psychology*. Vol. 1. Henry Holt and Co., 1890 (cit. on p. 5).

[109] LP Jiang and RPN Rao. "Dynamic predictive coding: A model of hierarchical sequence learning and prediction in the neocortex". In: *PLoS Computational Biology* 20.2 (2024), e1011801 (cit. on p. 50).

[110]Xin Jin, Fatuel Tecuapetla, and Rui M. Costa. “Basal ganglia subcircuits distinctively encode the parsing and concatenation of action sequences”. In: *Nature Neuroscience* 17.3 (2014) (cit. on p. 8).

[111]Robert Evan Johnson, Scott Linderman, Thomas Panier, et al. “Probabilistic Models of Larval Zebrafish Behavior Reveal Structure on Many Scales”. In: *Current Biology* 30.1 (2020), 70–82.e4 (cit. on p. 55).

[112]Hannah R. Joo and Loren M. Frank. “The hippocampal sharp wave–ripple in memory retrieval for immediate use and consolidation”. en. In: *Nature Reviews Neuroscience* 19.12 (Dec. 2018). Number: 12 Publisher: Nature Publishing Group, pp. 744–757 (cit. on p. 56).

[113]Rudolf E Kalman. “A new approach to linear filtering and prediction problems”. In: *Journal of Basic Engineering* 82.1 (1960), pp. 35–45 (cit. on p. 5).

[114]Greta Kaufeld, Hans Rutger Bosker, Sanne ten Oever, et al. “Linguistic Structure and Meaning Organize Neural Oscillations into a Content-Specific Hierarchy”. en. In: *Journal of Neuroscience* 40.49 (Dec. 2020). Publisher: Society for Neuroscience Section: Research Articles, pp. 9467–9475 (cit. on pp. 8, 55).

[115]Iring Koch and Joachim Hoffmann. “Patterns, chunks, and hierarchies in serial reaction-time tasks”. In: *Psychological Research* 63.1 (2000) (cit. on pp. 8, 9, 15, 16).

[116]Kurt Koffka. *Principles of Gestalt Psychology*. Retrieved 13 October 2019. New York: Harcourt, Brace, 1935, p. 176 (cit. on p. 6).

[117]George Konidaris. “On the necessity of abstraction”. en. In: *Current Opinion in Behavioral Sciences*. Artificial Intelligence 29 (Oct. 2019), pp. 1–7 (cit. on p. 41).

[118]Robert Kozma et al. “Artificial intelligence: Foundations, theory, and algorithms”. In: *Journal of Artificial Intelligence Research* 61 (2018), pp. 203–207 (cit. on p. 49).

[119]Frank R. Kschischang, Brendan J. Frey, and Hans Andrea Loeliger. “Factor graphs and the sum-product algorithm”. In: *IEEE Transactions on Information Theory* (2001) (cit. on p. 5).

[120]John E Laird, Paul S Rosenbloom, and Allen Newell. “Towards Chunking as a General Learning Mechanism.” In: *AAAI*. 1984, pp. 188–192 (cit. on p. 15).

[121]B. Lake, R. Salakhutdinov, and J. Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* 350 (2015), pp. 1332–1338 (cit. on pp. 28, 29).

[122]Brenden M. Lake and Marco Baroni. *Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks*. arXiv:1711.00350 [cs]. June 2018 (cit. on p. 3).

[123]Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* (2015) (cit. on p. 23).

[124]George Lakoff and Mark Johnson. *Metaphors we live by*. University of Chicago Press, 1980 (cit. on pp. 38, 42).

[125]KS Lashley. “The problem of serial order in behavior”. In: *Cerebral mechanisms in behavior*. Ed. by L. A. Jeffress. 7. New York: Wiley, 1951 (cit. on pp. 8, 15, 21).

[126] John M. Lawler. “Metaphors We Live by”. In: *Language* 59.1 (1983), pp. 201–207 (cit. on pp. 38, 42).

[127] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444 (cit. on pp. 3, 27).

[128] Albert K. Lee and Matthew A. Wilson. “Memory of sequential experience in the hippocampus during slow wave sleep”. In: *Neuron* 36 (2002), pp. 1183–1194 (cit. on p. 56).

[129] Stan Z Li. *Markov Random Field Modeling in Image Analysis*. Springer, 1995 (cit. on p. 5).

[130] Ze Liu, Yutong Lin, Yue Cao, et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *arXiv preprint arXiv:2103.14030* (2021) (cit. on p. 57).

[131] John Locke. *An Essay Concerning Human Understanding*. London: Thomas Bassett, 1690 (cit. on p. 6).

[132] Hongjing Lu, Nicholas Ichien, and Keith J. Holyoak. *Probabilistic Analogical Mapping with Semantic Relation Networks*. Tech. rep. arXiv:2103.16704. arXiv:2103.16704 [cs] type: article. arXiv, Oct. 2021 (cit. on p. 42).

[133] Artur Luczak, Paul Bartho, and Ken D. Harris. “Gating of sensory input by spontaneous cortical activity”. In: *Journal of Neuroscience* 35.11 (2015), pp. 4103–4111 (cit. on p. 56).

[134] Floris Luyckx, Hannah Sheahan, and Christopher Summerfield. “Non-monotonic spatial integration of hierarchical rules in human frontoparietal cortex”. In: *Journal of Neuroscience* 39.22 (2019), pp. 4366–4376 (cit. on p. 4).

[135] Christopher W. Lynn, Ari E. Kahn, Nathaniel Nyema, and Danielle S. Bassett. “Abstract representations of events arise from mental errors in learning and memory”. en. In: *Nature Communications* 11.1 (May 2020), p. 2313 (cit. on p. 44).

[136] Donald G MacKay. “The Problems of Flexibility, Fluency, and Speed-Accuracy Trade-Off in Skilled Behavior”. en. In: *Psychological Review* 89.5 (1982), pp. 483–506 (cit. on p. 4).

[137] Maxime Maheu, Florent Meyniel, and Stanislas Dehaene. “Rational arbitration between statistics and rules in human sequence processing”. en. In: *Nature Human Behaviour* 6.8 (May 2022), pp. 1087–1103 (cit. on p. 8).

[138] Christopher D. Manning and Hinrich Schütze. “Foundations of Statistical Natural Language Processing”. In: *MIT Press* (1999) (cit. on p. 54).

[139] G. F. Marcus, S. Vijayan, S. Bandi Rao, and P. M. Vishton. “Rule learning by seven-month-old infants”. eng. In: *Science (New York, N.Y.)* 283.5398 (Jan. 1999), pp. 77–80 (cit. on p. 34).

[140] Gary Marcus. “Deep learning: A critical appraisal”. In: *arXiv preprint arXiv:1801.00631* (2018) (cit. on p. 43).

[141] Gary Marcus. “The Acquisition of the English Past Tense in Children and Connectionist Models: Rules, Roadblocks and Resolutions”. In: *Cognition* 56.3 (1995), pp. 271–339 (cit. on p. 34).

[142] Gary Marcus. *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. Cambridge, MA: MIT Press, 2001 (cit. on p. 34).

[143] Gary Marcus. "The next decade in AI: Four steps towards robust artificial intelligence". In: *arXiv preprint arXiv:2002.06177* (2020) (cit. on p. 4).

[144] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco: W.H. Freeman and Company, 1982 (cit. on pp. 4, 55).

[145] Fabien Mathy and Jacob Feldman. "What's magic about magic numbers? Chunking and data compression in short-term memory". In: *Cognition* (2012) (cit. on p. 7).

[146] John McCarthy. "Programs with common sense". In: *RLE and MIT Computation Center Research Laboratory of Electronics* 125 (1960), pp. 99–157 (cit. on p. 4).

[147] Andréane Melançon and Rushen Shi. "Representations of abstract grammatical feature agreement in young children". In: *J Child Lang* 42.6 (Nov. 2015). Epub 2015 Jan 30, pp. 1379–1393 (cit. on p. 41).

[148] James Mill. *An Analysis of the Phenomena of the Human Mind*. London: Longman, Rees, Orme, Brown, and Green, 1829 (cit. on p. 6).

[149] John Stuart Mill. *A System of Logic, Ratiocinative and Inductive*. London: John W. Parker, 1843 (cit. on p. 6).

[150] George A Miller, Eugene Galanter, and Karl H Pribram. *Plans and the Structure of Behavior*. Henry Holt and Co., 1960 (cit. on p. 54).

[151] George A. Miller. "The magical number seven, plus or minus two: Some limits on our capacity for processing information". In: *Psychological Review* 63.2 (1956), pp. 81–97 (cit. on p. 7).

[152] George A. Miller. "The magical number seven, plus or minus two: some limits on our capacity for processing information". In: *Psychological Review* (1956) (cit. on pp. 7, 8, 15, 21).

[153] Jack Miller, Charles O'Neill, and Thang Bui. *Grokking Beyond Neural Networks: An Empirical Exploration with Model Complexity*. 2024. arXiv: `2310.17247 [cs.LG]` (cit. on p. 49).

[154] Beren Millidge. *Towards a Mathematical Theory of Abstraction*. Tech. rep. arXiv:2106.01826. arXiv:2106.01826 [cs, stat] type: article. arXiv, June 2021 (cit. on p. 38).

[155] Melanie Mitchell. "Why AI is harder than we think". In: *arXiv preprint arXiv:2104.12871* (2021) (cit. on p. 43).

[156] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518 (2015), pp. 529–533 (cit. on p. 3).

[157] Nick Moran. "Complexity, learning and compositionality". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 35. 2013, pp. 1056–1061 (cit. on p. 54).

[158] Michael C. Mozer, Harold Pashler, and Matthew H. Wilder. "Neural mechanisms of reactivation-induced updating in human memory". In: *Proceedings of the National Academy of Sciences* 110.43 (2013), pp. 17450–17455 (cit. on p. 24).

[159] Jaap MJ Murre and Joeri Dros. "Replication and analysis of Ebbinghaus' forgetting curve". In: *PLOS ONE* 10.7 (2015), e0120644 (cit. on p. 24).

[160]Diana M Müssgens and Fredrik Ullén. “Transfer in Motor Sequence Learning: Effects of Practice Schedule and Sequence Context”. In: *Frontiers in Human Neuroscience* 9.November (2015) (cit. on pp. 15, 16, 23).

[161]Allen Newell and Herbert A. Simon. “Computer science as empirical inquiry: Symbols and search”. In: *Communications of the ACM*. Vol. 19. 3. 1976, pp. 113–126 (cit. on pp. 4, 24).

[162]Mary Jo Nissen and Peter Bullemer. “Attentional requirements of learning: Evidence from performance measures”. In: *Cognitive psychology* 19.1 (1987), pp. 1–32 (cit. on p. 15).

[163]Timothy J. O’Donnell, Noah D. Goodman, and Joshua B. Tenenbaum. *Fragment Grammars: Exploring Computation and Reuse in Language*. Tech. rep. 2009 (cit. on p. 18).

[164]Stellan Ohlsson and Erno Lehtinen. “Abstraction and the acquisition of complex ideas”. en. In: *International Journal of Educational Research* 27.1 (Jan. 1997), pp. 37–48 (cit. on p. 44).

[165]Gergö Orbán, József Fiser, Richard N. Aslin, and Máté Lengyel. “Bayesian learning of visual chunks by human observers”. In: *Proceedings of the National Academy of Sciences of the United States of America* 105.7 (2008) (cit. on pp. 8, 9, 18, 27).

[166]Kevin Paterson, Min Chang, Zhao Sainan, et al. “Effects of Normative Aging on Eye Movements during Reading”. In: *Vision* 4 (Jan. 2020) (cit. on p. 26).

[167]I.P. Pavlov. *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Ed. and trans. by G.V. Anrep. London: Oxford University Press, 1927, p. 142 (cit. on p. 6).

[168]Judea Pearl. “Causal inference in statistics: An overview”. In: *Statistics Surveys* 3 (2009), pp. 96–146 (cit. on p. 5).

[169]Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988 (cit. on p. 5).

[170]Virginia B. Penhune and Christopher J. Steele. *Parallel contributions of cerebellar, striatal and M1 mechanisms to motor sequence learning*. 2012 (cit. on pp. 8, 9).

[171]Pierre Perruchet and Annie Vinter. “Learning "what" and "how" about sequence”. In: *Trends in Cognitive Sciences* 12.12 (2008), pp. 576–582 (cit. on p. 24).

[172]Pierre Perruchet and Annie Vinter. “PARSER: A Model for Word Segmentation”. In: *Journal of Memory and Language* 39.2 (1998), pp. 246–263 (cit. on pp. 7–9, 18, 23).

[173]J. Piaget, ed. *The construction of reality in the child*. Basic Books, 1954 (cit. on p. 41).

[174]Jean Piaget. *The equilibration of cognitive structures: The central problem of intellectual development*. New translation of the development of thought: Child’s conception of geometry. Chicago: University of Chicago Press, 1985 (cit. on p. 44).

[175]Steven T. Piantadosi. “Zipf’s word frequency law in natural language: A critical review and future directions”. In: *Psychonomic Bulletin & Review* 21.5 (2014), pp. 1112–1130 (cit. on p. 54).

[176]Steven Pinker. *The Language Instinct: How the Mind Creates Language*. William Morrow and Company, 1994 (cit. on p. 48).

[177]Xaq Pitkow and Dora E. Angelaki. “Inference in the Brain: Statistics Flowing in Redundant Population Codes”. In: *Neuron* 94.5 (2017), pp. 943–953 (cit. on p. 5).

[178]Samuel Planton, Timo van Kerkoerle, Leïla Abbih, et al. “A theory of memory for binary sequences: Evidence for a mental compression algorithm in humans”. In: *PLOS Computational Biology* 17.1 (Jan. 2021), pp. 1–43 (cit. on pp. 7, 8).

[179]Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. “Grokking: Generalization beyond overfitting on small algorithmic datasets”. In: *arXiv preprint arXiv:2201.02177* (2022) (cit. on p. 49).

[180]M. Rabinovich, A. Volkovskii, P. Lecanda, et al. “Dynamical Encoding by Networks of Competing Neuron Groups: Winnerless Competition”. In: *Phys. Rev. Lett.* 87 (6 July 2001), p. 068102 (cit. on p. 24).

[181]Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. “Explain yourself! Leveraging language models for commonsense reasoning”. In: *arXiv preprint arXiv:1906.02361* (2019) (cit. on p. 3).

[182]RPN Rao. “A sensory–motor theory of the neocortex”. In: *Nature Neuroscience* 27.7 (2024), pp. 1221–1235 (cit. on p. 50).

[183]Keith Rayner, Timothy J Slattery, Denis Drieghe, and Simon P Liversedge. “Eye movements and the perceptual span in beginning and skilled readers”. In: *Journal of Experimental Psychology: Human Perception and Performance* 32.3 (2006), p. 516 (cit. on p. 19).

[184]Blake A. Richards and Paul W. Frankland. “The persistence and transience of memory”. In: *Neuron* 94.6 (2017), pp. 1071–1084 (cit. on p. 24).

[185]Edwin M Robertson. “The serial reaction time task: implicit motor skill learning?” In: *Journal of Neuroscience* 27.38 (2007), pp. 10073–10075 (cit. on p. 15).

[186]David A. Rosenbaum, Sandra B. Kenny, and Marcia A. Derr. “Hierarchical control of rapid movement sequences”. In: *Journal of Experimental Psychology: Human Perception and Performance* (1983) (cit. on pp. 8, 23).

[187]Joshua S. Rule, Joshua B. Tenenbaum, and Steven T. Piantadosi. “The Child as Hacker”. In: *Trends in Cognitive Sciences* 24.11 (Nov. 2020), pp. 900–915 (cit. on p. 29).

[188]Jenny R Saffran, Elizabeth K Johnson, Richard N Aslin, and Elissa L Newport. “Statistical learning of tone sequences by human infants and adults”. In: *Cognition* 70.1 (1999), pp. 27–52 (cit. on pp. 8, 24).

[189]Jenny R. Saffran, Elissa L. Newport, and Richard N. Aslin. “Word Segmentation: The Role of Distributional Cues”. In: *Journal of Memory and Language* 35.4 (1996), pp. 606–621 (cit. on pp. 8, 18, 24).

[190]Lorenza Saitta and Jean-Daniel Zucker. “A model of abstraction in visual perception”. In: *Applied Artificial Intelligence* 15.8 (Sept. 2001). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/088395101317018591, pp. 761–776 (cit. on p. 42).

[191]David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. *Analysing Mathematical Reasoning Abilities of Neural Models*. Tech. rep. arXiv:1904.01557. arXiv:1904.01557 [cs, stat] type: article. arXiv, Apr. 2019 (cit. on p. 3).

[192] Elizabeth R Schotter, Bernhard Angele, and Keith Rayner. “Parafoveal processing in reading”. In: *Attention, Perception, & Psychophysics* 74.1 (2011), pp. 5–35 (cit. on p. 19).

[193] Atilla Schreiber, Shuchen Wu, Chenxi Wu, Giacomo Indiveri, and Eric Schulz. “Biologically-plausible hierarchical chunking on mixed-signal neuromorphic hardware”. In: *Machine Learning with New Compute Paradigms*. 2023 (cit. on pp. 11, 55).

[194] Eric Schulz, Francisco Quiroga, and Samuel J Gershman. “Communicating compositional patterns”. In: *Open Mind* 4 (2020), pp. 25–39 (cit. on p. 19).

[195] Eric Schulz, Joshua B. Tenenbaum, David Duvenaud, Maarten Speekenbrink, and Samuel J. Gershman. “Compositional inductive biases in function learning”. In: *Cognitive Psychology* (2017) (cit. on pp. 19, 23).

[196] Philippe G. Schyns, Robert L. Goldstone, and Jean Pierre Thibaut. *The development of features in object concepts*. 1998 (cit. on pp. 8, 9).

[197] Emile Servan-Schreiber and John Anderson. “Learning Artificial Grammars With Competitive Chunking”. In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 16 (July 1990), pp. 592–608 (cit. on pp. 9, 17).

[198] Emile Servan-Schreiber and John R Anderson. “Learning artificial grammars with competitive chunking.” In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 16.4 (1990), p. 592 (cit. on pp. 15, 23).

[199] Hannah Sheahan, Floris Luyckx, Stefania Nelli, and Christopher Summerfield. “Structure learning drives active abstraction of hierarchical rules in human cognition”. In: *Cognition* 212 (2021), p. 104701 (cit. on p. 4).

[200] Hannah Sheahan, Floris Luyckx, and Christopher Summerfield. “Neural dynamics of hierarchical rule learning”. In: *Nature Communications* 13.1 (2022), p. 2175 (cit. on p. 4).

[201] Sara J Shettleworth. *Cognition, Evolution, and Behavior*. Oxford University Press, 2010 (cit. on p. 3).

[202] David Silver, Aja Huang, Christopher J. Maddison, et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529 (2016), pp. 484–503 (cit. on p. 3).

[203] Herbert A. Simon. “Invariants of Human Behavior”. In: *Annual Review of Psychology* 41 (Feb. 1990), pp. 1–20 (cit. on p. 4).

[204] B. F. Skinner. *Science and Human Behavior*. New York: Macmillan, 1953 (cit. on p. 6).

[205] Jake Snell, Kevin Swersky, and Richard S. Zemel. “Prototypical networks for few-shot learning”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017 (cit. on p. 42).

[206] Elizabeth S Spelke. “Principles of object perception”. In: *Cognitive Science* 14.1 (1990), pp. 29–56 (cit. on p. 3).

[207] Nathan Srebro. “Maximum Likelihood Bounded Tree-Width Markov Networks”. In: *CoRR* abs/1301.2311 (2013). arXiv: 1301.2311 (cit. on p. 5).

[208] Greg J Stephens, Bethany Johnson-Kerner, William Bialek, and William S Ryu. "Dimensionality and dynamics in the behavior of C. elegans". In: *PLoS Computational Biology* 4.4 (2008), e1000028 (cit. on p. 54).

[209] Francis Stevens, R.A. Hurley, and Katherine Taber. "Anterior cingulate cortex: Unique role in cognition and emotion". In: *Journal of Neuropsychiatry and Clinical Neurosciences* 23 (Jan. 2011), pp. 121–125 (cit. on p. 27).

[210] Christopher Summerfield and Floris P De Lange. "Perceptual classification in a rapidly changing environment". In: *Neuron* 103.2 (2019), pp. 227–236 (cit. on p. 4).

[211] Christopher Summerfield, Floris Luyckx, and Hannah Sheahan. "Deep learning models of the mind". In: *Neuron* 109.2 (2020), pp. 222–234 (cit. on p. 4).

[212] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. "How to grow a mind: Statistics, structure, and abstraction". In: *Science* 331 (2011), pp. 1279–1285 (cit. on p. 20).

[213] Joshua B. Tenenbaum, Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. "How to Grow a Mind: Statistics, Structure, and Abstraction". In: *Science* 331.6022 (Mar. 2011). Publisher: American Association for the Advancement of Science, pp. 1279–1285 (cit. on pp. 15, 42).

[214] H. S. Terrace. "Chunking by a pigeon in a serial learning task". In: *Nature* 325.6100 (1987) (cit. on p. 7).

[215] E. Thorndike et al. *Adult Learning*. New York: Macmillan, 1928 (cit. on p. 6).

[216] E. Thorndike. *Educational Psychology: The Psychology of Learning*. New York: Teachers College Press, 1913 (cit. on p. 6).

[217] E. Thorndike et al. *The Measurement of Intelligence*. New York: Teachers College Press, 1927 (cit. on p. 6).

[218] E. Thorndike. *The Psychology of Arithmetic*. New York: Macmillan, 1922 (cit. on p. 6).

[219] E. Thorndike. *The Teacher's Word Book*. New York: Teachers College, 1921 (cit. on p. 6).

[220] Nikolaas Tinbergen. *The study of instinct*. 1951 (cit. on p. 54).

[221] Michael Tomasello and Raquel Olguin. "Twenty-three-month-old children have a grammatical category of noun". en. In: *Cognitive Development* 8.4 (Oct. 1993), pp. 451–464 (cit. on p. 41).

[222] Momchil S. Tomov, Samyukta Yagati, Agni Kumar, Wanqian Yang, and Samuel J. Gershman. "Discovery of hierarchical representations for efficient planning". In: *PLoS computational biology* (2020) (cit. on pp. 8, 19).

[223] Hugo Touvron, Matthieu Cord, Matthijs Douze, et al. "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10347–10357 (cit. on p. 57).

[224] Endel Tulving. "Precis of elements of episodic memory". In: *Behavioral and Brain Sciences* 7.2 (1984), pp. 223–238 (cit. on p. 33).

[225] Ashish Vaswani et al. "Attention is All You Need". In: *Advances in Neural Information Processing Systems (NIPS)*. 2017, pp. 5998–6008 (cit. on p. 57).

[226] Alex P. Vaz, John H. Wittig, Sara K. Inati, and Kareem A. Zaghloul. “Replay of cortical spiking sequences during human memory retrieval”. In: *Science (New York, N.Y.)* 367.6482 (Mar. 2020), pp. 1131–1134 (cit. on p. 56).

[227] Willem B. Verwey. “Buffer Loading and Chunking in Sequential Keypressing”. In: *Journal of Experimental Psychology: Human Perception and Performance* 22.3 (1996) (cit. on pp. 8, 21).

[228] Sarah Vinette and Signe Bray. “Variation in functional connectivity along anterior-to-posterior intraparietal sulcus, and relationship with age across late childhood and adolescence”. In: *Developmental Cognitive Neuroscience* 31 (Apr. 2015) (cit. on p. 27).

[229] Andrew J Viterbi. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. In: *IEEE Transactions on Information Theory* 13.2 (1967), pp. 260–269 (cit. on p. 5).

[230] Jonathan D. Wallis, Kathleen C. Anderson, and Earl K. Miller. “Single neurons in prefrontal cortex encode abstract rules”. en. In: *Nature* 411.6840 (June 2001), pp. 953–956 (cit. on p. 56).

[231] Jane X Wang. “Meta-learning in natural and artificial intelligence”. In: *Current Opinion in Behavioral Sciences* 38 (2021). Computational cognitive neuroscience, pp. 90–95 (cit. on p. 42).

[232] Jue Wang, Ning Yang, Wei Liao, et al. “Dorsal anterior cingulate cortex in typically developing children: Laterality analysis”. In: *Developmental Cognitive Neuroscience* 15 (2015), pp. 117–129 (cit. on p. 27).

[233] Quan Wang, Constantin A. Rothkopf, and Jochen Triesch. “A model of human motor sequence learning explains facilitation and interference effects based on spike-timing dependent plasticity”. In: *PLoS computational biology* (2017) (cit. on pp. 18, 23).

[234] Wenhai Wang, Enze Xie, Xiang Li, et al. “Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions”. In: *arXiv preprint arXiv:2102.12122* (2021) (cit. on p. 57).

[235] Jason Wei, Yi Tay, Rishi Bommasani, et al. *Emergent Abilities of Large Language Models*. 2022. arXiv: 2206.07682 [cs.CL] (cit. on p. 49).

[236] Max Wertheimer. “Untersuchungen zur Lehre von der Gestalt, I: Prinzipielle Bemerkungen [Investigations in Gestalt theory: I. The general theoretical situation]”. In: *Psychologische Forschung* 1 (1922), pp. 47–58 (cit. on pp. 6, 22).

[237] Max Wertheimer. “Untersuchungen zur Lehre von der Gestalt, II. [Investigations in Gestalt Theory: II. Laws of organization in perceptual forms]”. In: *Psychologische Forschung* 4 (1923), pp. 301–350 (cit. on pp. 6, 22).

[238] Wayne A. Wickelgren. “Chunking and consolidation: A theoretical synthesis of semantic networks, configuring in conditioning, S-R versus cognitive learning, normal forgetting, the amnesic syndrome, and the hippocampal arousal system”. In: *Psychological Review* 86.1 (1979) (cit. on pp. 8, 9).

[239] Daniel B Willingham, Mary J Nissen, and Peter Bullemer. “On the development of procedural knowledge.” In: *Journal of experimental psychology: learning, memory, and cognition* 15.6 (1989), p. 1047 (cit. on p. 15).

[240] Matthew A. Wilson and Bruce L. McNaughton. “Reactivation of hippocampal ensemble memories during sleep”. In: *Science* 265 (1994), pp. 676–679 (cit. on p. 56).

[241] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, et al. “Mapping sub-second structure in mouse behavior”. In: *Neuron* 88.6 (2015), pp. 1121–1135 (cit. on p. 54).

[242] Terry Winograd. “Procedures as a representation for data in a computer program for understanding natural language”. In: *MIT Research Lab of Electronics (MAC-TR-84)* (1971) (cit. on p. 4).

[243] Patrick Henry Winston and Berthold Klaus Paul Horn. *Artificial intelligence*. Addison-Wesley Pub. Co., 1972 (cit. on p. 4).

[244] S. Wu, M. Yoerueten, F. A. Wichmann, and E. Schulz. “Normalized Cuts Characterize Visual Recognition Difficulty of Amorphous Image Sub-parts”. In: *Computational and Systems Neuroscience (COSYNE)*. Lisbon, Portugal, 2024 (cit. on p. 11).

[245] Shuchen Wu, Noemi Elteto, Ishita Dasgupta, and Eric Schulz. “Learning Structure from the Ground up—Hierarchical Representation Learning by Chunking”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, et al. Vol. 35. Curran Associates, Inc., 2022, pp. 36706–36721 (cit. on pp. 10, 30, 55).

[246] Shuchen Wu, Noémi Éltető, Ishita Dasgupta, and Eric Schulz. “Chunking as a rational solution to the speed–accuracy trade-off in a serial reaction time task”. In: *Scientific Reports* 13.1 (May 2023), p. 7680 (cit. on pp. 10, 20).

[247] Shuchen Wu, Mirko Thalmann, and Eric Schulz. “Building, Reusing, and Generalizing Abstract Representations from Concrete Sequences”. In: (May 2024) (cit. on pp. 11, 50).

[248] Shuchen Wu, Mirko Thalmann, and Eric Schulz. “Motif Learning Facilitates Sequence Memorization and Generalization”. In: (Dec. 2023) (cit. on pp. 10, 39).

[249] Fan Xia et al. “Temporal Dynamics of Neural Representations in Human Cortex”. In: *Nature Communications* 11 (2020), pp. 1–12 (cit. on p. 49).

[250] Charles D. Yang. “Universal Grammar, statistics or both?” en. In: *Trends in Cognitive Sciences* 8.10 (Oct. 2004), pp. 451–456 (cit. on pp. 8, 34).

[251] Guangyu Robert Yang, Mangesh R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. “Task representations in neural networks trained to perform many cognitive tasks”. In: *Nature Neuroscience* 22.2 (2019), pp. 297–306 (cit. on pp. 42, 43).

[252] Jonathan S Yedidia, William T Freeman, and Yair Weiss. “Constructing free-energy approximations and generalized belief propagation algorithms”. In: *IEEE Transactions on Information Theory*. Vol. 51. 7. 2005, pp. 2282–2312 (cit. on p. 5).

[253] Eiling Yee. “Abstraction and concepts: when, how, where, what and why?” In: *Language, Cognition and Neuroscience* 34.10 (Nov. 2019). Publisher: Routledge _eprint: https://doi.org/10.1080/23273798.2019.1660797, pp. 1257–1265 (cit. on p. 43).

[254] Lichen Zhang, Xiaoming Yang, Shuangjun Li, Wenzhe Liu, and Gim Hee Lee. “Interpretable human action recognition by learning bases of action attributes and parts”. In: *IEEE Transactions on Multimedia* 22.9 (2020), pp. 2334–2347 (cit. on p. 3).

[255] Sen Zhang, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Peng Li. “Symbolic mathematics reasoning in transformers”. In: *arXiv preprint arXiv:2109.15085* (2021) (cit. on p. 3).

[256] Song-Chun Zhu and Siyuan Huang. *Computer Vision: Stochastic Grammars for Parsing Objects, Scenes, and Events*. Springer, 2021 (cit. on pp. 4, 56).

[257] Song-Chun Zhu and Ying Nian Wu. *Computer Vision: Statistical Models for Marr’s Paradigm*. Springer, 2021 (cit. on p. 56).

[258] Song-Chun Zhu and Yixin Zhu. *Cognitive Models for Visual Commonsense Reasoning*. Springer, 2021 (cit. on p. 56).

# 8 Statement of Contribution

*This chapter encloses a statement of contributions provided to abide by the guidelines of the Graduate Training Center.*

The work reported in this thesis is entirely my own. All chapters have been directly supervised by my primary supervisor, Prof. Dr. Eric Schulz.

### Chunking as a rational solution to the speed-accuracy trade-off in a serial reaction time task

*Framework:* I authored this paper as the lead author as the first individual project of my PhD. This project was conducted under the main supervision of Prof. Dr. Eric Schulz (ES) and in collaboration with Noemi Éltető (NE) and Dr. Ishita Dasgupta (ID). I am the sole first author on this paper.

*Contributions:*

- I designed the experiments under the supervision of ES and suggestions from NE and ID. It was during the initiation phase of COVID-19 that we experienced some delays in obtaining ethical approvals.
- I programmed the experiments as an online study, piloted it and collected the data.
- I developed the mathematical models, implemented them in code and ran and analyzed simulations under the main supervision of ES. ES helped with the data analysis for experiment 2. ID and NE provided suggestions for data analysis.
- I wrote the entire initial paper and revision under the main supervision of ES. ES, ID, and NE provided suggestions and editing.

### Learning Structure from the Ground up — Hierarchical Representation Learning by Chunking

*Framework:* This paper was the second individual project of my PhD. I am the sole first author on this paper. This project was conducted under the main supervision of Prof. Dr. Eric Schulz (ES) and in collaboration with Noemi Éltető (NE) and Dr. Ishita Dasgupta (ID).

*Contributions:*

- I came up with the conceptualization of the paper, developed the model, and programmed the model and its generalization to higher dimensional sequences.
- NE and ID provided suggestions for the model comparison experiments. NE helped with the model comparison with PARSER and AL.
- ID suggested the transfer experiments.
- ES suggested the experiments on fMRI data.
- I wrote the entire initial paper under the main supervision of ES. ID, and NE provided suggestions and editing.
- I discussed extensively with ES to come up with experiments and analysis during revision. I programmed the experiments and evaluation. ID and NE provided additional suggestions for editing.

MOTIF LEARNING FACILITATES SEQUENCE MEMORIZATION AND GENERALIZATION

*Framework:* This paper was the third individual project of my PhD. I am the sole first author of this paper. This project was conducted under the main supervision of Prof. Dr. Eric Schulz (ES) and in collaboration with Dr. Mirko Thalmann (MT).

*Contributions:*

- I conceptualized the paper together with ES.
- I developed the model and programmed the model.
- I designed the experiments together with MT and ES. I programmed the experiments and conducted iterations of pilot studies with suggestions from MT and ES.
- I analyzed the data. MT and ES provided suggestions on data analysis.

- I wrote the entire initial paper, MT provided extensive editing, which went on several iterations with ES.

- I discussed extensively with ES and MT on additional analysis. I conducted additional analysis during the rebuttal. MT and ES provided suggestions and editing.

BUILDING, REUSING, AND GENERALIZING ABSTRACT REPRESENTATIONS FROM CONCRETE SEQUENCES

*Framework:* This paper was the fourth individual project of my PhD. I am the sole first author on this paper. This project was conducted under the supervision of Prof. Dr. Eric Schulz (ES) and in collaboration with Dr. Mirko Thalmann, Prof. Dr. Peter Dayan (PD) and Prof. Dr. Zeynep Akata (ZA).

*Contributions:*

- I conceptualized the paper, developed the algorithm and evaluation methods and programmed the model and experiments.

- PD suggested the comparison and discussion in relation to compression and models of compression.

- ES suggested the LLM experiments and the evaluation of language sequences.

- I wrote the entire initial paper. PD, and ES provided extensive suggestions and editing. MT, and ZA provided additional suggestions and editing.

This appendix includes the original papers and preprints discussed in this thesis.

1. Chunking as a rational solution to the speed–accuracy trade-off in a serial reaction time task (Wu, Élteto, Dasgupta, & Schulz, *Nature Scientific Reports* 13, 7680 (2023), doi:10.1038/s41598-023-31500-3)

2. Learning Structure from the Ground-up—Hierarchical Representation Learning by Chunking (Wu, Élteto, Dasgupta, & Schulz, 2022, *36th Conference on Neural Information Processing Systems (NeurIPS 2022)* 35, 36706 - 36721 (2022)).

3. Two Types of Motifs Enhance the Recall and Generalization of Long Sequences. Preprint (Wu, Thalmann, Schulz, 2023, *PsyArXiv* and has been accepted in *Nature Communications Psychology*, doi:10.31234/osf.io/2a49z)

4. Building, Reusing, and Generalizing Abstract Representations from Concrete Sequences (Wu, Thalmann, Dayan, Akata, & Schulz, 2024, is under review at *International Conference on Learning Representations*, doi:10.48550/arXiv/2410.21332)

# Part V

## Manuscripts

# Chunking as a rational solution to the speed-accuracy trade-off in a serial reaction time task

scientific reports

Check for updates

OPEN

# Chunking as a rational solution to the speed–accuracy trade-off in a serial reaction time task

Shuchen Wu[1]✉, Noémi Éltető[2], Ishita Dasgupta[3] & Eric Schulz[1]

**When exposed to perceptual and motor sequences, people are able to gradually identify patterns within and form a compact internal description of the sequence. One proposal of how sequences can be compressed is people's ability to form chunks. We study people's chunking behavior in a serial reaction time task. We relate chunk representation with sequence statistics and task demands, and propose a rational model of chunking that rearranges and concatenates its representation to jointly optimize for accuracy and speed. Our model predicts that participants should chunk more if chunks are indeed part of the generative model underlying a task and should, on average, learn longer chunks when optimizing for speed than optimizing for accuracy. We test these predictions in two experiments. In the first experiment, participants learn sequences with underlying chunks. In the second experiment, participants were instructed to act either as fast or as accurately as possible. The results of both experiments confirmed our model's predictions. Taken together, these results shed new light on the benefits of chunking and pave the way for future studies on step-wise representation learning in structured domains.**

William James famously said that we are born into a "blooming, buzzing confusion", and that we escape that confusion by gradually making sense of the series of events we perceive. How we perceive a sequence of perceptual stimuli, process them, and extract underlying structure, is a fundamental question of psychological investigations. One proposal of how the blooming, buzzing confusion of seemingly disparate sequential events can become one cognitive unit is chunking[1–4]. Upon exposure to sequential stimuli, humans and animals can identify repeated patterns and segment sequences into chunks of patterns[5]. To this end, separate sequential elements merge into one cognitive entity. This cognitive entity is then recalled and identified as a whole[6]: a phenomenon known as *chunking*[7,8].

Chunking is a phenomenon spanning across sequence learning, grammar learning, visual and working memory tasks, and function learning, among others[8–12]. The ability to discover statistical regularities in sequences, and to identify them as discrete, disparate units of chunks enables us to form a compact and compressed memory representation[13], readily transferable to novel domains[14], and enables us to progress from novices to experts[15,16]. As primitive building blocks of cognitive construction units, a complex and lengthy sequence reduces to several chunks. This property facilitates the organization of actions[7], and can subsequently help with compositionality in learning[17], communication of structure[18], hierarchical planning[19] and others. In short, chunking is a critical and universal learning phenomenon. Here we propose another benefit of chunking in sequential tasks: the ability to more easily predict future outcomes and thereby act faster. Thus, our work connects the literature of chunking with that of the speed-accuracy trade-off.

The speed-accuracy trade-off is observed both in humans and animals across various task domains[20]. When speed is emphasized, participants in both lab and naturalistic settings tend to make more mistakes while reacting faster than when accuracy is emphasized[21,22]. While earlier work focused on analyzing reaction times and accuracy[21,23], little work has been done to relate the speed-accuracy trade-off to chunking and examine it affects the process and outcome of learning representations.

The serial reaction time task (SRT), a classical paradigm to study motor sequence learning[12,24–26], is ideal for studying the speed-accuracy trade-off and chunking. In SRTs, sequences of instruction cues appear consecutively on the screen, after which participants react by pressing the corresponding key that maps to the cue. If particular patterns, for example, ABC, keep repeating, then grouping repeated chunks as a unit facilitates the prediction of upcoming sequences. The detection of a chunk's beginning, in this case, A, implies that the within-chunk items

[1]MPRG Computational Principles of Intelligence, Max Planck Institute for Biological Cybernetics, Tübingen, Germany. [2]Department of Computational Neuroscience, Max Planck Institute for Biological Cybernetics, Tübingen, Germany. [3]Google DeepMind, New York City, NY, USA. ✉email: shuchen.wu@tue.mpg.de

B and then C will follow. This anticipation of the following elements of a given chunk can allow participants to anticipate what is coming next and thereby react faster[12,14]. Chunking sequence elements, however, can also come at a cost when the sequence is probabilistic. By assuming deterministic transitions between the within-chunk items AB, participants might lose fine-grained statistical information about single-item instructions and thereby occasionally miss between-chunk transitions such as AC. This, in turn, can decrease their accuracy.

We propose a model that trades off between speed and accuracy when performing SRTs. Our model calculates the utility of acquired chunk representations as a weighted sum of how well they capture the statistical structure in the SRT (accuracy) and whether they permit faster responses (speed). Our model then iteratively decides whether or not to chunk consecutive items. This model makes two distinct predictions. First, in environments where deterministic chunks exist, adding them to the representation is beneficial because they speed up reaction times without losing accuracy. Thus, people should chunk more in environments with more or longer chunks. In our first experiment, we tested this prediction by training participants on sequences containing underlying chunks. We designed a couple of analysis methods to test and verified this prediction. The results of this experiment suggest that participants adapt their chunking behavior to the underlying chunks in the sequence when they are given universal instructions to act as fast and accurately as possible. A subsequent prediction following the first experiment from the model is that when participants are given distinct instructions to perform on the task, these instructions will induce distinct chunking behavior even when the sequence have the same underlying statistics for the two groups. Specifically, this will be a rational strategy for the model to learn chunks in cases where the underlying environment is non-deterministic and does not contain any chunks. As the utility of speed increases (at the cost of accuracy), participants might also chunk consecutive elements more often and learn longer chunks. Since chunking frequently co-occurring events improves reaction time at the cost of overall accuracy, chunking can be a rational strategy to act faster. We tested and verified this prediction in a second experiment by training participants on sequences generated from a first-order Markovian transition matrix with "illusory" chunks while instructing one group to focus on speed and the other group to focus on accuracy. The results of this second experiment suggest that the group focusing on speed chunked more than the group focusing on accuracy. The fast group learns more chunks and makes more mistakes. While the accurate group learns the underlying generative model of the sequence better, but smaller chunks than the fast group. Our results shed new light on the benefits of chunking under specific task instructions and pave the way for future studies on structural inference in statistical learning domains.

**Serial reaction time task.** We study chunking in a serial reaction time task (SRT, see Fig. 1b). Participants are instructed to press keys corresponding to a sequence of cues that appear on the screen. The instruction cross turns green after a correct keypress and red after an incorrect keypress. The subsequent trial starts after a 500ms response-to-stimulus interval. The task starts with two baseline blocks followed by six training blocks and ends with two test blocks. Each block consists of 100 trials. For both experiments, the same generative mechanism produces the baseline and the test blocks. To study whether participants' chunking behavior adapts to task demands in an SRT task, we manipulate various properties of the training blocks to examine how they affect behavior in the test block, using the baseline block as a comparison. The observed differences between the test and baseline blocks reflect the changes in representations elicited by the training blocks.

There are various approaches to generating sequences in an SRT paradigm. One type of instruction involves repeated sequence[26,27], while others avoid direct repetitions or runs such as 1234[28], where 1,2,3,4 refer to 4 targets on the computer screen. One probabilistic way of generating the sequence is the alternating serial reaction time task[29,30], where instruction patterns can be 1r4r3r2r, with r being a randomly chosen target. Other probabilistic ways of generating the presented sequences include choosing successive images according to a probabilistic first-order Markov transition process, specified by a conditional probability matrix[31]. Schvaneveldt and Gomez used two sequences, such as 1243 and 1342 and drew the target sequence via weighted coin flip results[32]. Several reasons have been put forward in the literature for using probabilistic transitions to generate SRT sequences. One is that probabilistic transitions allow continuous and flexible assessment of learning progression. Another one is that the probabilistic nature of the sequences allows for a larger variety of sequence chunks to be generated and learned[33,34].

In both of our experiments, the sequences in the baseline and test blocks are generated from a non-deterministic, first-order Markovian transition matrix between the four instruction keys. In particular, out of all 16 transitions specified between the four keys, the transitions from A to B and C to D are highly probable ($P$ = 0.9), and the transitions from B to C and from D to A are medium probable ($P$ = 0.7) (see Fig. 1c). In this way, participants often observe reoccurring sequence segments such as AB and CD and could possibly perceive them as "illusory" chunks, even though the generative model is nondeterministic first-order Markovian.

We manipulate the training block sequences across the two experiments. In Experiment 1, three groups of participants were trained on sequences containing no chunks (independent), chunk AB (size 2 chunk), or chunk ABC (size 3 chunk). In Experiment 2, the same "illusory" transition matrix generates the training block sequences but the instructions differ across the two experimental groups. One group is instructed to respond as accurately as possible, while the other is instructed to respond as fast as possible. In order to control for motor effects due to hand and finger dominance, the instructions "A", "B", "C", "D" are randomly mapped to the keys "D", "F", "J", "K" for individual participants. In the next section, we discuss the predictions of our rational model of chunking for the two different experiments and their conditions.

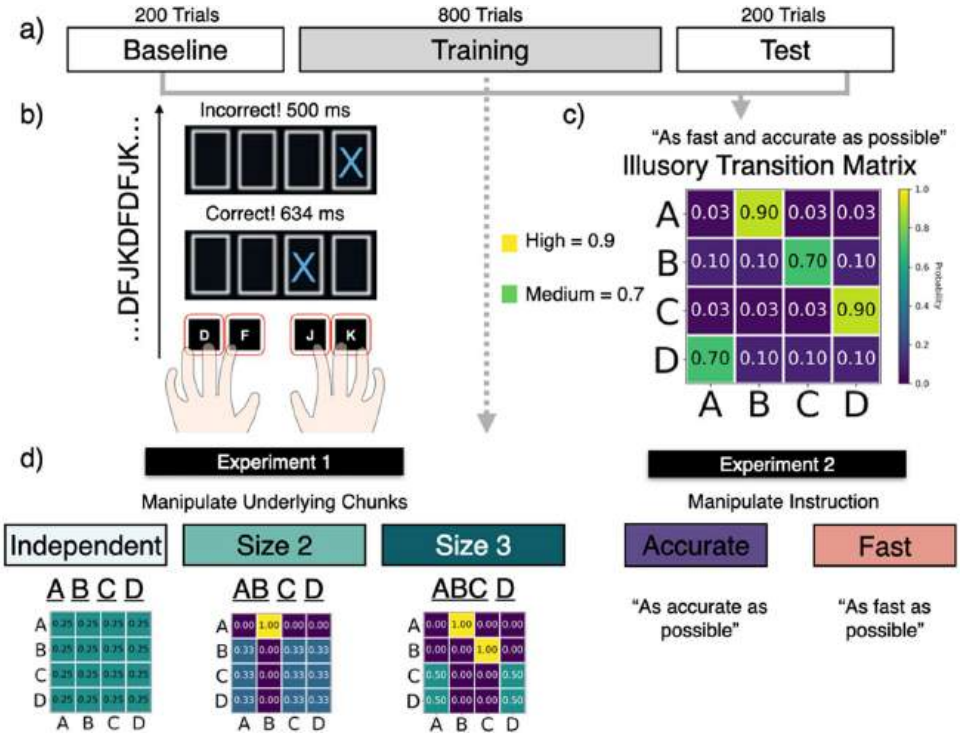


**Figure 1.** (**a**) Task structure for both experiments. Six training blocks are sandwiched between two baseline and two test blocks. The baseline and test blocks contain sequences generated from the "illusory" transition matrix in (**c**). (**b**) Participants are instructed to press the corresponding key on the keyboard according to trial-by-trial displayed instructions. They are given feedback on their performance, including accuracy and reaction times before the subsequent trial. (**c**) A non-deterministic, "illusory" transition matrix of the four possible key-presses is used to generate sequences for the baseline and test blocks for both experiments. The generative transition matrix with the two high (from A to B, C to D) and two medium transition probabilities (from B to C, D to A) produces "illusory" chunks that can be perceived as frequently occurring. To control the effect of habitual presses from consecutive fingers, a random mapping from "A", "B", "C", "D" to "D", "F", "J", "K", is generated independently for each participant. (**d**) The instructions for training blocks differed between the two experiments and corresponding groups. In Experiment 1, participants were divided into three groups who learned independent, size 2, and size 3 chunks from a predefined set of chunks with equal probability. In Experiment 2, the sequences in the training blocks were also generated from the "illusory" transition matrix. One group was instructed to act as accurately as possible and the other groups was instructed to act as fast as possible.

## Related work

Three major types of chunking models have been proposed in the cognitive science literature. The first type are symbolic models, including PARSER and CCN (competitive chunker)[35,36]. Symbolic models learn chunks from already-encountered items and constructs a hierarchy of chunks as participants remember sentences. Sevan-Schreiber and Anderson showed that these models can replicate the behavior of participants' judgment of grammaticality from sequences with distinct hierarchy levels (e.g., word level vs. phrase level)[36]. Additionally, they replicate the participants' tendency to overtly chunk the training sentences even when they are presented in an unstructured way. Another model of this kind is PARSER[35]. Proposed by Perruchet and Vinter, PARSER randomly samples the size of the next chunk of syllables and parses the sequence by disjunctive chunks. Each chunk learned by the model is associated with a weight, which increments with observational frequency and decrements via a forgetting mechanism. PARSER can produce artificial language stream segmentations of continuous input streams without episodic cues such as pauses.

The second type are connectionist models of chunk learning. This includes TRACX[37] and SRN (simple recurrent network)[38]. TRACX uses a three-layer feedforward backpropagation autoassociator and adapts the autoassociator's weights to the difference between its prediction and actual sequential units when this difference exceeds a pre-defined threshold. Wang et al. trained a self-organized recurrent spiking neural network with spike-timing-dependent plasticity and homeostatic plasticity on sequences. The model was shown to reproduce several sequence learning effects[39].

The two model types mentioned above are process models. In contrast to process models stand normative statistical models, which model the ideal observers' behavior. This approach includes variants of the Bayesian ideal observer framework[40]. Given a linguistic corpus, these models find a segmentation with the highest probability that contains relatively few word types, exploiting the minimal description length principle. These

models are also rational because their inference is evaluated on observational instances. They provide accounts for high-level computation required for chunk learning. As normative and process models rely on different principles, they are usually not compared against each other.

While these models focused on the benefit of chunking in memory compressibility and grammaticality sensibility, little work relates task instruction with the chunks acquired during learning. Since sequence statistics was the main guidance for chunk learning in these models, instruction modulation has rarely been taken into account.

One typical instruction that changes participants learning behavior is to focus on either speed or accuracy[20]. When task instructions emphasize speed, participants in both lab and naturalistic settings tend to make more mistakes while reacting faster than when instructions emphasize accuracy[21,22]. While earlier work focused on reaction time and accuracy of decision-making tasks[21,23], little work relates chunking to the speed-accuracy trade-off. It is unclear how instruction will affect chunking and what type of models can take this particular aspect of the task into account.

Here we propose a rational chunking model that takes sequence statistics and task instruction as two parts of a utility function for learning. The model tries to find rational ways of chunking the sequence under task demands, trading off speed with accuracy. Each aspect of the utility function implies a specific prediction: the same task instruction but different sequence statistics should lead to distinct chunking behavior; the same sequence statistics but different instructions should also lead to differently learned chunks. We propose two experiments to test the two aspects of our model's predictions. For the first experiment, we look at the case when three groups of participants learn from sequences with varying underlying chunks, how chunking changes with varying underlying sequential statistics with different embedded chunks, and show that our model captures participants' learned chunks. For the second experiment, we look at how participants' behavior differs when the task instructions focus separately on speed versus accuracy with the same underlying sequence. In doing so, we also propose several novel ways of analyzing RT data based on the speed-up of reaction time, thereby giving insights into how chunks build up across practice trials.

## A rational model of chunking

In the SRT, single instructions $z$ out of an instruction set $Z$ are presented sequentially. We told participants to press the corresponding key as soon as a new instruction appears. The subsequent instruction shows up in a fixed interval after a participant's completion of the previous trial. The model learns a set of chunks $C = \{c_1, ..., c_n\}$ and uses the set to parse the sequence. It evaluates the probability $P(c)$ of parsing each chunk $c$ and the conditional probability $P(c_j|c_i)$ that $c_j$ follows $c_i$ for every pair of chunks.

The set of chunks $C$ is initialized as the set of available single instructions $Z$ at the beginning of all simulations. The model updates this set by potentially concatenating existing pairs of chunks in $C$. Adding a chunk expands the parsing horizon as the rest of the within-chunk items are predicted to deterministically follow the initiation item of the chunk. Therefore, the subsequent within-chunk items are anticipated in the following trials. The model's accuracy might diminish if the subsequent instructions are inconsistent with the predicted within-chunk items. We relate subsequent item predictions to reaction times in the next section, and then explain the process by which a rational model updates chunks based on the trade-off between reaction times and accuracy.

**Accounting for reaction times.** We use a linear ballistic accumulator (LBA) model to simulate reaction times (RT). LBAs are a common class of multi-choice models[41,42]. In the LBA, each choice corresponds to an evidence accumulator, translated to each four possible key-presses in our task. At every trial of the SRT task, each evidence accumulator starts with an initial evidence $k = \log(P(z_i))$, which reflects the model's prediction on the upcoming instructions. The trials are divided into within-chunk trials and between-chunk trials. For a within-chunk trial, the prediction for the within-chunk item is the initial evidence for the accumulator $\log(1)$, the rest being $\log(\epsilon)$. Note that the model still integrates information from the SRT instructions but with a high offset which biases it to choose the response which is consistent with the chunk, even if it is inconsistent with the instructed item. This term encourages the model to create longer chunks to reduce the average reaction time.

For a between-chunk trial, the initial evidence for each accumulator $z_i$ is determined by the transition probability $P(c_i|c_j)$ of the chunk $c_i$ that initiates with the accumulator $z_i$, given the previously parsed chunk $c_j$. All response accumulators start from the initial evidence, and drift towards the decision threshold with positive drift rates $v_A, v_B, v_C, v_D$ sampled from a normal distribution with mean $v_{instruction}$ and standard deviation $\sigma$. To simulate the Rt of a particular trial, the current instruction carries the highest drift rate $v_{instruction} = 0.5$ and the evidence accumulators corresponding to the other instructions have an equal but lower drift rate $v_{\neg instruction} = \frac{1-v_{instruction}}{3}$. The drift rates for all accumulators sum up to 1. For example, if the current instruction is $A$, then $v_A = 0.5$, $v_B = v_C = v_D = \frac{0.5}{3}$. Evidence accumulation terminates when a positive response threshold $b$ is first crossed by any accumulator. The accumulator that crosses the decision threshold first becomes the overt response, and the time it takes to reach the decision threshold is the simulated RT on that trial. In all of the model simulations, we use the same $v_{instruction} = 0.5$, decision threshold $b = 1$, $\epsilon = 0.01$, and standard deviation $\sigma = 0.03$ across all accumulators.

**Balancing speed and accuracy.** We assume that chunking enables participants to predict upcoming instructions further into the future and thereby to react faster by initializing their evidence at a higher starting point. However, chunking also bears a risk of making mistakes when the upcoming instructions are not the subsequent items within a chunk. We formulate this speed-accuracy trade-off using the loss function

$$\mathscr{L} = wRt + (1 - w)Err, \tag{1}$$

where $Rt$ is the average reaction time in the SRT, given a learned chunk representation and sequence, and $Err$ is the average error rate. $w$ is a parameter that specifies the trade-off between accuracy and reaction time. When $w = 0$, only the reaction time term $Rt$ occupies the loss, and when $w = 1$, the error term $Err$ dominates.

Based on the LBA reaction time simulation, the average reaction time of parsing chunk $c_j$ after previously having parsed the chunk $c_i$ is $\frac{rt_{between}(c_j,c_i)+(|c_j|-1)rt_{within}}{|c_j|}$. $|c_j|$ is the length of the chunk. The reaction time on the first item is denoted as $rt_{between}(c_j, c_i)$, since $P(c_j|c_i)$ influences the evidence accumulation for this between-chunk key press and only the boundary of the chunk contains transition uncertainty and contributes to the slow down of reaction times. As the initial chunk item determines the chunk identification, the subsequent reaction time to press within-chunk keys in $c_i$ is denoted as $rt_{within}$. This term does not depend on $c_i$ as the procession to the within-chunk items contains no uncertainty. Taken together, the average reaction time can be formulated as follow, averaging the probability of parsing each acquired chunk $c_j$ given the previously parsed chunk $c_i$

$$Rt = \sum_{c_i \in C} P_C(c_i) \sum_{c_j \in C} P_C(c_j|c_i) \left[ \frac{rt_{between}(c_j, c_i) + (|c_j| - 1)rt_{within}}{|c_j|} \right], \tag{2}$$

Similarly, if we formulate $R_{LBA}(z_j)$ as the response choice of the LBA model when the instruction is $z_j$, then we can denote an error occurrence as $\mathbb{1}\left[z_j \neq response(z_j)\right]$, which is an indicator function that becomes 1 when the instruction $z_j$ is inconsistent with the LBA response. The average error rate can be evaluated by averaging the error rate with the probability of single-element transitions from the generative model $P_I$, enabling the formulation of the expected error rate as

$$\text{Err} = \sum_{z_i \in \{A,B,C,D\}} P_I(z_i) \sum_{z_j \in \{A,B,C,D\}} P_I(z_j|z_i) \mathbb{1}\left[z_j \neq R_{LBA}(z_j)\right] \tag{3}$$

This utility function, therefore, induces a trade-off between being accurate (predicting elements correctly) and being fast (finding a chunk representation to predict further ahead and speed up one's reaction time). Together, these parts of the loss are used to evaluate the utility of a chunk representation under specific task demands.

**The rational update of chunking.** The model updates the chunk representation rationally by concatenating chunks within the chunk set $C$ that induce a lower loss. $C$ is initialized with single sequential items $\{A, B, C, D\}$. For one set of chunks $C$, the model evaluates the marginal probabilities of each chunk $P_C(c_i), c_i \in C$ and the transition probability $P_C(c_j|c_i)$ of parsing chunk $c_j$ after having parsed chunk $c_i$. $P_C(c_i)$ and $P_C(c_j|c_i)$ are stored in the marginal and transition probability matrices as shown in Fig. 2b. The marginal and transition probability is evaluated empirically over an entire sequence parse using chunks in $C$.

We can calculate the joint occurrence probability of concatenating chunk $c_i$ with $c_j$ as $P(c_i, c_j) = P_C(c_j|c_i)P_C(c_i)$. The chunk pair $c_i$, $c_j$ with the highest joint probability is suggested as a new chunk to replace $c_i$ to form a new to the set of chunks $C_{new}$. As the initiation of $c_i$ is predictive of the subsequent chunk items. For example, an addition to $\{A, B, C, D\}$ could be a new chunk $AB$. The new chunk $AB$ then replaces $A$ and the new proposed set of chunks $C_{new}$ becomes $\{AB, B, C, D\}$.

We then compute whether $C_{new}$ is accepted to replace the original set of chunks $C$. The acceptance depends on whether the new set of chunks $C_{new}$ and the induced reaction time in addition to the marginal and transition probabilities upon parsing the sequence lead to a lower loss. In case it is so, $C_{new}$ replaces $C$, which becomes the basis of proposing the next chunk. This chunk proposal process continues until a fixed iteration number, as shown in Fig. 2a.

**Model predictions for experiment 1.** We first examined the model's chunk learning behavior on the three groups of Experiment 1. In this simulation, the underlying generative model either contained no chunks (independent), the chunk AB (size 2 chunks), or the chunk ABC (size 3 chunks). We then fixed the trade-off parameter $w$ to optimize accuracy more than speed by setting it to $w = 0.2$. Figure 2c shows the probability of chunk AB, BC, and ABC being learned as subchunks by the rational model of chunking over 120 simulations in total. The model uses the entire sequence to learn its chunk representation in each simulation. With the same trade-off between speed and accuracy, the rational chunking model trained on sequences with size 2 and 3 chunks has a higher probability of learning AB as a subchunk than a model trained on the independent sequence. Chunk BC has a higher probability of being learned by the model trained on sequences containing size 3 chunks than models trained on sequences with independent instructions or sequences with size 2 chunks. Only the model trained on sequences containing size 3 chunks learned about the chunk ABC. Taken together, these simulations predict that participants in the different conditions will be more likely to learn the corresponding chunks than participants for whom a chunk is not part of the training sequence.

**Model predictions for experiment 2.** We examined the model's chunk learning behavior for Experiment 2. According to our model, changing the trade-off between accuracy and speed translates to changing the cost function's $w$ away from 0 and towards 1. We therefore simulated the behavior of our model with changing $w$ (Fig. 2d). As $w$ goes from 0 to 1, i.e. the cost function shifts from minimizing the model's error rate to minimizing its reaction time, the average length of chunks learned by the model increases. Thus, our model predicts that participants in the fast group, which demands speedier responses, should learn longer chunks as compared

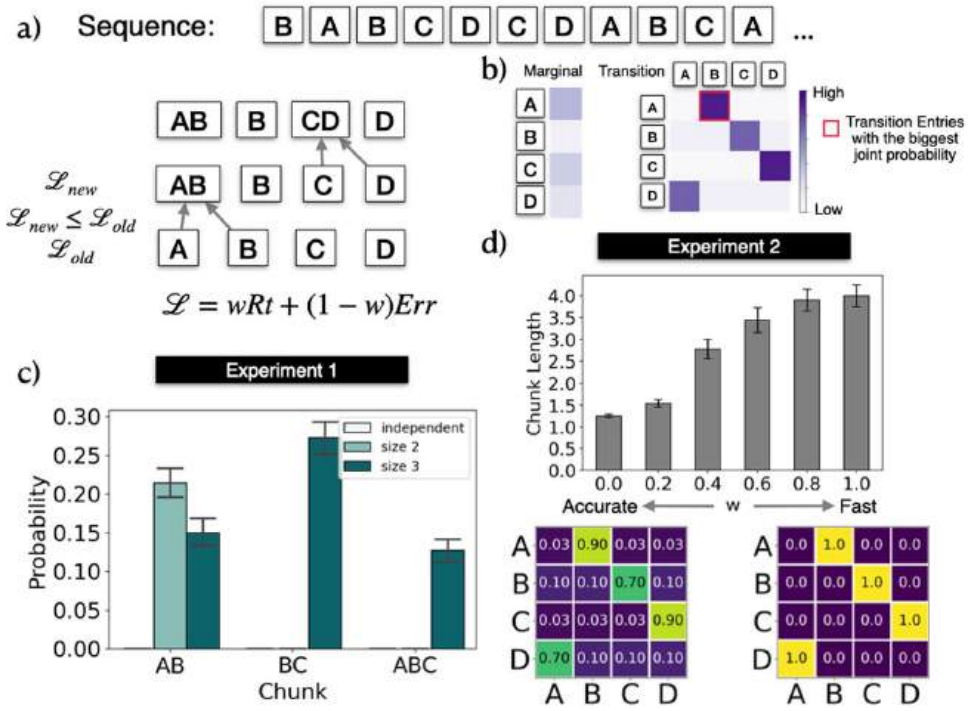


**Figure 2.** (**a**) Chunking mechanism of rational model. The model keeps track of marginal and transitional probabilities among every pair of pre-existing chunks, and combines chunk pairs that yield the greatest joint probability as the next candidate to be chunked together. At the start, the four different keys are initialized to be the primitive chunks. A loss function that trades off reaction times and accuracy is evaluated on the pre-existing set of chunks. If a chunk update reduces the loss function, then the two pre-existing chunks are combined together. A parameter $w$ determines how much more the model weighs an decrease of reaction times compared to an increase in accuracy. (**b**) Example model simulations of learning sequences of Experiment 1. A, B, C, D, are randomly mapped to D, F, J, K for individual participants. Because the transition AB occurred frequently, the model proposes this transition as a possible chunk. (**c**) Model simulation for Experiment 1. Bars represent the probability of a particular chunk parsed in a simulation over the whole experiment. The bars for the independent group on chunk AB, the independent and the size 2 group on chunk BC, and the independent and size 2 group on chunk ABC contain the probability of 0 and are therefore not visible in the graph. Note that these bars can be arbitrarily increased by changing $w$ while the qualitative results remain the same. (**d**) Model simulation for Experiment 2. Top: Average chunk length of different simulations when increasing $w$ from 0 (optimizing only accuracy) to 1 (optimizing only speed). As $w$ increases the average chunk length increases, indicating that the model learns longer chunks when asked to care more about acting fast. Bottom: Transition probabilities learned by model with $w = 0$ and $w = 1$, corresponding to the rational maximization of accuracy and speed. If the model tries to act as accurately as possible, then it recovers the true transition probabilities of the "illusory" transition matrix. If the model tries to act as fast as possible, then sets the medium and high transition probabilities to be 1, i.e. deterministic. All results are averaged across 120 independent simulations. Error bars represent the standard error of the mean.

to participants in the accurate group. Evaluating the single-element transition probability with $w = 0$ and 1 (Fig. 2d) shows that if only accuracy is the optimization goal of the cost function, then the model preserves the original transition matrix. However, if the model optimizes for speed, then it learns a polarized transition probability where all the high and medium single element transitions attract more probability mass, i.e. are closer to 1. Correspondingly, the remaining probabilities are closer to 0. Thus, as the high and medium transitions are more integrated into the chunks, this gives the model a speed-up in its reaction times, because it can start its evidence accumulation at a higher initial point. This comes at the cost of accuracy, because the initialization may be incorrect.

## Experiment 1: learning about true chunks

In Experiment 1, we test the model's prediction that chunking behavior adapts to the statistics of the sequence. When chunks are used to generate the sequence, participants should learn more than those trained on sequences without chunks.

Experiment 1 was conducted using a between-groups designs in which 122 participants were randomly assigned to one of three groups at the beginning of the experiment. These groups were the independent, size 2, and and size 3 conditions. The experiment was comprised of 10 blocks in total. The middle six blocks were the training blocks where participants practised the independent, size 2 or size 3 sequences. The first two and

the last two blocks were the baseline and test blocks. In those blocks, all three groups of participants received the same sequence generated from the "illusory" transition matrix. Training blocks differed amongst the three groups, as shown in Fig. 1, while the baseline and the test blocks remained the same. For the training blocks, the independent group practiced sequences that contained no chunks, the size 2 group practiced sequences with chunk AB, and the size 3 group practiced sequences with chunk ABC, as shown in Fig. 1d. The sequence for the independent group was randomly and independently sampled from single-item elements A, B, C, and D with equal probability. This means that this sequence contained no chunks. The sequence for the size 2 group was generated by sampling AB, C, and D with an equal probability of 1/3. In other words, this sequence contained the chunk AB. The sequence for the size 3 was generated by sampling ABC and D with an equal probability of 1/2. Thus, this sequence contained the chunk ABC. All three groups received the same instruction to act "as fast and accurately as possible" throughout the experiment. On each trial, participants received feedback on the reaction time and correctness of the previous trial, followed by a 500ms response-to-stimulus interval. Participants were informed that their performance bonus on top of a base-pay was based on a mixture of their reaction times and accuracy. The baseline and test blocks were sequences generated by the illusory transition matrix in Fig. 1c. The main prediction was that if people have learned chunks present in the training blocks, then they will use them even in the test blocks. We measured this by examining differences in accuracy and reaction times from the baseline block. We also used Experiment 1 to validate several of our empirical measures of chunking which we will use in Experiment 2.

We decided to examine model prediction on multifaceted prospects of participants' chunk learning behavior by proposing and applying various chunk learning measures at distinct stages of the task. Since many aspects of the measure are novel, we conduct these measures in the hope that their results complement each other.

The first two measures, chunky boost, and chunkiness, evaluate indicators of learning size 2 and size 3 chunks by comparing the performance of the baseline and test blocks. The regression on chunky RT evaluated on the test block examines the transition probability's influence on reaction time. The last three measures rely on chunks as identified by the mixture of the Gaussian method. They are directly-measured from the chunking profile of the participants. The chunk growth rate evaluates chunk size increase during training. The chunk increase measure shows the quantitative differences between counted chunks between the baseline and the test blocks. The last measure on chunk reuse probability looks for the character of reusing previously learned chunks to construct new chunks, as demonstrated by participants' chunk learning.

**Manipulation check.** We first checked if participants' behavior during the training blocks reflected the underlying chunks in the generative model. In particular, we tested whether the size 2 group showed evidence for learning chunk AB, and the size 3 group learning chunk ABC. We used a Gaussian mixture model to categorize reaction times of each response from the same participant into fast "within" or slow "between" chunk transitions, based on the assumption of a within-chunk speed-up. This method gave us a glimpse into how the action sequence was partitioned by participants, reflecting their internal representation of chunks (more details in Methods). We then counted the number of times chunks AB and ABC showed up in the training block, denoted as $N_{AB}$ and $N_{ABC}$. If the size 2 group and the size 3 group had learned chunk AB and ABC, separately, then $N_{AB}$ should be higher for these two groups than the independent group, and $N_{ABC}$ should be higher for size 3 group than the other two. Figure 3a shows the average $N_{AB}$ and $N_{ABC}$ returned by this analysis across the three conditions during the training blocks.

For $N_{AB}$, fitting a linear regression model using condition as the independent variable and the number of chunks $N_{AB}$ as the dependent variable showed a significant effect of condition ($F(2) = 45.02$, $p < 0.001$). $N_{AB}$ was higher for both the size 2 group ($\hat{\beta} = 59.43$, $t(139) = 8.20$, $p < 0.001$) and the size 3 group ($\hat{\beta} = 59.39$, $t(139) = 8.32$, $p < 0.001$) than for the independent group. This means that training on sequences that contained either AB or ABC chunks induced participants to learn AB as a chunk.

To investigate differences in the acquisition of the ABC chunk between groups, we repeated the same regression with $N_{ABC}$ as the dependent variable. We found a significant effect of groups ($F(2) = 71.45$, $p < 0.001$), indicating that participants' responses reflected ABC chunks more often in both the size 2 ($\hat{\beta} = 9.40, t(139) = 1.89$, $p = 0.06$) and size 3 ($\hat{\beta} = 54.17, t(139) = 11.07$, $p < 0.001$) than in the independent group. Interestingly, we observed higher $N_{ABC}$ with the size 2 group than in independent group. This can be because building on top of a previously learned chunk (AB → ABC) is more accessible for the size 2 group than the independent group (as the independent group needs to learn chunk AB first, then ABC). Furthermore, $N_{ABC}$ was significantly higher for the size 3 than the size 2 group ($\hat{\beta} = 44.76, t(139) = 9.25$, $p < 0.001$), suggesting that training on sequences that contained ABC chunks resulted in the strongest tendency of participants to learn ABC as a chunk.

Given these results, we conclude that our experimental manipulation of the three groups induced the intended behavior during the training blocks.

**Chunky boost.** When trained on sequences with underlying chunks ABC and BC, the rational chunking model learns chunk ABC and BC separately. To check this prediction of our model, we look at participants learning of size 2 chunks, i.e. AB and BC, separately. In particular, we look at participants' reaction time of pressing within chunk items, B in AB and C in BC, and how these items speed up differently across the three groups from the baseline to the test blocks. In SRT tasks, the reaction time difference before and after training is usually used as a sensitive measure of skill[25]. If participants' behavior is consistent with our model's prediction, then the size 2 and size 3 groups should have a stronger sign of learning chunk AB than the independent group. The size 3 group should have a stronger sign of learning chunk BC than the size 2 group.

We look at how the training schedule changes the value of within-chunk (value marked by the red boundary) reaction time for AB and BC (since a sign of chunking is that the reaction time of within-chunk items is

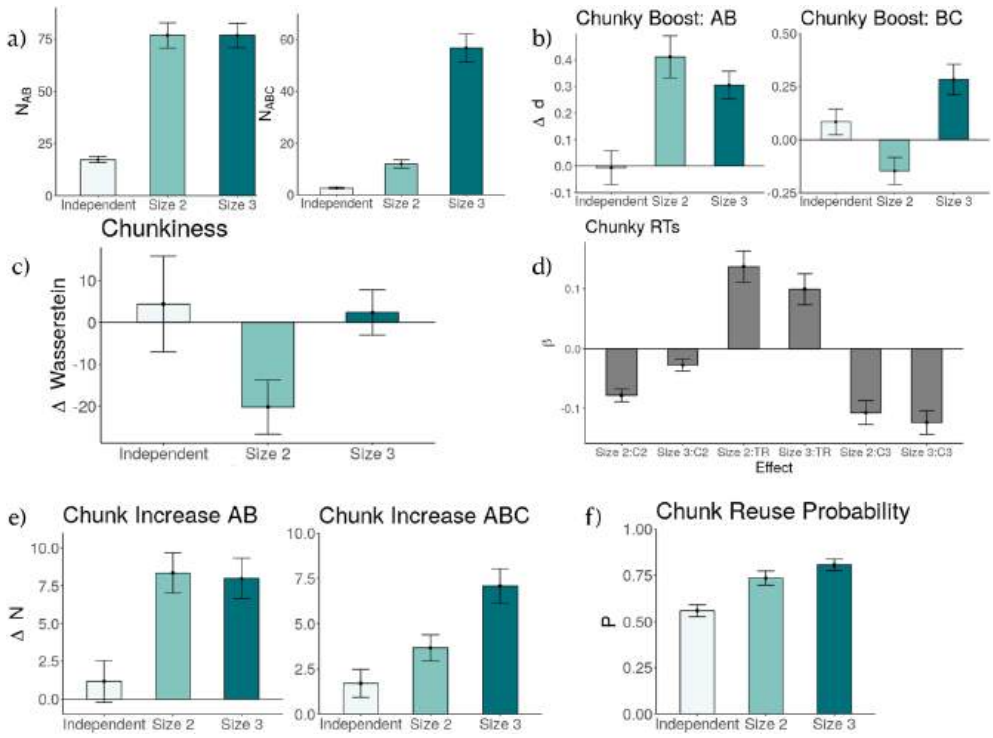


**Figure 3.** Results of Experiment 1. (**a**) Manipulation check. The number of chunks AB and ABC learned by participants during the training blocks by group. Chunks were retrieved using a categorization of between- and within-chunk transitions by a mixture of Gaussians analysis of participants' reaction times. (**b**) Chunky Boost of size 2 chunks AB and BC by group. A chunky boost is measured by the relative change of Cohen's *d* between baseline and test blocks for the highly and medium probable transitions. (**c**) Chunkiness of size-3 chunks ABC. Chunkiness is measured by the relative change of Wasserstein distance between the baseline and test blocks of between-chunk reaction times of all possible size 3 chunks. (**d**) Regression coefficients of interaction effects between condition and size 2, size 3, and true transition probabilities on reaction times during the final test blocks. (**e**) Chunk increase from the baseline to the test blocks by group for chunk AB and chunk ABC. Chunk increase is measured by the number of returned chunks from the mixture of Gaussians analysis. (**f**) Chunk reuse probability by group. Chunk reuse probability was calculated based on whether or not part of an earlier chunk were used in a later chunk that occurred within the next 30 trials. For all plots, error bars indicate the standard error of the mean.

typically faster than between-chunk items[8,12,43]); a figurative explanation of this method can be found in Fig. 6 in the appendix. We look at the within-chunk reaction time of AB and BC for all groups at the baseline and the test blocks and compute the difference by the signed effect size, Cohen's d, of the baseline blocks, compared to the test block $d_{AB}$. Cohen's d is a standardized measure of how far the means of two probability distributions are apart. In this case, these two distributions are the reaction time in the baseline blocks and the reaction time in the test blocks. We used a signed version of Cohen's d to convey the relative change of the reaction time distributions. $d_{AB}$ is positive when, on average, the reaction time of B in AB at the test block is faster than the reaction time in the training block – a sign of learning. However, solely looking at AB and BC is not enough, as a general learning factor will speed up participants' reaction time naturally. Therefore we compared the signed effect size AB and BC with the reaction time speed up of the control chunks. For the controlled between-chunk items, we evaluated the signed Cohen's d on AA, AB, and AC for chunk AB; and on BA, BB, and BD for chunk BC. Finally, we arrived at the chunky boost measure $\Delta d$ by subtracting the relative speed-up of AB and BC from their corresponding control chunks. We named this a chunky boost measure. Figure 3b shows the Chunky Boost of AB and BC across the three groups.

For chunk AB, fitting a linear model onto participants signed Cohen's d change showed a significant effect of group ($F(2) = 10.613$, $p < 0.001$); participants in the size 2 group had a higher relative change of Cohen's d than the independent group ($\hat{\beta} = 0.41, t = 4.41$, $p < 0.001$). Thus, training on the chunks with size 2 made the size 2 group respond to B faster after having seen item A. Additionally, participants in the size 3 group also had a higher relative change of reaction times responding to chunk AB than the independent group ($\hat{\beta} = 0.31, t = 3.35$, $p = 0.001$), showing that their reaction to B also sped up relative to control. These results are consistent with the model prediction that chunk AB should be acquired by the size 2 and size 3 group, separately.

For chunk BC, fitting a linear model onto the chunky boost measure $\Delta d$ on BC with group as the independent variable also showed a significant effect ($F(2) = 10.802$, $p < 0.001$). Interestingly, the size 2 group had a negative chunky boost to BC ($\hat{\beta} = -0.23, t = -2.44$, $p = 0.02$), showing a relative reaction time slow-down compared

to control. This effect was expected because identifying B as the end of a chunk will result in the transition to C as a "between-chunk" transition. In other previous SRT experiments, a slow-down in between-chunk reaction times was also observed[44]. This slow-down can contribute to the negative chunky boost of the size 2 group. Relative to the independent group, the size 3 group had a significantly higher chunky boost $\Delta d$ ($\hat{\beta} = 0.20, t = 2.14$, $p = 0.03$). This shows that learning chunks changes the reaction time profile of this group. Their response to C upon previous instruction B was speeding up their reaction times much more from the training blocks to the test blocks compared to control. This is consistent with the model prediction that the size 3 group should be more likely to learn chunk ABC.

In summary, participants' reaction times changed in a predictable fashion, with the independent group not getting faster for either AB or BC, the size 2 group becoming faster for AB and slower for BC, and the size 3 group becoming faster for both AB and BC. These observations confirmed previous work studying chunking in SRT tasks, which has argued that RTs in structured sequences decrease more quickly than in non-structured sequences[12] and are consistent with the predictions from the rational chunking model.

**Chunkiness.** The rational chunking model predicts that the size 2 group should learn more chunks AB, and the size 3 group should learn more chunks ABC, compared to the independent group. To access this prediction, we formulated a measure of chunkiness as an indicator of learning size 3 chunks. If participants have learned a size 3 chunk, such as ABC, then the distributions of within-chunk reaction times (i.e. the reaction time of B and C) should become more similar to each other[8]. We use the Wasserstein distance to evaluate the homogeneity of reaction time distribution of B and C, $rt_B$ and $rt_C$, following the presentation of A. The Wasserstein distance is also known as the "earth mover's" distance. It can be seen as the minimum amount of "work" required to transform one distribution into another. "Work" is the amount of distributional weight that must be moved multiplied by the distance (see also Supporting Information). This is simply just measuring how similar the two reaction time distributions are.

We evaluated the Wasserstein distance between the distribution of $rt_B$ and $rt_C$ on the baseline blocks, when all groups of participants are trained on the illusory transition sequences, to arrive at $Wasserstein(rt_B, rt_C)_{baseline}$. This assesses the initial separation of the two distributions, how participants learn from the illusory transition sequences, when all groups of participants have not been exposed to any training that involves chunks. Then we evaluate the same Wasserstein distance in the test blocks, also during the illusory transition sequence, to arrive at $Wasserstein(rt_B, rt_C)_{test}$, to assess how much training influences $rt_B$ and $rt_C$ in the test blocks. If participants have learned ABC as a chunk during the training blocks, then $rt_B$ and $rt_C$ should become more homogeneous in the test blocks, resulting in a smaller Wasserstein distance, as compared to the baseline blocks. We subtracted $Wasserstein(rt_B, rt_C)_{test}$ from $Wasserstein(rt_B, rt_C)_{baseline}$ to calculate this change of reaction time homogeneity: $\Delta W_{chunk}$. This relative change of Wasserstein should be positive if reaction times became more homogeneous in the test blocks. Since training may result in an overall increase of reaction time homogeneity for all items, we compared this change of *Wasserstein* with size 3 sub-sequences that were not ABC, as a control. $\Delta W_{control}$ as the difference between $W_{test}$ and $W_{train}$ is evaluated on the control sequences.

Finally, we subtract $\Delta W_{control}$ from $\Delta W_{chunk}$, to arrive at the resulting measure of "chunkiness". Chunkiness can be seen as the relative change of the Wasserstein distance of chunk ABC $\Delta W_{ABC}$ compared to control $\Delta W_{control}$. The resulting evaluation of chunkiness on the three groups is shown in Fig. 3c. Chunkiness differed significantly between the three conditions ($F(2) = 3.20$, $p = .04$).

In particular, the size 2 group had a negative relative Wasserstein shift ($\hat{\beta} = -26.92$, $t(137) = -2.37$, $p = 0.02$). This means that the size 2 group's reaction time distribution became less homogeneous after training, indicating that the reaction time to press B deviated more from C. This was expected as for the size 2 group, pressing B and C after A should be one within and one between-chunk reaction time. On the other hand, the change of Wasserstein distance between the size 3 and the independent group condition was not significant, even though we would have expected this group to become more homogeneous in their reaction times. One reason for this surprising result could be that, while the reaction time distribution upon the instructions "B" and "C" became closer to each other relative to control, the shift may have not uniformly impacted the calculation of Wasserstein distance. It could also be that positions at the end of a chunk can be learned faster than the intermediate elements, as found in[45].

In summary, we verified the prediction that the size 2 group had less homogeneous transition times within the chunk ABC than the other groups. However, we did not observe an increased chunkiness for the size 3 group, possibly due to non-uniform speed-ups of RTs.

**Reaction time regression.** The learning of chunks during the training blocks will influence how participants perceive the transition from one instruction to another. As the rational chunking model predicts that participants in the size 3 group will learn chunk ABC, size 2 group will learn chunk AB, and no such chunks in the independent group. This chunk learning will influence how participants perceive the items in the test blocks in a way that size 2 group and size 3 group may react to the sequence in a more deterministic manner. Therefore we studied the influence of transition probabilities on participants' reaction times during the test blocks (Fig. 3d) on the correct trials. We use three transition probability matrices as regressors. One is the true transition (TR), which is the ground truth transition probability used to generate a sequence in the test block. The second one is C2 transition matrix that contains a deterministic transition from A to B. And the third one is C3 transition matrix, with a deterministic transition from A to B, and B to C. The rest of the entries of C2 and C3 are the same as TR.

We fitted a linear mixed-effects regression using log-reaction times as the dependent variable, assuming a random intercept for each participant. The independent variables were the TR, C2, and C3 transition probabilities, group, as well as interaction effects between group and each of the transition probabilities.

The best regression contained the predicted transition probabilities as well as three interaction effects with groups ($\chi^2(8) = 129.6$, $p < 0.001$). The first interaction was between TR and the size 2 group ($\hat{\beta} = 0.13$, $t(25040) = 5.24$, $p < 0.001$), showing that the effect of TR learned by the size 2 group was significantly up-weighted. The interaction between TR and the size 3 group was also significantly up-weighted ($\hat{\beta} = 0.10$, $t(25040) = 3.84$, $p < 0.001$). TR transition probabilities as an independent variable slowed down the reaction times for the size 2 and size 3 groups more than for the independent group.

The interaction was significantly down-weighted between the C2 chunky transition probabilities and the size 2 group ($\hat{\beta} = -0.08$, $t(25040) = -7.51$, $p < .001$) and the size 3 group ($\hat{\beta} = -0.03$, $t(25040) = -2.68$, $p = 0.007$). This indicates that C2 transition probabilities sped up the reaction times for the size 2 and the size 3 group more than for the independent group, and the effect is stronger for the size 2 group.

Finally, the interaction was down-weighted between the C3 transition probabilities and the size 3 group ($\hat{\beta} = -0.12, t(25040) = -6.27$, $p < .001$), as well as an interaction between the C3 transitions and the size 2 group ($\hat{\beta} = -0.11, t(25040) = -5.34$, $p < 0.001$). These significant interaction effects indicate that C3 transition probabilities sped up the reaction times for the size 2 and the size 3 group more than that for the independent group. In summary, we found predictable relations between participants' reaction times and the chunk-implied transition probabilities across groups. In particular, C2 transitions were more significantly related to speed-ups for the size 2 group than the size 3 group, while C3 transitions significantly related to speed-ups for the size 3 group more than size 2 group. This relation is consistent with the prediction generated by the rational chunking model.

**Chunk increase.** The rational model of chunking, as shown in Fig. 2c, predicts that participants' learned chunks should reflect the underlying chunks used to generate the sequence. That is, the chunk 3 group should learn more chunk ABC than the chunk 2 group and the independent group. Additionally, both chunk 3 and chunk 2 group should learn more chunk AB than the independent group. This acquisition of chunks during the 6 training blocks is going to influence how participants behave in the baseline and test blocks. We here test this prediction concretely by examining how often chunk AB and chunk ABC are used by the three groups in the test blocks, using baseline blocks as a control.

We look at the exact chunks used by participants by classifying within and between chunk reaction time using the mixture of Gaussian method (see section method). As an illustration, Fig. 5 shows the participants' data. The reaction time, instruction displayed, and this participant's actual key press is shown on the first, second, and third row. Instruction for A, B, C, D, is separately color-coded by green, blue, magenta, and orange boxes. When this participant has pressed a key incorrectly, that trial is marked by a red box. Using the distribution of reaction time data accumulated for this participant over all trials, we classify individual trials into within or between chunk key press, whichever results in a higher likelihood in the mixture. Once each trial is classified as within or between chunk trials, we can mark the chunks learned by this participant by connecting all within-chunk trials and the first between chunk trial that starts before those within-chunk trials as belonging to the same chunk (thereby, we can mark the size of chunks by connecting black round dots shown on the fourth row of the above). In this way, we can identify the content of the chunks learned by this participant as reflected by their reaction time speed-up, in addition to the chunk size learned by each participant, as illustrated in Fig. 5 in supplementary information.

We measured the number of times each participant chunked AB and ABC in baseline and test blocks and evaluated the increase $\Delta N = N_{test} - N_{baseline}$. Figure 3e shows $\Delta N_{AB}$ and $\Delta N_{ABC}$, measured separately for the three groups.

Fitting a linear model setting $\Delta N$ of AB as the dependent variable and group as an independent variable showed a significant effect of group ($F(2) = 8.64$, $p < 0.001$). Compared to the independent group, the size 2 group ($\hat{\beta} = 7.16, t(139) = 3.68$, $p < 0.001$) and the size 3 group ($\hat{\beta} = 6.80, t(139) = 3.55$, $p < 0.001$) chunked AB significantly more often in the test blocks than in the baseline blocks.

The same analysis for $\Delta N$ of chunk ABC also showed a significant effect of group ($F(2) = 10.63$, $p < 0.001$). The size 3 group chunked significantly more ABC as chunks than the independent group ($\hat{\beta} = 5.36$, $t(139) = 4.53$, $p < 0.001$). Participants in the size 2 group also chunked more ABC than the participants in the independent group ($\hat{\beta} = 1.94, t(139) = 1.62$, $p = 0.10$). Compared with the size 2 group, the size 3 group chunked ABC also significantly more often ($\hat{\beta} = 3.42, t(139) = 2.92$, $p = 0.004$). Overall, participants' behavior is qualitatively consistent with model prediction. Training on sequences with chunks increased participants' tendency to use those chunks in the test blocks.

**Chunk reuse.** Since the rational chunking model reuses previously learned chunks to construct new ones, we wanted to study whether participants' chunking behavior reflected this feature of our model. As explained in the method section, the mixture of the Gaussian rt classification method returns the estimated learning progress of chunks for each participant throughout the experiment. We examined participants' chunk reuse probability based on how individual participants learned the chunks during the training blocks (Fig. 3f).

The chunk reuse probability was evaluated on chunks of size three or bigger (excluding chunks with single-item repetitions). Every time such a chunk occurs in the sequence, we check whether it reuses any of the 30 previous chunks. Figure 8 in SI shows an example of chunk CDABCD reusing BCD as one of its previous chunks. By tagging each chunk learned by the participant as reusing one of the previous chunks or not, we arrived at

a chunk reuse probability for each participant. Figure 3f shows the average chunk reuse probability across the three groups.

We found that the chunk reuse probability differed significantly between the groups. Fitting a linear model taking the reuse probability as the dependent variable and group as the independent variable showed a significant effect of condition ($F(2) = 13.99$, $p < 0.001$). Both the size 2 ($\hat{\beta} = 0.17, t(139) = 3.63$, $p < 0.001$) and the size 3 group ($\hat{\beta} = 0.24, t(139) = 5.17$, $p < 0.001$) reused chunks significantly more often than the independent group. There was no significant difference between the size 2 and the size 3 group ($\hat{\beta} = 0.07, t(139) = 1.51$, $p = 0.13$). The tendency to reuse previously learned chunks is consistent with how our model creates chunks, i.e., creating a new chunk by combining previously learned chunks. Interestingly, sequence statistics modulated participants' tendency to reuse chunks. When the sequence contained embedded chunks that render reuse beneficial to performance, participants tended to reuse previously acquired chunks more often than when the sequence only contained independent item instantiations. The observation that participants reused previously acquired chunks echoes previous findings in the literature on transferring motor skills, which showed that people transfer chunks from a practiced sequence to a test sequence when shared chunks between the two[14]. The reuse and transfer process in the current task was an ongoing learning behavior while participants practiced the training sequence.

## Experiment 2: learning chunks of different sizes to balance the speed–accuracy trade-off

In Experiment 2, we test the model prediction that chunk learning adapts to task demands. Participants should chunk more under time pressure, even given a sequence without chunks within.

We randomly assigned participants to one of two groups: the fast group and the accurate group, creating a two-groups between-subjects design. Both groups were trained on sequences generated from the "illusory" matrix that contained no true chunks but high and medium single item transitions (Fig. 1c). The experiment structure was identical to the structure of Experiment 1: 10 blocks with 100 trials each. The training blocks were sandwiched between baseline and test blocks, see Fig. 1a. In those blocks, accuracy and reaction times were displayed right at the end of each trial. In the middle 6 blocks, from block 3 to block 8 (i.e. the training blocks), participants in the fast group were instructed to act "as fast as possible even if it might lead to mistakes", and participants in the accurate group were instructed to act "as accurate as possible even it might slow you down". The fast group was told that their reward depended on how fast they pressed the instructed key and were given trial-by-trial feedback on their reaction times. The accurate group was told that their reward depended on their accuracy and were given trial-by-trial feedback on the correctness of their responses. Both groups received the same instruction to act "as fast and accurately" as possible during the baseline and the test blocks (block 1-2 and 9-10, see Fig. 1a.

**Manipulation check.** We first assessed whether the instructions to be fast or accurate influenced participants' reaction times and accuracy during the training blocks. Shown in Fig. 4a are the average reaction time and accuracy for the two groups. Fitting a linear mixed-effects regression onto participants' reaction times assuming a random intercept over individual participants showed a significant effect of group ($\chi^2(1) = 9.84$, $p = .002$), showing that participants in the fast group responded faster during the training blocks than participants in the accurate group ($\hat{\beta} = 81.71$, $t(113.93) = 3.19$, $p = .0001$). We also fitted a mixed-effects logistic regression of group to test whether participants responded correctly on each trial, adding a random intercept for each participant. This analysis also showed a significant effect of group ($\chi^2(1) = 9.67$, $p = .002$), with participants in the accurate group responding on average more accurately during the training blocks than participants in the fast group ($\hat{\beta} = 0.54$, $z = 3.18$, $p = .001$). Thus, we conclude that our experimental manipulation induced the intended behavior for the two groups during the training blocks.

**Chunky boost.** The rational chunking model predicts that the fast group, compared to the accurate group, should learn more chunks. This influence of different instructions will affect the behavioral change of both groups' performance in the test blocks relative to the baseline blocks. We again look at an indicator of learning size-2 chunks by evaluating chunky boost on the within-chunk reaction times of the size 2 chunks. This time, the chunky boost was evaluated on the most frequently occurring size-2 chunks in the sequence produced by the "illusory" transition matrix: AB, BC, CD, and DA. AB and CD are the size-2 chunks with high transition probability ($p = 0.9$). BC and DA are size 2 chunks with medium transition probability ($p = 0.7$). The corresponding control chunks were size two subsequences that did not begin with the first chunk items. As an example, the control chunks for AB were BB, CB, and DB.

We conjectured that the fast group would learn more size two chunks with high and medium probability. We look at how the training schedule changes the value of within-chunk (value marked by red boundary) reaction time for size 2 chunks in the test blocks compared to the baseline blocks. We calculate the signed effect size, Cohen's d, on the within-chunk reaction time from the baseline to the test block. The same procedure was applied for the control chunks. Then the Cohen's d of the control chunks was subtracted from the size 2 chunks to arrive at the chunky boost measure. The chunky boost measured by a change of Cohen's d $\Delta d$ is shown in Fig. 4b. Fitting a linear mixed-effects regression onto participants' change of Cohen's d, assuming a random intercept over participants showed a significant effect of the group ($\chi^2(1) = 7.25$, $p = .007$). Participants in the fast condition showed a greater relative boost in reaction times to chunky transitions as compared to participants in the accurate group ($\hat{\beta} = 0.24, t(73) = 2.71$, $p = .008$). We, therefore, concluded that participants in the fast group chunked more size two chunks than participants in the accurate group, as was predicted by our model.

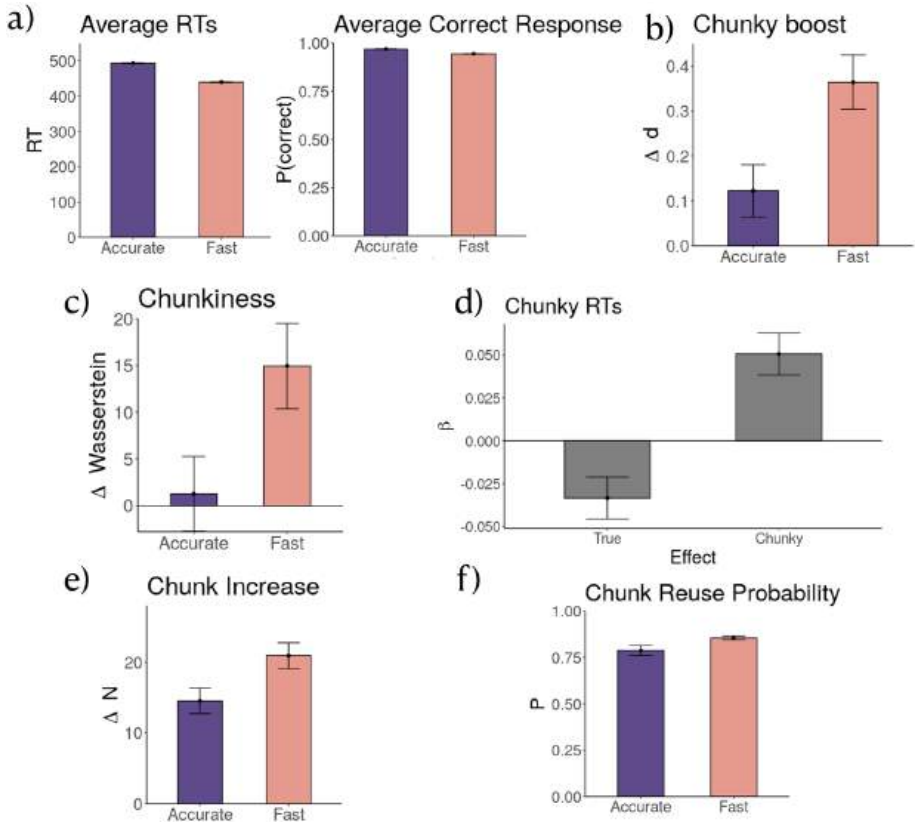


**Figure 4.** Results of Experiment 2. (**a**) Manipulation check. Average reaction times and average response accuracy during training blocks by group. (**b**) Chunky Boost of size 2 chunks as measured by change of Cohen's *d* by group evaluated on baseline and test blocks. The size 2 chunks include AB, BC, CD, and DA. (**c**) Chunkiness measured by a relative change of Wasserstein distance of size 3 chunks including ABC, BCD, CDA, DAB between the baseline and the test blocks. (**d**) Coefficient of interaction effect between chunky and true transition probabilities on reaction times during the test blocks. (**e**) Chunk increase from the baseline to the test blocks by condition for size-2 (AB, CD, BC, DA) and for size-3 chunks (ABC, BCD, CDA, DAB). (**f**) Chunk reuse probability by group. For all plots, error bars indicate the standard error of the mean.

**Chunkiness.** The rational chunking model that exerts a speed-accuracy trade-off predicts that the fast group should learn longer chunks than the accurate group. Similar to the analysis in experiment 1, we examined this prediction from the model by evaluating the chunkiness measure as an indicator of participants learning size-3 chunks (ABC, BCD, CDA, DAB) that can occur in the training sequence generated by the "illusory" transition matrix. If participants have learned any of those size-3 chunks, then the distributions of within-chunk reaction times $rt_2$ and $rt_3$ should become more similar to each other following the presentation of the first item. We use the Wasserstein distance to evaluate the homogeneity of this reaction time distribution, illustrated in Fig. 7.

To assess the initial separation of the two distributions for both groups before training, we evaluated the Wasserstein distance between the distribution of $rt_2$ and $rt_3$ on the baseline blocks, $Wasserstein(rt_2, rt_3)_{baseline}$, when both groups of participants are trained on the illusory transition sequences. To assess how much training influences $rt_2$ and $rt_3$ in the test blocks, we evaluate the same Wasserstein distance on both groups in the test blocks, to arrive at $Wasserstein(rt_2, rt_3)_{test}$. An indicator of learning the frequent size-3 chunk (ABC, BCD, CDA, or DAB) during the training blocks, is that $Wasserstein(rt_2, rt_3)_{test}$ should become smaller in the test blocks than in the baseline blocks, resulting in more homogeneous $rt_2$ and $rt_3$. We subtracted $Wasserstein(rt_2, rt_3)_{test}$ from $Wasserstein(rt_2, rt_3)_{baseline}$ to calculate $\Delta W_{chunk}$, the change of reaction time homogeneity.

To control for the effect of training resulting in an overall increase of reaction time homogeneity, we compared this change of Wasserstein with size 3 sub-sequences that are not the frequent size-3 chunks (ABC, BCD, CDA, DAB), as a control, to arrive at $\Delta W_{control}$ as the difference between $W_{test}$ and $W_{train}$.

Finally, we subtracted $\Delta W_{control}$ from $\Delta W_{chunk}$, to arrive at the resulting measure of "chunkiness". This is the relative change of the Wasserstein distance of size 3 chunks $\Delta W_{chunk}$ compared to control $\Delta W_{control}$. According to the model prediction, if participants in the fast group learned more size three chunks than those in the accurate group, one would expect the fast group to have a higher measure of chunkiness than those in the accurate group. Figure 4c shows the resulting chunkiness measure. The change $\Delta W$ on size 3 chunks differed significantly between the two groups ($\chi^2(1) = 4.71$, $p = .02$), with the fast group showing a higher chunkiness compared to the accurate group ($\hat{\beta} = 12.33, t(88) = 2.15$, $p = .03$). Thus, participants in the fast condition showed a higher

relative chunkiness in their reaction times to size three chunks in the sequence than participants in the accurate group, as indicated by the chunkiness measure.

**Reaction time regression.** Our model simulations showed that when the speed-accuracy trade-off parameter $w \to 0$ and accuracy becomes the only optimizing term, the model learns about the original transition matrix. However, as $w \to 1$ and speed is the only optimization term, the model learns a polarized transition probability where all the high and medium single element transitions become 1.

We aimed to test whether or not the $w$ parameter of our model captured how the "fast" versus "accurate" instructions affected participants' chunking behavior. In this way, the instruction will influence how participants perceive the items in the test block, the polarized transition should resemble more of the fast group, and the original transition matrix shall resemble more of the reaction time in the accurate group. An illustration of this procedure is shown in Fig. 6 in SI.

To this end, we fitted a linear mixed-effects regression to participants' log reaction times in the correct trials of the test blocks (so that the skew of RT distribution as deviating from a normal distribution is removed), assuming a random intercept for each participant. The independent variables are the group, the true transition probabilities learned with $w = 0$ (Fig. 2d left), and chunky transition probabilities that correspond to the learning result of the model with $w = 1$ (Fig. 2d right). The best regression model contained the main effects of the chunky and true transition probabilities as well as two interaction effects with the given condition ($\chi^2(3) = 34.86$, $p < 0.001$). The first interaction was between the true probabilities and group (see Fig. 4d; $\hat{\beta} = -0.03$, $t(16740) = -2.74$, $p = 0.006$), as the true transition probabilities were more consistent with participants' responses in the accurate group than with those in the fast group. The second interaction was between the chunky transition probabilities and group ($\hat{\beta} = 0.05$, $t(16740) = 4.13$, $p < .001$), indicating that the simulated effect of a higher tendency to chunk was more predictive of the behavior of the fast group than that of the accurate group. Thus, the chunking bias induced by the speed-accuracy trade-off parameter of our model matched the bias observed in the reaction time pattern of the participants under speed demands.

**Chunk increase.** The rational model of chunking that trades off speed with accuracy learns longer chunks as the emphasis on speed weights more than accuracy. It predicts that the fast group should learn longer chunks more often than the accurate group. This acquisition of longer chunks during the 6 training blocks should influence participants' chunking behavior in the test blocks relative to the baseline blocks. A concrete examination of this prediction is to take the frequency of concrete size 2 and size 3 chunks in the test block to see if they are used more often compared to the baseline blocks.

Similar to experiment 1, the exact chunks learned by participants are tagged using the mixture of Gaussian method. We studied the number of times size 2 chunks appear in participants' chunking profiles (AB, BC, CD, DA) and size 3 chunks (chunk ABC, BCD, CDA, DAB). We compared the increase in those chunks from the baseline and the test blocks and compared $\Delta N$ across the two groups. Shown in Fig. 4e is the increase in the number of size 2 and size 3 chunks.

Fitting a linear model on $\Delta N$ with group as the independent variable revealed a significant effect of group ($F(1) = 8.13$, $p = 0.005$). Compared to the accurate group, the fast group acquired more chunks from the baseline to the test block ($\hat{\beta} = 6.41$, $t(358) = 2.85$, $p = 0.005$). Consistent with the model prediction that the fast group should chunk longer chunks, participants in the fast group indeed has a higher increase in the number of size 2 and size 3 chunks compared to the accurate group.

**Chunk reuse.** We also look at whether participants tend to reuse previously learned chunks to construct new chunks during the training blocks, similar to what we tried in experiment 1, to see whether the participant's behavior reflects the feature of this model. The progress of chunk learning as identified by the mixture of Gaussian method is used to examine participants' chunk reuse probability (illustrated in Fig. 8 in SI). The chunk reuse probability was evaluated on chunks of size three or bigger (excluding chunks with single-item repetitions). Every time such a chunk occurs in the sequence, we check whether it reuses any of the 30 previous chunks. By tagging each chunk learned by the participant as reusing one of the previous chunks or not, we arrived at a chunk reuse probability for each participant across the fast and accurate group. Figure 4f shows the average chunk reuse probability across the three groups. Participants' high reuse probability echoes the model feature of reusing previously learned chunks to construct new ones. Interestingly, instruction also influences the tendency of chunk reuse. Fitting a linear model to participants' chunk reuse probability showed that chunk reuse differed significantly between the two groups ($F(1) = 4.75$, $p = .03$). Participants in the fast group reuse chunks more frequently than those in the accurate group ($\hat{\beta} = 0.07$, $t(114) = 2.18$, $p = .03$). This may show that reuse is especially prominent when participants are trying to be fast, since recycling the previously learned chunks can make more progress towards reaction time speed-up.

## Discussion

How people perceive and extract structure from a sequence of perceptual stimuli has been a longstanding question of psychological investigations. *Chunking* has been proposed as a mechanism to identify repeated patterns and segment sequences into those patterns. This way of segregating patterns into discrete chunks can improve storage, retrieval, and planning across multiple psychological domains.

In the current work, we have proposed that chunking benefits the timely and accurate execution of sequential actions. We used a rational model of chunking that adapts its representation to optimize a trade-off between speed and accuracy to simulate chunk learning in a serial reaction time task. Our simulations predicted that

participants should chunk more if chunks are indeed part of the generative model and should, on average, learn longer chunks when optimizing for speed than accuracy. We tested these predictions in two experiments. In Experiment 1, participants learned from sequences with different embedded chunks. In Experiment 2, participants were instructed to act as fast or accurately as possible. Multiple measures of chunking confirmed our model's predictions in both experiments. In summary, our results shed new light on the benefits of chunking and pave the way for future studies on step-wise representation learning in structured domains.

The model's prediction relating chunking to reaction time speed up relied partially on the Linear Ballistic Accumulator framework to translate within-chunk action prediction to an elevated starting point of the evidence accumulation, making the within-chunk action more likely to cross the decision threshold. Yet it remains challenging to explicitly fit a hierarchical LBA model over all participants, trials, and between-subject differences using our current data. This divergence is potentially due to a large number of observations. Therefore, one part of our analyses used model-predicted transition probabilities with accuracy and speed extremes to fit participants' reaction times. Nonetheless, future studies should look into the influence of chunking on the starting point of the LBA model in a fully Bayesian and hierarchically-structured model.

Currently, our model's predictions were primarily qualitative, and we did not compare across a more extensive set of alternative models. Even though we tested model-specific predictions such as the reuse of previously created chunks to parse the sequence and the speed-up and increased homogeneity of reaction times for within-chunk reaction times, future studies should further compare explicit predictions of different chunking models. We believe that our current work is a concrete first step towards building fine-grained models of human chunking in SRTs. We plan to compare our model to several alternatives in future tasks requiring participants to learn increasingly more hierarchically-structured chunks.

Furthermore, it would be very hard to exclude the contribution of associative learning to the effect observed in Experiment 1, as the rational chunk learning model also learns chunks by association. However, an associative learning model does not explain our observation in Experiment 2, which can only be accounted for by a rational chunking model that trades off speed with accuracy.

Finally, not all of our measures of increased chunking provided evidence for our model's predictions. In particular, in Experiment 1, the measure of chunkiness did not increase for the size 3 group even though we would have a priori expected such an increase. We believe that this increase did not appear because participants' speed-up of within-chunk reaction times was not uniform across both transitions of the size three chunks. Moreover, we did find a decrease of homogeneity for the size 2 group, which was as expected because learning about the size 2 chunk should make the reaction time discrepancy between B and C larger. Importantly, we did find systematic differences across all other measures in both experiments and, therefore, believe that the current data support our model's predictions.

## Conclusion

We investigated chunking behavior across two experiments and several measures. We found that chunking behavior depends on sequence statistics and task demands. When there are chunks in the training sequence, participants learn the underlying embedded chunks. Additionally, task demands modulate chunking behavior. Participants tend to chunk more when they are optimizing for speed rather than accuracy. Such chunking behavior occurs even in sequences lacking any deterministic transition probabilities. Our results suggest characteristics of chunking and how they interact with task demands. Our rational model of chunking captures and predicts these findings. The success of model predictions depends primarily on the gradual change in previously acquired representations to rationally adapt to sequence structure and task demands. We hope that our findings and model are a good step towards understanding human chunk learning across multiple domains.

## Methods

**Ethics statement.** Informed consent was obtained from all participants before participation, and the experiments were performed in accordance with the relevant guidelines and regulations approved by the ethic committee of the University of Tuebingen (Ethik-Kommission an der Medizinischen Fakultät der Eberhard-Karls-Universität und am Universitätsklinikum Tübingen), under the study title: Experimente zum Sequenz- und Belohnungslernen, with application number 701/2020BO.

Participants' data were analyzed anonymously. Upon agreement to participate in the study, they consented on a data protection sheet approved by the data protection officer of the MPG (Datenschutzbeauftragte der MPG, Max-Planck-Gesellschaft zur Förderung der Wissenschaften).

**Recruitment of participants.** For Experiment 1, we recruited 142 participants from Amazon Mechanical Turk, out of which sixty-nine were female. Their median age was between 30 and 40, and the overall age ranged from 18 to above 50. This experiment took around 25 minutes to complete. After completing the task, participants received a base pay of $2 and a performance-dependent bonus of up to $6.

For Experiment 2, we recruited a total of 116 participants for our study, again from Amazon Mechanical Turk. Forty-eight participants were female; participants' median age was between 30 and 40, and the overall age ranged from 18 to above 50. After completing the task, participants received a base pay of $2 and a performance-dependent bonus of up to $4.

**Payment.** For Experiment 1, a performance-dependent bonus was calculated as the weighted sum of participants' accuracy and reaction times. When the average accuracy was below 70%, the bonus was set to 0. The bonus for being fast was calculated as $bonusfast = bonusmax - (\overline{rt} - 600) \times 0.025$, where $\overline{rt}$ indicates the average reaction time and *bonusmax* indicates the maximal bonus. Participants were rewarded with a reaction time

bonus when their average reaction time was below 600ms. Additionally, an accuracy bonus was calculated as the mean performance accuracy times the maximal bonus, $bonusacc = \overline{acc} \times bonusmax$. At the end of the experiment, the total bonus was calculated as a weighted average between the bonus for participants' accuracy and the bonus for their reaction times, $bonus = 0.5 \times bonusacc + 0.5 \times bonusfast$. If the final bonus was below 0, it was set to 0. If it was above the maximal bonus, it was set to the maximal bonus. On average, participants received $5.64 for their participation.

For Experiment 2, USD 2 were awarded as a base pay for every participant who completed the experiment. Additionally, participants received a performance-dependent bonus, ranging from 0 to the maximum of USD 4. This bonus was calculated separately for the fast and accurate groups. For the fast group, the bonus was 0 when their average accuracy was below 60%. If the average accuracy was above 60%, then the bonus was calculated as $bonusfast = (1000 - \overline{rt})/800 \times bonusmax$ This reward function penalized average reaction times that were slower than 1000ms. For the accurate group, the bonus was calculated as the percentage of their accuracy multiplied by the maximal bonus for the accuracy group, $bonusacc = \overline{acc} \times bonusmax$. Finally, the final bonus was again forcet to be between 0 and *bonusmax* (USD 4). The mean reward earned by participants for this experiment was $4.16.

**Filtering criteria.** We decided to discard participants using a fixed RT threshold based on independent pilot data we had collected earlier. Since the study was conducted on MTurk, and we do not have access to the conditions on how the experiment was conducted, we applied stricter filtering criteria. For Experiment 1, we excluded participants with an average reaction time longer than 1000ms or an average accuracy lower than 90%. 95.1% of participants had an accuracy above 90%. 91% of participants had an average reaction time below 1000 ms. Out of 142 participants who participated in Experiment 1, 20 participants were excluded and 122 remained after applying this exclusion criterion.

For Experiment 2, the same exclusion criteria were applied on the baseline and test blocks when both groups were asked to be as fast and accurate as possible. 96.7% of participants had an accuracy above 90%, and 96.7% had an average reaction time below 1000 ms. The exclusion criteria differed between the two groups on the training blocks. Participants in the fast group were excluded when the average reaction time was above 750ms ($n = 13$). Those in the accurate group were excluded when their accuracy was below 90% ($n = 10$). Additionally, we excluded participants who repeatedly failed attention checks before and after the experiment ($n = 3$). Out of 116 participants, 26 were excluded in total. All of the following analyses were performed on the data of the remaining 90 participants.

For the reaction-time based analysis including the chunky boost, chunkiness, and the mixture of Gaussians classification, we further excluded trials in which participants took more than 1000ms to respond. This amounted to 8.4% of all trials in Experiment 1, and 3.3% of all trials in Experiment 2.

**Mixture of Gaussians model.** We used a mixture of Gaussians model to retrieve chunky transitions for each participant's responses from their reaction times. Chunks are classified based on participants' responses, irrespective of their correctness. In the case of an error where a participant has pressed A B D C although the instruction was A B D B, and the reaction time classification for each of these trials are between, within, within, within (and the subsequent trial is between again), then A B D C is classified as a chunk. In the case of errors, we consider their erroneous response rather than the instruction because the response reflect their underlying prediction.

The reaction time distribution for individual participants was used to classify individual trials as between or within-chunk reaction times. These reaction time distributions were fitted by a mixture of Gaussians model. The likelihood of belonging to the smallest mixture component was used to classify a reaction time as a within-chunk reaction time. This classification was then used for the identification of chunks for all experimental trials of every participant.

The classification of RTs by using multi-modal distributions was motivated by the idea that distinct processes might generate the within-chunk and between-chunk reaction times. During an SRT trial, if the participant has no expectation for the next upcoming instruction, she will first have to identify the instructed key on the screen before beginning to press a key. This will make her between-chunk reaction times larger. In contrast, if a participant has learned chunks, she anticipates the next instruction before it is even shown. If the upcoming instruction is within her expected chunk, the action to look for instructions displayed on the screen can be omitted, and she can directly engage in pressing their expected subsequent key. This will make her within-chunk reaction times smaller. Additionally, the mixture of Gaussians model also takes into account participants' post-error slow-downs. These correspond to the trials when a participant has made or almost made a mistake and corrects this tendency to press an expected key upon the observation of a conflicting instruction. The behavior of modifying the wrong key-press is slowing down the reaction time even more.

Because these three processes contribute to distinct components of participants' reaction times, a mixture of 3 Gaussian distributions was used to fit their reaction time distributions. We assumed that the within-chunk reaction time distributions had the lowest mean, the between-chunk reaction time distributions had a higher mean, and the post-error slow-down reaction time distribution had the highest mean. We fitted the mixture of Gaussian model to individual participants' reaction times, filtering out RTs above 1000ms. A likelihood estimate belonging to each distribution amongst the mixture was assigned to the reaction times of each trial. A validation of this method has been included to the Supplementary Information.

## Data and code availability

The data collected and code used for analyzing this study can be found in this github repository: https://github.com/swu32/experimental_chunking.

Received: 28 March 2022; Accepted: 13 March 2023
Published online: 11 May 2023

## References

1. Miller, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev.*https://doi.org/10.1037/h0043158 *(1956).*
2. Laird, J. E., Rosenbloom, P. S. & Newell, A. Towards chunking as a general learning mechanism. In AAAI, 188–192 (1984).
3. Graybiel, A. M. The basal ganglia and chunking of action repertoires. *Neurobiol. Learn. Mem.* **70**, 119–136 (1998).
4. Servan-Schreiber, E. & Anderson, J. R. Learning artificial grammars with competitive chunking. *J. Exp. Psychol. Learn. Mem. Cogn.* **16**, 592 (1990).
5. Terrace, H. S. Chunking by a pigeon in a serial learning task. *Nature*https://doi.org/10.1038/325149a0 *(1987).*
6. Mathy, F. & Feldman, J. What's magic about magic numbers? Chunking and data compression in short-term memory. *Cognition-*https://doi.org/10.1016/j.cognition.2011.11.003 *(2012).*
7. Lashley, K. S. *The Problem of Serial Order in Behavior* Vol. 21 (Bobbs-Merrill Oxford, United Kingdom, 1951).
8. Gobet, F. *et al.* Chunking mechanisms in human learning. *Trends Cog. Sci.*https://doi.org/10.1016/S1364-6613(00)01662-4 *(2001).*
9. Graybiel, A. M. The basal ganglia and chunking of action repertoires. *Neurobiol. Learn. Memory* **70**, 1–2. https://doi.org/10.1006/nlme.1998.3843 (1998).
10. Egan, D. E. & Schwartz, B. J. Chunking in recall of symbolic drawings. *Memory Cogn.*https://doi.org/10.3758/BF03197595 *(1979).*
11. Ellis, N. C. Sequencing in SLA: Phonological memory, chunking, and points of order. *Stud. Second Lang. Acquis.*https://doi.org/10.1017/S0272263100014698 *(1996).*
12. Koch, I. & Hoffmann, J. Patterns, chunks, and hierarchies in serial reaction-time tasks. *Psychol. Res.*https://doi.org/10.1007/PL00008165 *(2000).*
13. Brady, T. F., Konkle, T. & Alvarez, G. A. Compression in visual working memory: Using statistical regularities to form more efficient memory representations. *J. Exp. Psychol.: General*https://doi.org/10.1037/a0016797 *(2009).*
14. Müssgens, D. M. & Ullén, F. Transfer in motor sequence learning: Effects of practice schedule and sequence context. *Front. Human Neurosci.*https://doi.org/10.3389/fnhum.2015.00642 *(2015).*
15. Chase, W. G. & Simon, H. A. Perception in chess. *Cogn. Psychol.*https://doi.org/10.1016/0010-0285(73)90004-2 *(1973).*
16. Gobet, F. & Simon, H. A. Expert chess memory: Revisiting the chunking hypothesis. *Memory*https://doi.org/10.1080/741942359 *(1998).*
17. Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M. & Gershman, S. J. Compositional inductive biases in function learning. *Cogn. Psychol.*https://doi.org/10.1016/j.cogpsych.2017.11.002 *(2017).*
18. Schulz, E., Quiroga, F. & Gershman, S. J. Communicating compositional patterns. *Open. Mind* **4**, 25–39 (2020).
19. Tomov, M. S., Yagati, S., Kumar, A., Yang, W. & Gershman, S. J. Discovery of hierarchical representations for efficient planning. *PLoS Comput. Biol.*https://doi.org/10.1371/journal.pcbi.1007594 *(2020).*
20. Wickelgren, W. A. Speed-accuracy tradeoff and information processing dynamics. *Acta Physiol. (Oxf)* **41**, 67–85. https://doi.org/10.1016/0001-6918(77)90012-9 (1977).
21. Bogacz, R., Hu, P. T., Holmes, P. J. & Cohen, J. D. Do humans produce the speed-accuracy trade-off that maximizes reward rate?. *Q. J. Exp. Psychol.* **63**, 863–891. https://doi.org/10.1080/17470210903091643 (2010).
22. MacKay, D. G. The problems of flexibility, fluency, and speed-accuracy trade-off in skilled behavior. *Psychol. Rev.* **89**, 483–506 (1982).
23. Fitts, P. M. Cognitive aspects of information processing: III. Set for speed versus accuracy. *J. Exp. Psychol.* **71**, 849–857. https://doi.org/10.1037/h0023232 (1966).
24. Nissen, M. J. & Bullemer, P. Attentional requirements of learning: Evidence from performance measures. *Cogn. Psychol.* **19**, 1–32 (1987).
25. Willingham, D. B., Nissen, M. J. & Bullemer, P. On the development of procedural knowledge. *J. Exp. Psychol. Learn. Mem. Cogn.* **15**, 1047 (1989).
26. Robertson, E. M. The serial reaction time task: Implicit motor skill learning?. *J. Neurosci.* **27**, 10073–10075 (2007).
27. Keele, S. W., Ivry, R., Mayr, U., Hazeltine, E. & Heuer, H. The cognitive and neural architecture of sequence representation. *Psychol. Rev.* **110**, 316–339. https://doi.org/10.1037/0033-295X.110.2.316 (2003).
28. Willingham, D. B., Salidis, J. & Gabrieli, J. D. Direct comparison of neural systems mediating conscious and unconscious skill learning. *J. Neurophysiol.* **88**, 1451–1460. https://doi.org/10.1152/jn.2002.88.3.1451 (2002).
29. Howard, J. H. & Howard, D. V. Age differences in implicit learning of higher order dependencies in serial patterns. *Psychol. Aging* **12**, 634–656. https://doi.org/10.1037//0882-7974.12.4.634 (1997).
30. Romano, J. C., Howard, J. H. & Howard, D. V. One-year retention of general and sequence-specific skills in a probabilistic, serial reaction time task. *Memory* **18**, 427–441. https://doi.org/10.1080/09658211003742680 (2010).
31. Bornstein, A. & Daw, N. Cortical and hippocampal correlates of deliberation during model-based decisions for rewards in humans. *PLoS Comput. Biol.* **9**, e1003387. https://doi.org/10.1371/journal.pcbi.1003387 (2013).
32. Schvaneveldt, R. W. & Gomez, R. L. Attention and probabilistic sequence learning. *Psychol. Res.* **61**, 175–190. https://doi.org/10.1007/s004260050023 (1998).
33. Cleeremans, A. & McClelland, J. L. Learning the structure of event sequences. *J. Exp. Psychol. Gen.* **120**, 235–253 (1991).
34. Provyn, J. P. Associative processes in statistical learning: Paradoxical predictions of the past. *Psychol.-Diss.* **179**, 78 (2013).
35. Perruchet, P. & Vinter, A. Parser: A model for word segmentation. *J. Mem. Lang.* **39**, 246–263. https://doi.org/10.1006/jmla.1998.2576 (1998).
36. Servan-Schreiber, E. & Anderson, J. Learning artificial grammars with competitive chunking. *J. Exp. Psychol. Learn. Mem. Cogn.* **16**, 592–608. https://doi.org/10.1037/0278-7393.16.4.592 (1990).
37. French, R. M., Addyman, C. & Mareschal, D. TRACX: A recognition-based connectionist framework for sequence segmentation and chunk extraction. *Psychol. Rev.* **118**, 614–636. https://doi.org/10.1037/a0025255 (2011).
38. Cleeremans, A., Servan-Schreiber, D. & McClelland, J. L. Finite state automata and simple recurrent networks. *Neural Comput.* **1**, 372–381. https://doi.org/10.1162/neco.1989.1.3.372 (1989).
39. Wang, Q., Rothkopf, C. A. & Triesch, J. A model of human motor sequence learning explains facilitation and interference effects based on spike-timing dependent plasticity. *PLoS Comput. Biol.*https://doi.org/10.1371/journal.pcbi.1005632 *(2017).*
40. Goldwater, S., Griffiths, T. & Johnson, M. A bayesian framework for word segmentation: Exploring the effects of context. *Cognition* **112**, 21–54. https://doi.org/10.1016/j.cognition.2009.03.008 (2009).
41. Brown, S. & Heathcote, A. The simplest complete model of choice response time: Linear ballistic accumulation. *Cognit. psychol. Cogn. Psychol.* **57**, 153–78. https://doi.org/10.1016/j.cogpsych.2007.12.002 (2008).

42. Donkin, C., Brown, S. & Heathcote, A. Drawing conclusions from choice response time models: A tutorial using the linear ballistic accumulator. *J. Math. Psychol.* **55**, 140–151 (2011).
43. Verwey, W. *et al.* Buffer loading and chunking in sequential keypressing. *J. Exp. Psychol.* **00**, 544–562. https://doi.org/10.1037//0096-1523.22.3.544 (1996).
44. Du, Y. & Clark, J. New insights into statistical learning and chunk learning in implicit sequence acquisition. *Psychon. Bull. Rev.* **24**, 1225–1233 (2017).
45. Minier, L., Fagot, J. & Rey, A. The temporal dynamics of regularity extraction in non-human primates. *Cogn. Sci.* **40**, 1019–1030. https://doi.org/10.1111/cogs.12279 (2016).

## Acknowledgements
We thank Peter Dayan, Felix Wichmann, and Mirko Thalmann for helpful discussions. This work was supported by the Max Planck Society.

## Author contributions
Conceptualization: S.W., N. É., I.D., E.S. Formal analysis: S.W., E.S. Software: S.W. Visualization: S.W., N. É. Writing-original draft: S.W., E.S. Writing-review & editing: S.W., N. É., I.D., E.S.

## Funding
Open Access funding enabled and organized by Projekt DEAL.

## Competing Interests
The authors declare no competing interests.

## Additional information
**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1038/s41598-023-31500-3.

**Correspondence** and requests for materials should be addressed to S.W.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2023

# Supporting Information: Chunking as a rational solution to the speed-accuracy trade-off in a serial reaction time task

**Shuchen Wu**[1,*], **Noémi Éltető**[2], **Ishita Dasgupta**[3], **and Eric Schulz**[1]

[1]MPRG Computational Principles of Intelligence, Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[2]Department of Computational Neuroscience, Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[3]Google DeepMind, New York City, New York, USA
[*]shuchen.wu@tue.mpg.de

## Calculation of Chunky Boost

Figure 1 illustrates how Chunky Boost is calculated from reaction time data. As an example, when the relevant chunk is AB, we took the rt of pressing B (within-chunk press). The Cohen's $d$ was measured on the speed-up between the baseline and the test block of such within chunk key presses:

$$d_{AB} = \frac{\bar{rt}_{baseline} - \bar{rt}_{test}}{\sigma} \tag{1}$$

$\sigma$ was the standard deviation of rt.

The control reaction times came from size 2 subsequences that did not start with A, and the reaction times were measured on the second item of the subsequence, as illustrated in Figure 1. Cohen's $d$ was again evaluated as:

$$d_{AB_{control}} = \frac{\bar{rt}_{baseline} - \bar{rt}_{test}}{\sigma} \tag{2}$$

The chunky boost was calculated by subtracting the control from the Cohen's d measured on relevant chunks:

$$\Delta d = d_{AB} - d_{AB_{control}} \tag{3}$$

For Experiment 2, the relevant chunks were AB, BC, CD, and DA, the control subsequences were all size 2 subsequences excluding the relevant chunks.

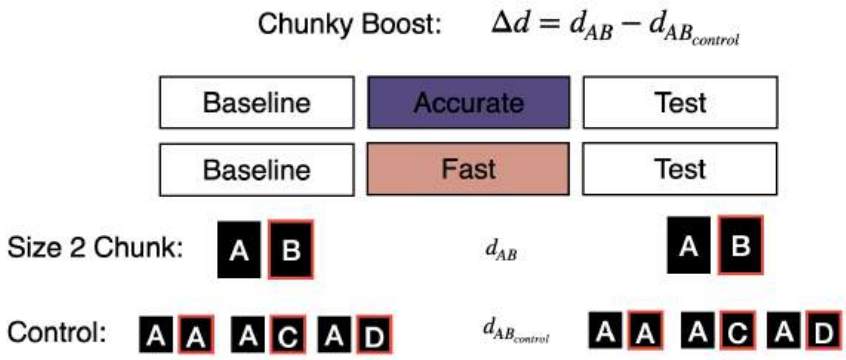


**Figure 1.** An illustration of the measurement of chunky boost.

## Calculation of Chunkiness

When the relevant size 3 chunk was ABC, the Wasserstein distance was evaluated on $rt_B$ and $rt_C$ separately as $Wasserstein(rt_B, rt_C)$. The chunkiness measure then was

$$\Delta W_{ABC} = Wasserstein(rt_B, rt_C)_{baseline} - Wasserstein(rt_B, rt_C)_{test} \tag{4}$$

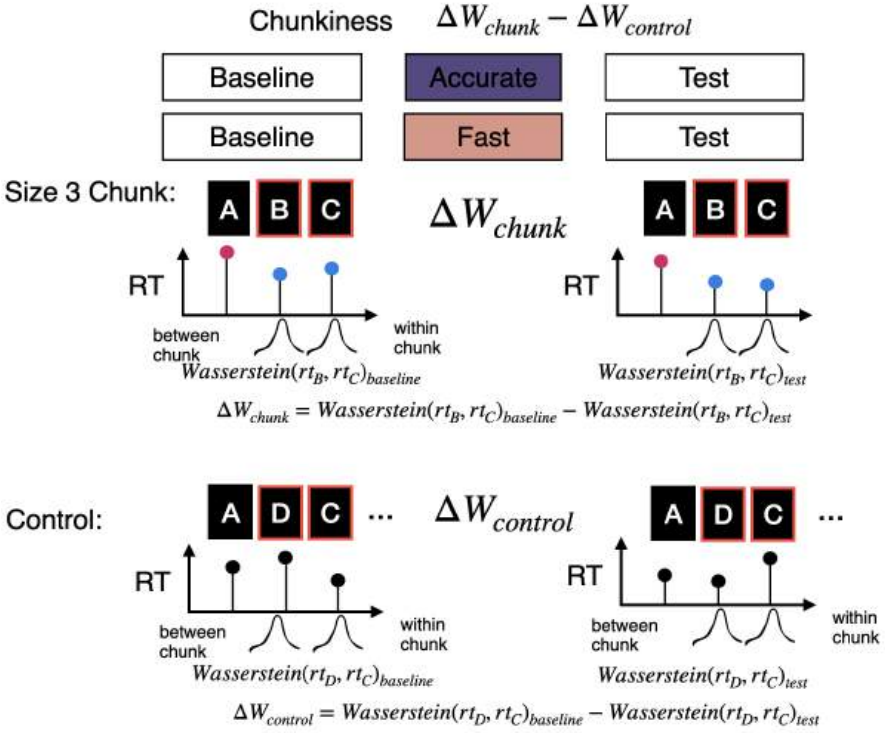


**Figure 2.** An illustration of the measurement of chunkiness.

The control chunks were subsequences that were not ABC, and their chunkiness measure became:

$$\Delta W_{control} = Wasserstein(rt_2, rt_3)_{baseline} - Wasserstein(rt_2, rt_3)_{test} \tag{5}$$

$rt_2$ and $rt_3$ denote the second and the third reaction time of the size 3 subsequences used as control. The measure of chunkiness is the control subtracted from the relevant chunks.

$$Chunkiness = \Delta W_{ABC} - \Delta W_{control} \tag{6}$$

For Experiment 2, the relevant chunks were ABC, BCD, CDA, DAB, and the controls were all size 3 subsequences that were not any of the chunks. Chunkiness was evaluated in the same way.

## Mixture of Gaussians Model

The mixture of Gaussians method categorizes the reaction time profile of each participant into within- or between-chunk reaction time as illustrated in Figure 3a. For each participant, the reaction time distribution was fit via a mixture of three Gaussians. When the likelihood of belonging to the mixture component with the lowest mean exceeds that of the other two components, the reaction time is classified as within-chunk.

When fitting the mixture of Gaussians model to individual participants' data, we separately varied the number of mixtures from 1 to 3 and used the Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) to evaluate the three models. Shown in Figure 3b is the distribution of the best fitting number of mixtures over all participants across the two experiments. For most participants and both evaluation criteria, the reaction time distribution was best described using 3 mixture components.

## Validation via Simulated RT Distributions

To perform a validation study on this method, we simulated within- and between-chunk reaction time distributions by generating two exponentially modified Gaussian distribution (ex-Gaussian) with distinct mean and standard deviations. The exponentially modified Gaussian distribution normally provides a good fit for the reaction time distributions, as the simulated RT is straightly

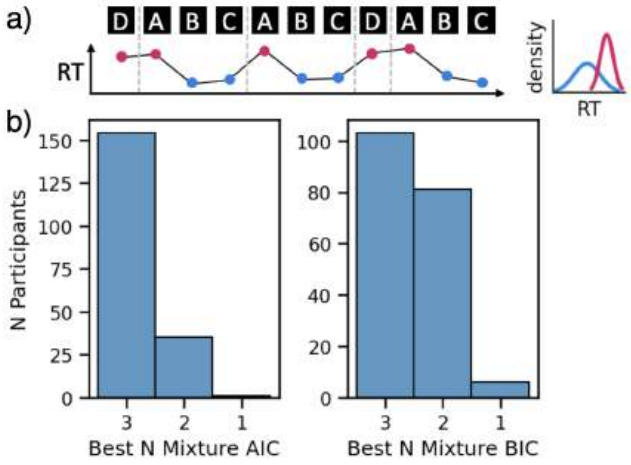


**Figure 3.** Mixture of Gaussians model for reaction time classification. a) Individual participant's reaction time of reacting upon key-press instructions across time. The distribution (right) is modelled as a mixture of two or three Gaussians. b) Histogram of the number of mixtures that led to best model selection criteria across all participants. Top: Best number of mixtures based on BIC (Bayesian Information Criteria). Bottom: Best number of mixtures based on AIC (Akaike Information Criteria).

positive, and the exponential decay component takes account of the observed right-skewness of RT distributions[1,2]. The ex-Gaussian distribution has the following probability density function:

$$f(x;\mu,\sigma,\lambda) = \frac{\lambda}{2} e^{\frac{\lambda}{2}\left(2\mu+\lambda\sigma^2-2x\right)} erfc\left(\frac{\mu+\lambda\sigma^2-x}{\sqrt{2}\sigma}\right) \tag{7}$$

$erfc$ is the complementary error function defined as:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt \tag{8}$$

$\mu$ is the mean of the independent Gaussian variable, $\sigma$ is the standard deviation, and $\tau$ is the mean of the exponential component. The first three moments of the resulting ex-Gaussian distribution are $\mu+\tau$, $\sigma^2+\tau^2$ and $2\tau^3$.

We fixed the dispersion parameter ($\lambda$) to be 1. We varied the difference in the mean $\mu$ between the within and between-chunk distributions, in addition to the spread parameter $\sigma$. After randomly interspersing samples coming from the two distributions across 1000 trials (the same length as in the experiments), we used the mixture of three Gaussians to classify the simulated rt data. Plotted in Figure 4 is the classification accuracy with increasing mean differences between the two simulated rt distributions varying spread parameters $\sigma$ ($\sigma_w$ for within-chunk and $\sigma_b$ for between-chunk distribution). Note that for most of the parameters, the worst classification accuracy is above 75%.

## Chunk Growth Rate Experiment 1

After obtaining the chunking profile of every participant as reflected in their reaction time speed-ups (as demonstrated in Figure 5), we evaluated the growth rate of chunks in the three groups of participants by looking at how fast chunk size increases (Figure 7). Figure 9 shows the rate of chunk increase between the three groups. Fitting a linear mixed-effects regression onto participants' chunk size, assuming random intercepts for each participant, showed a significant effect of trial number ($\chi^2(1) = 51.22, p < 0.001$). Generally, the rate of increase was positive, suggesting that the chunk size acquired by participants grew as a function of time. The chunk increase rate also differed significantly between the three conditions. The size 2 group ($\hat{\beta} = 1.43 \times 10^{-4}, t(84700) = 2.38, p = 0.02$) and the size 3 group had a significantly higher rate of chunk size increase compared to the independent group ($\hat{\beta} = 1.36 \times 10^{-4}, t(84700) = 2.93, p = 0.003$). To summarize, the chunk size acquired by participants increased over time and differed across the three groups. This observation is consistent with the design of the rational chunking model which assumes that participants reuse the previously learned chunks to construct longer chunks during the task.

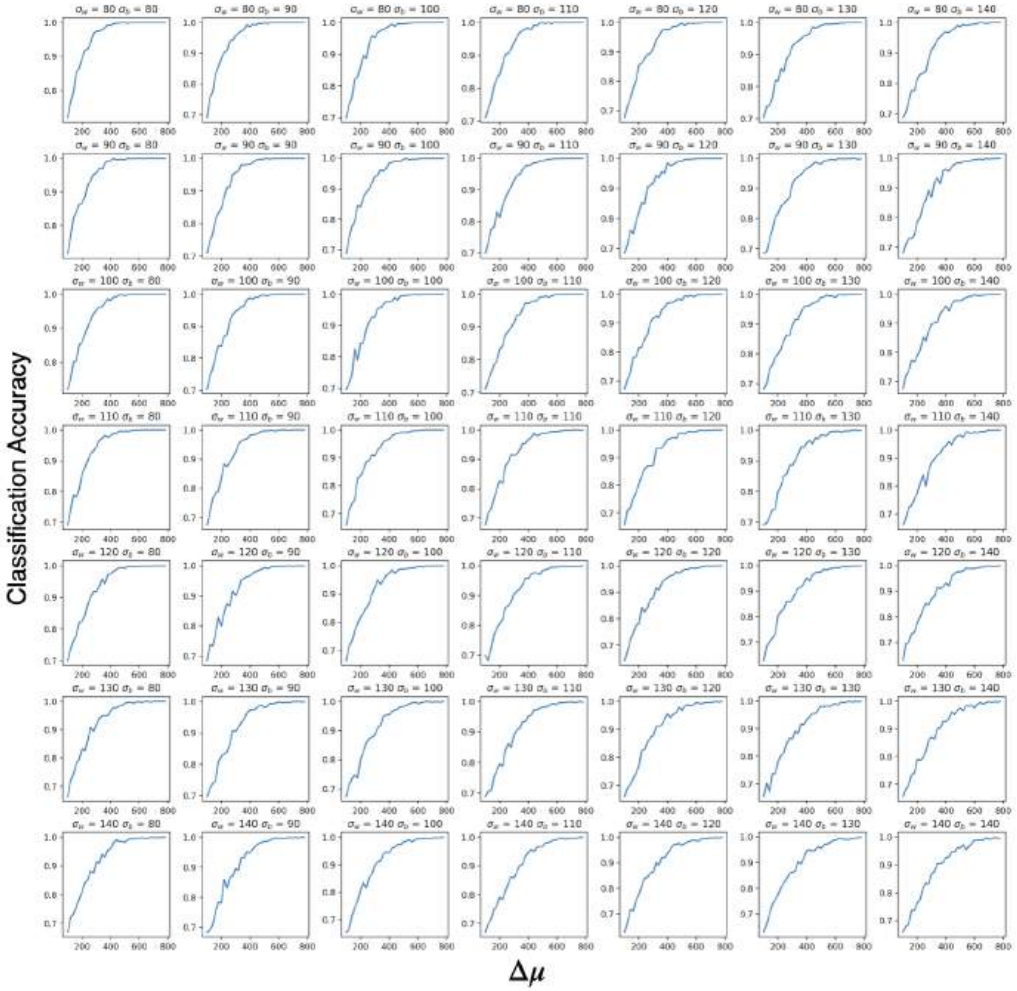


**Figure 4.** Classification accuracy (y-axis) of the mixture of Gaussians model on simulated reaction time data with increasing difference in mean $\Delta\mu$ (x-axis) and varying the standard deviation of the within-chunk ($\sigma_w$) and between-chunk ($\sigma_b$) distributions. $\sigma_w$ increases from the top to the bottom, and $\sigma_b$ increases from the left to the right.

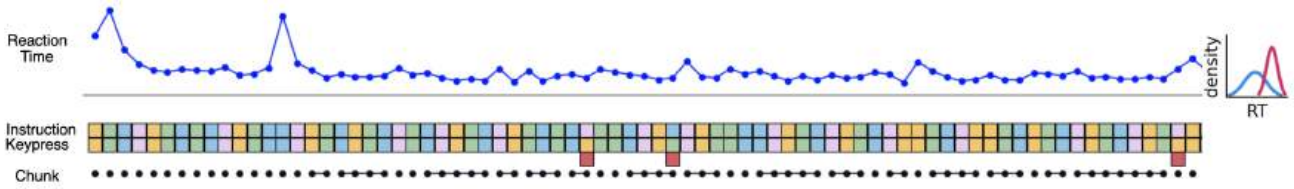


**Figure 5.** Chunk learning profile of one participant. The first, second, and third row show reaction times, instruction displayed, and this participant's key press (A, B, C, D, are separately color-coded by green, blue, magenta and orange boxes). We look at the exact chunks used by participants by classifying within and between chunk reaction time using the mixture of Gaussian method (see section method). When the participant pressed an incorrect key, that trial is marked by a red box in the fourth row. Using the distribution of reaction time data accumulated for this participants over all trials, we classify individual trials into within or between-chunk key presses, based on the likelihood assigned by the mixture model. Chunks learned by this participant are marked by connecting each between-chunk trial with the subsequent within-chunk trials, displayed by connected black dots in the last row.

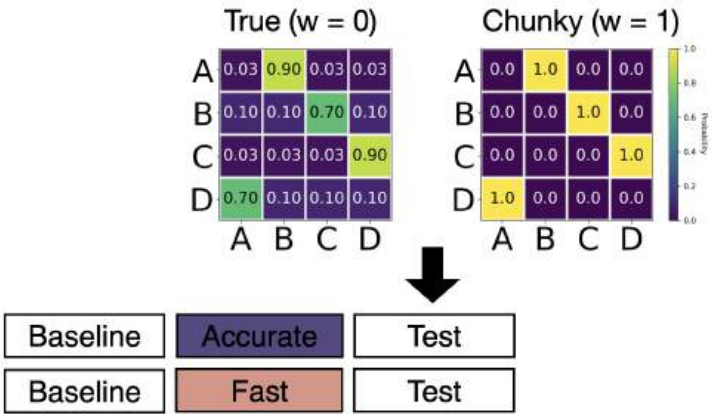


**Figure 6.** Transition probability regression.

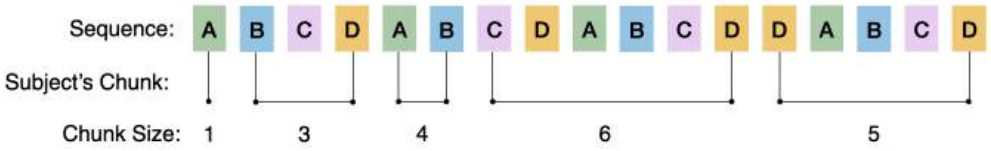


**Figure 7.** Measurement of chunk size.

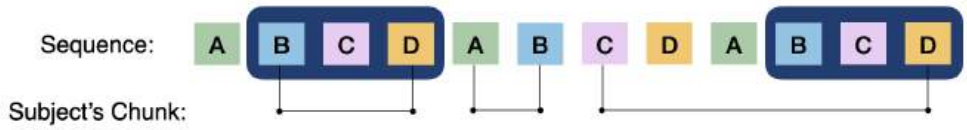


**Figure 8.** Chunk reuse measurement. After the classification of chunks, we look back in history to evaluate the probability of each chunk reusing the components of the previous chunks. In this case, chunk CDABCD is reusing the previous chunk BCD.

## Chunk Growth Rate Experiment 2

We also evaluated the rate of increase of every participant in experiment 2, as illustrated in Figure 9. Fitting a linear mixed-effects regression onto participants' chunk size assuming random intercepts of participants also showed a significant effect on trial number ($\chi^2(1) = 778.82, p < 0.001$). Additionally, there was an interaction effect between the groups and trial numbers. The fast group had a higher chunk learning rate than the accurate group ($\hat{\beta} = 2.68 \times 10^{-4}, t(62870) = 4.887, p < 0.001$). This result suggests that the fast group had a higher tendency to build chunks than the accurate group, which is consistent with model prediction.


## Acknowledgements

We thank Peter Dayan, Felix Wichmann, and Mirko Thalmann for helpful discussions. This work was supported by the Max Planck Society.


## Code and Data Availability Statement

The data collected and code used for analyzing this study can be found in this github repository: https://github.com/swu32/experimental_chunking

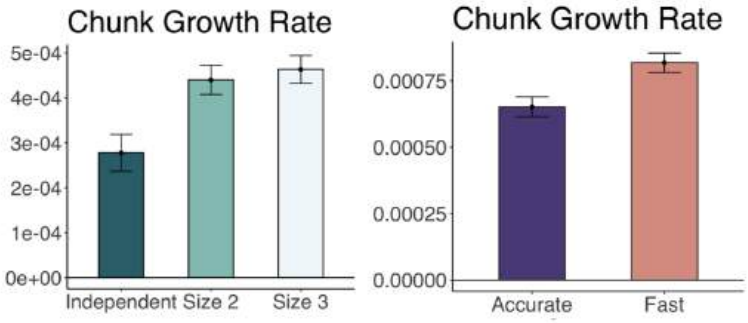


**Figure 9.** Average chunk growth rate for Experiment 1 (left) and Experiment 2 (right)

## Additional Information

The authors have declared that there are no competing interests.

## References


1. Heathcote, A. Rtsys: A dos application for the analysis of reaction time data. *Behav. Res. Methods, Instruments, & Comput.* **28(3)**, 427–445, DOI: 10.3758/BF03200523 (1996).

2. Burbeck, S. L. & Luce, R. D. Evidence from auditory simple reaction times for both change and level detectors. *Percept. & Psychophys.* **32(2)**, 117–133, DOI: 10.3758/BF03204271 (1982).


## Author Contributions

**Conceptualization:** Shuchen Wu, Noémi Éltető, Ishita Dasgupta, Eric Schulz.
**Formal analysis:** Shuchen Wu, Eric Schulz.
**Software:** Shuchen Wu.
**Visualization:** Shuchen Wu, Noémi Éltető.
**Writing – original draft:** Shuchen Wu, Eric Schulz.
**Writing – review & editing:** Shuchen Wu, Noémi Éltető, Ishita Dasgupta, Eric Schulz.

# Learning Structure from the Ground-up—Hierarchical Representation Learning by Chunking

# Learning Structure from the Ground up—Hierarchical Representation Learning by Chunking

**Shuchen Wu**
Computational Principles of Intelligence Lab
Max Planck Institute for Biological Cybernetics
Tübingen, Germany
shuchen.wu@tuebingen.mpg.de

**Noémi Éltető**
Department of Computational Neuroscience
Max Planck Institute for Biological Cybernetics
Tübingen, Germany
noemi.elteto@tuebingen.mpg.de

**Ishita Dasgupta**
Computational Cognitive Science Lab
Department of Psychology
Princeton University
dasgupta.ishita@gmail.com

**Eric Schulz**
Computational Principles of Intelligence Lab
Max Planck Institute for Biological Cybernetics
Tübingen, Germany
eric.schulz@tuebingen.mpg.de

## Abstract

From learning to play the piano to speaking a new language, reusing and recombining previously acquired representations enables us to master complex skills and easily adapt to new environments. Inspired by the Gestalt principle of *grouping by proximity* and theories of chunking in cognitive science, we propose a hierarchical chunking model (HCM). HCM learns representations from non-i.i.d. sequential data from the ground up by first discovering the minimal atomic sequential units as chunks. As learning progresses, a hierarchy of chunk representations is acquired by chunking previously learned representations into more complex representations guided by sequential dependence. We provide learning guarantees on an idealized version of HCM, and demonstrate that HCM learns meaningful and interpretable representations in a human-like fashion. Our model can be extended to learn visual, temporal, and visual-temporal chunks. The interpretability of the learned chunks can be used to assess transfer or interference when the environment changes. Finally, in an fMRI dataset, we demonstrate that HCM learns interpretable chunks of functional coactivation regions and hierarchical modular and sub-modular structures supported by the neuroscientific literature. Taken together, our results show how cognitive science in general and theories of chunking in particular can inform novel and more interpretable approaches to representation learning.

## 1 Introduction

Sequential data in our everyday life is often hierarchically structured. From streaming this sequential sensory perceptual data, we can identify repeated patterns – and bootstrap these to recognize higher order patterns. In cognitive science, identifying repeated, invariant patterns from sequences in units is known as *chunking*. To get an intuition for chunking, try to read through the following sequence

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

of letters: "DFJKJKJKDFDFJKJKDFDF". Upon reaching the end, if you were asked to repeat the letters from memory, you might recall fragments of the sequence such as "DF" or "JK". By parsing the sequence of letters only once, you have already detected frequently occurring patterns and memorized them together as units, i.e. *chunks*. Chunking has been observed in a range of sensory and behavioral modalities including language learning [1, 2], action organization [3, 4] and visual perception of structures [5–7]. Chunking as a mechanism is a basis for humans to identify patterns as objects, assigning labels to them to facilitate memory compression [8, 9], sequence prediction [10, 11], communication [12, 13], and generalization[14]. Learning hierarchical representations of the world is a feature central to human intelligence.

Despite recent success, deep learning models, on the other hand, do not represent explicit hierarchies. Neural networks contain sub-symbolic, nested, non-linear structures whose prediction processes are hard to comprehend. This lack of interpretability raises concerns over their fairness, privacy, robustness and trust-worthiness [15–18] and manifests itself as a key shortcoming of these models [19, 20]. To address these shortcomings, researchers have urged to seek inspiration from cognitive science to construct models that resemble the hierarchical and interpretable representations as observed in human learners [21, 22]. We take a two-fold approach to this problem. First, instead of learning from iid data, we ask: what if the time series data that comprises streams of perception comes from a hierarchical structure? Under this assumption, what could be an algorithm that learns the embedded hierarchical structure? We take inspiration from models in cognitive science showing that people perceive structures based on the Gestalt principle of *grouping by proximity*, and formulate a generic hierarchical pattern discovery algorithm that enables the rational discovery of structures with embedded hierarchies. We refer to this model as the hierarchical chunking model (HCM).

HCM starts out learning a minimal set of units sufficient to explain the sequence and gradually combines these units into increasingly larger and more complex chunks, constructing interpretable hierarchical structures. We derive learning guarantees on an idealized generative model and demonstrate convergence on sequential data coming from this generative model. Thereby, Gestalt principles of grouping can be understood as a rational way of learning representations from sequences with an inherent hierarchical structure. We then show that HCM resembles more to human chunking in qualitative ways compared to a recurrent neural network and flexibly transfers components learned from one task to another. We extend HCM to the visual-temporal domain capable of learning visual-temporal parts and wholes from higher dimensional sequential data. Taking it one step further, we deploy HCM to learn from high-dimensional fMRI data, which exerts a hierarchical structure. We demonstrate HCM's interpretable feature extraction ability to discover submodules of brain activations directly linkable to behavior supported by the literature.

## 2 Hierarchical Chunking Model

We define a chunk as a unit created by concatenating several atomic sequential units together. Taking the training sequence shown in Figure 1a as an example, the sequence is made up of discrete atomic units from an atomic alphabet set $\mathbb{A}_0$: in this case $\mathbb{A}_0 = \{0, 1, 2\}$. A chunk $c$ is made up of a combination of one or more atomic units in $\mathbb{A}_0 \backslash \{0\}$. 0 denotes an empty observation in the sequence.

Intuitively, if a sequence contains inherent hierarchical structure, then there are patterns which span several sequential units sharing these internal structures, examples of such sequences are repeated melodies and sub-melodies in music. If the pattern occurs in the sequence, observations between sequential units within the pattern will be correlated. In this case, chunking patterns within a sequence as units simplifies perceptual processing in the sense that the sequence can be perceived one chunk after another, instead of one sequential unit at a time. Furthermore, the acquired "primary" chunks serve as building blocks to discover larger chunks that are embedded within the hierarchy of the sequential structure.

Formally, HCM acquires a belief set $\mathbb{B}$ of chunks, and uses chunks from the belief set to parse the sequence. HCM assumes that a sequence is generated from samples of independently occurring chunks with probability of $P_{\mathbb{B}}(c)$ evaluated on the belief set $\mathbb{B}$. The probability of observing a sequence of parsed chunks $c_1, c_2, ..., c_N$ can be denoted as $P(c_1, c_2, ..., c_N) = \prod_{c_i \in \mathbb{B}} P_{\mathbb{B}}(c_i)$. Chunks as perceiving units serve as independent factors that disentangle observations in the sequence.

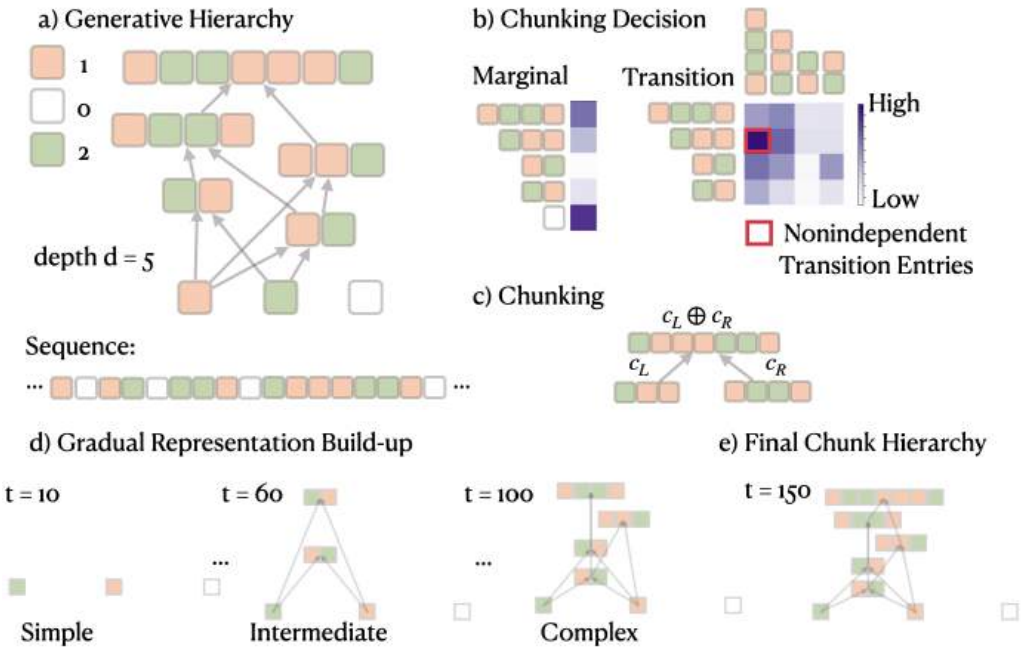


Figure 1: The Hierarchical Chunking Model. **a)** Example of a hierarchical model generating training sequences. **b)** Intermediate representation of learned marginal and transition matrices. The most frequent transition that violates the testing criterion is marked in red and can be turned into a new chunk. **c)** HCM combines the two chunks $c_L$ and $c_R$ to form a new chunk. **d)** As HCM observes longer sequences, it gradually learns a hierarchical representation of chunks. **e)** HCM arrives at the finally chunk hierarchy isomorphic to the generative hierarchy.

The training sequence is parsed by HCM in chunks. At every parsing step, the longest chunk in the belief set consistent with the upcoming sequence is chosen to explain the up-coming sequential observations. The end of the previous parse initiates the next parse.

Observing a hierarchically structured sequence as illustrated in Figure 1a, HCM gradually builds up a hierarchy of chunks starting from an empty belief set. It first identifies a set of atomic chunks to construct its initial belief set $\mathbb{B}$. Initially, these will be chunks of length one, yielding one-by-one processing of the primitive elements.

For one belief set $\mathbb{B}$, HCM keeps track of the marginal parsing frequency $\boldsymbol{M}(c_i)$ for each chunk $c_i$ in $\mathbb{B}$, a vector with size $|\mathbb{B}|$ and the transition frequency $\boldsymbol{T}$ between chunk $c_i$ followed by chunk $c_j$, as illustrated in Figure 1b. Entries in $\boldsymbol{M}$ and $\boldsymbol{T}$ are used to test the hypothesis that consecutive chunk parses have a correlated consecutive occurrence within the sequence via a $\chi^2$-independence test. If two chunks $c_L$ and $c_R$ have a significant adjacency dependence based on their entries in $\boldsymbol{M}$ and $\boldsymbol{T}$, they are chunked together to become $c_L \oplus c_R$, which augments the belief set $\mathbb{B}$ by one. One example of chunk merging is shown in Figure 1c.

**Independence Test** We use a $\chi^2$-test of independence to assess the correlation of consecutive occurrences of $c_L$ followed by $c_R$. Let $c_L$ be an indicator variable that is 1 when chunk $c_l$ is parsed and 0 otherwise, similarly we formulate $c_R$ as another indicator variable of parsing the chunk $c_r$. We evaluate the $\chi^2$-value as a criterion to reject the null hypothesis that the consecutive observation of $c_l$ followed by $c_r$ is statistically independent:

$$\chi^2 = \sum_{c_L=\{0,1\}} \sum_{c_R=\{0,1\}} \frac{N(p(c_L, c_R) - p(c_L)p(c_R))^2}{p(c_L)p(c_R)}$$

$p(c_L, c_R)$ and $p(c_L)p(c_R)$ are evaluated from $\boldsymbol{M}$ an $\boldsymbol{T}$. The degree of freedom is 1. A $\chi^2$-probability of less than 0.05 is the criterion to reject the null hypothesis (i.e. that $c_l$ and $c_r$ occur independently).

There are two versions of HCM. The Rational Chunk Learning HCM learns chunks in an idealized way which we use to study learning guarantees. The online version of HCM is an approximation

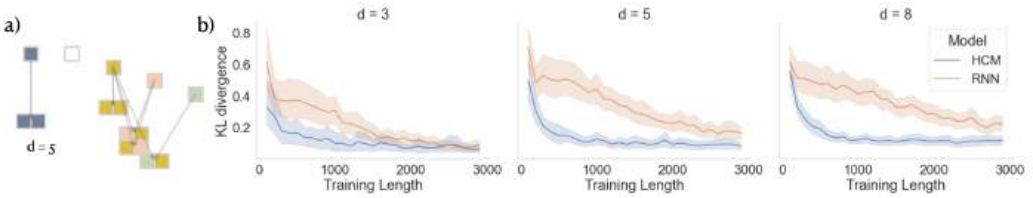


Figure 2: **a)** Example graph generated from the hierarchical generative model with a depth of $d = 5$. **b)** Learning performance of HCM and RNN with increasing training length and for increasing depths. Performance was averaged over 30 randomly-generated graphs.

to the rational HCM that can be adapted to different environments and processes sequences online. Pseudo-code for both algorithms can be found in the SI.

**Rational Chunk Learning: HCM as an Ideal Observer** HCM is initiated with an empty belief set and it first finds a minimally complete belief set after the first sequence parse. In each iteration, the entire sequence is parsed to evaluate $\boldsymbol{M}$ and $\boldsymbol{T}$, which are used to find consecutive chunk parses in the existing belief set that violate the independence testing criterion. From these dependent chunk pairs, the pair with the largest estimated joint probability is combined into a new chunk. The new chunk enlarges the belief set by one. The chunks in the new belief set are used to parse the sequence in the next iteration. This process repeats until all of the chunks in the belief set pass an independence halting criterion, which measures if all of the chunks in the belief set are currently independent, again assessed via a $\chi^2$-test (see SI).

**Online Chunk Learning** The online chunk learning HCM approximates the ideal observer HCM by learning new chunks when the training sequence is processed on the go. To have a feature that encourages adaptation to new environmental statistics, entries in $\boldsymbol{M}$ and $\boldsymbol{T}$ can be subject to memory decay. We use the ideal observer model to demonstrate learning guarantees, but use the online model to learn representations in realistic and more complex set-ups.

### 2.1 HCM Learns Representations from the Ground Up

As HCM learns from a sequence, it starts with no representation and gradually builds up interpretable representations described by a chunk hierarchy graph $\hat{\mathcal{G}}$ with the vertex set being the chunks and edges pointing from constituents to composites. Shown in Figure 1d is the gradual build-up of one such graph as the model learns from a training sequence coming from the generative hierarchy in Figure 1a. At $t = 10$, HCM learns only the atomic chunks, at $t = 60$, HCM has already constructed two additional chunks; when $t = 100$, two more additional chunks are constructed. HCM arrives at the final chunk hierarchy at $t = 150$.

## 3 Generating Sequences with a Hierarchical Structure

We construct a generative model to study HCM's behavior formally and empirically. The generative model constructs random chunk hierarchies from which non-iid sequences are sampled. Such graph $\mathcal{G}_d$ contains vertex set $V_{\mathbb{A}_d}$ and edge set $E_{\mathbb{A}_d}$ to describe the relation between chunks and their constituents. One example is illustrated in Figure 1a. $\mathbb{A}_d$ is the set of chunks used to construct the sequence. The depth $d$ specifies the number of chunks created in the generative process.

Starting with an initial set of atomic chunks $\mathbb{A}_0$, at the i-th iteration, two chunks $c_L$, $c_R$ are randomly chosen from the current set of chunks $\mathbb{A}_i$ and are concatenated into a new chunk $c_L \oplus c_R$, augmenting $\mathbb{A}_i$ by one to $\mathbb{A}_{i+1}$. Meanwhile, an independent occurrence probability is assigned to each chunk under the constraint that the probability of occurrence for every new chunk $c_i$ in the construction process evaluated on the support set $\mathbb{A}_i$ carries the largest probability mass.

Once a graph hierarchy is constructed, we construct non-iid observational sequences by consecutively sampling chunks from the hierarchy with their corresponding probability, under the constraint that no two chunks with a child chunk are sampled consecutively.

### 3.1 Learning Guarantee

**Theorem**: As the length of the sequence approaches infinity, HCM learns a hierarchical chunking graph $\hat{\mathcal{G}}$ isomorphic to the generative hierarchical graph $\mathcal{G}$.

*Proof Sketch*: We approach this proof by induction. Further details can be found in the SI. Base step: The first step of the rational chunking algorithm is to find the minimally complete atomic set of chunks to form its initial belief set. This procedure guarantees that $\hat{\mathcal{G}}_0 = \mathcal{G}_0$. Additionally, the probability mass of the learning model at step 0 and the generative model at step 0 is asymptotically the same as the sequence length approaches infinity. Induction hypothesis: Assume that the learned belief set $\mathbb{B}_i$ at step $i$ contains the same chunks as the alphabet set $\mathbb{A}_i$ in the generative model, the chunk combination pair with the biggest evaluated joint occurrence probability violating the independence test is picked to be concatenated into a chunk to extend the belief set: this chunk is the same chunk node created by the hierarchical generative model. End step: The chunk learning process stops once the independence criterion is no longer violated. This is the case once the chunk learning algorithm has learned a belief set $\mathbb{B}_d = \mathbb{A}_d$.

### 3.2 Learning Convergence and Comparative Data-efficiency

To evaluate and show HCM's learning performance, we trained HCM to learn hierarchies of chunks from sequences generated by the hierarchical generative model. Shown in Figure 2 is HCM's learning performance as sequence length increases, averaged over 30 independently generated random graphs with the same depth $d$. One example of such graphs is shown in Figure 2a. Kullback-Leibler divergence was used to evaluate learning performance. To this end, learned hierarchies by HCM were used to generate sequences, which were then evaluated on the support set of the alphabet set in the generative model.

Figure 2b shows the KL-divergence between the learned and ground-truth distribution for increasing depths $d$ of the generative graphs. For each depth, the KL-divergence was evaluated on 30 random generative models with sequence length increasing from 50 to 3000. Overall, the KL-divergence decreased as the training sequence length increased and converged with longer training sequences, showing a closer representation resemblance to the generative model.

A similar training and learning evaluation was conducted on a 3-layer Recurrent Neural Network (RNN) with 40 hidden units for comparison. As the length of the training sequence increased, the KL-divergence of RNN converged at a slower rate than HCM. This competitive advantage in data efficiency became more pronounced with increasing depth of the generative hierarchy.

## 4 HCM Resembles Human Chunk Learning

Here we compare the chunk learning behavior of HCM to the learning characteristics of humans. To that end, we used data collected from a sequence learning study by [23] with 47 participants under the license CC-BY 4.0. As shown in Figure 3a, the training sequence comprised chunks ABC and D, independently occurring with equal probability. The study assessed how participants built up chunk knowledge gradually. Participants' reaction times reflected that, after enough training, they were anticipating several upcoming sequence elements, suggesting that they have acquired longer chunks (Figure 3b, left) [23].

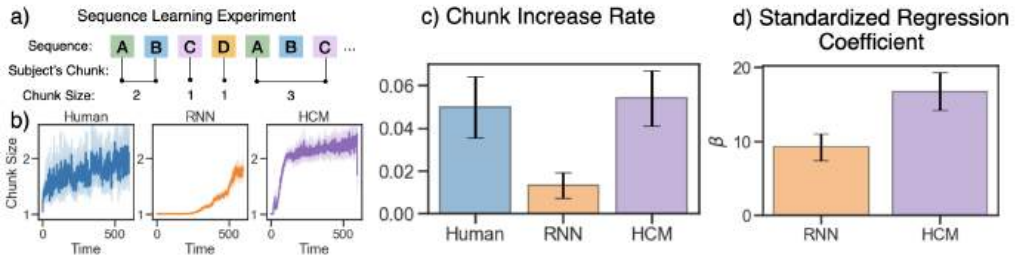


Figure 3: **a)** A sequence learning experiment with chunks ABC and D. **b)** Chunk size increase of human participants, RNN, and HCM during training. **c)** Average chunk increase rate during training. **d)** Regression coefficient of RNN and HCM's confidence estimates on human reaction time data.

In a similar vein, we measured online chunk size increase of HCM using the same method as in [23] and, for comparison, RNNs (see SI for further comparisons to other algorithms). HCM, similarly to humans, started learning longer chunks early in the sequence. By contrast, RNN did not start to chunk until after step 300, and when it started to learn chunks, the increase rate of the predictive horizon was not as steep as HCM's. Evaluating the average rate of chunk growth also showed that HCM builds up chunks as learning progresses was more similar to participants' than the RNN's (Figure 3c). The negative log-probabilities of sequence elements generated by the HCM and RNN were both significantly related to human reaction times (that reflect the certainty of their internal predictions [24]). Yet, the relationship was substantially stronger (Figure 3d) between HCM ($\beta = 16.74$, $p \leq 0.001$, $\tau = 0.165$, $BIC = 313586.5$) and human participants compared to that of the RNN ($\beta = 9.24$, $p \leq 0.001$, $\tau = 0.085$, $BIC = 314236.4$). These results suggest that HCM resembles human chunk learning more strongly than RNNs and can therefore be seen as the cognitively more plausible approach to hierarchical representation learning.

## 5 HCM Permits Transfer Between Environments

One characteristic of human learning is that previous learning experience facilitates and sometimes interferes with acquiring a new skill [25]. Having an interpretable representation can inform us about positive or negative transfer a priori.

An HCM has learned a chunking graph in Figure 4a from an environment. When it switches to another environment with a generative model overlapping with its previously acquired representation, it can reuse the learned subgraphs of chunks marked in gray as in Figure 4b and learns faster than a naive HCM in Figure 4c. Vice versa, transfer can be detrimental when there is no or little overlap between the learned chunking graph and the generative model of the new environment. For the same HCM as in Figure 4a, transferring to an environment with a chunking graph in Figure 4d implies learning the shaded chunks anew, in addition to running the risk of being misled by the previous representations. As a result, the early performance of the pre-trained HCM suffers more from an interfering environment than a naive model in Figure 4e. Interpretability of HCM's representations enables the assessment of facilitation or interference when the environment changes.

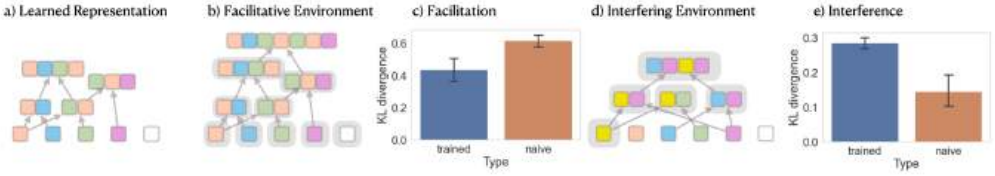


Figure 4: **a)** Example of a representation learned by an HCM. **b)** A facilitative environment with a generative model overlapping in its structure with the test environment. Gray shadows mark the chunks that can be directly transferred. **c)** Average performance over the first 500 trials after the environment switches. **d)** Interfering environment. Gray shadows marks chunks that need to be acquired anew. **e)** Average performance over the first 500 trials after the environment switches.

## 6 Generalizing to Visual Temporal Chunks via the Principle of Proximal Grouping

Humans excel at finding structures in hierarchical visual objects and grouped movements. The Gestalt principle of *grouping by proximity* suggests that we tend to group objects that are close to one another into a cohesive unit [26, 27]. This principle has been suggested to play a key role in human perceptual grouping [28], benefits working memory [29] and reduces visual complexity [30]. Indeed, in humans and other animals, learning of adjacent relationships prevails over non-adjacent ones [31]. Therefore, the adjacent dependency structure can be expanded to chunking in visual temporal domains [32]. To emulate this ability of chunking via proximal grouping, we extend HCM to learn visual temporal chunks.

Visual temporal chunks subsume temporal length and varying visual slices in each temporal slice (Figure 5a). One can imagine a visual temporal chunk as having a 3D shape — the first two

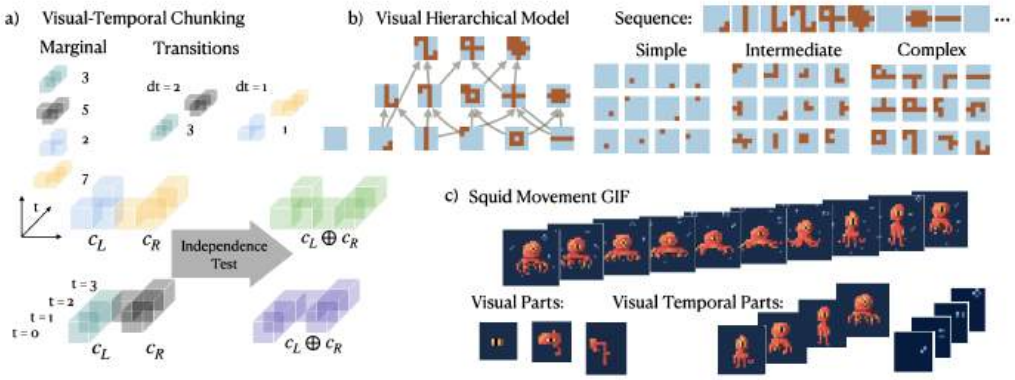


Figure 5: **a)** HCM learns visual-temporal chunks by extending the transition matrix to take account of time differences. **b)** Left: A visual hierarchical model where complex images are composed of simpler images. Right: Initial, intermediate, and complex chunks learned by HCM trained on sequences of images sampled from the visual hierarchical model.**c)** Top: A GIF of a moving visual used as a sequence to train HCM. Bottom: Examples of temporal and visual chunks learned by HCM.

dimensions are the visual part of the chunk, and the object's length is the temporal part, made of stacked visual-temporal pixels. Within each temporal slice are the visual features identified by the chunk. As the model iterates through data across its temporal slice, the chunk that attains the biggest visual temporal volume explains part of the observational sequence. Multiple visual temporal chunks can occur simultaneously. Starting at the visual temporal time point marked by the previous chunk, chunks are identified and stored in $\boldsymbol{M}$. The transition matrix $\boldsymbol{T}$ is modified to account for the temporal lag difference between adjacent chunk pairs within a proximity parameter and records the frequency that one chunk transitions into another for each time lag. Whenever a pair of adjacent chunks are identified, an independence hypothesis test evaluates whether the adjacent observation are correlated. Chunks that violate the hypothesis test are combined to parse future sequences.

**Learning Part-Whole Relationship Between Visual Components** We show HCM learned chunks in the visual domain from a sequence of independently sampled images. Figure 5b left shows a hierarchical generative model in the pixel-wise image domain. A set of elementary visual units in the lowest hierarchy level combines into intermediate and more complex visual units higher up in the hierarchy. All of the constructed elements in the hierarchy occurred independently according to a probability drawn from a Dirichlet flat distribution. Images in the hierarchy were independently sampled from the generative distribution to become the training sequence. In Figure 5b, right we show the chunk representations learned by HCM at different stages. Initially, HCM acquires the individual pixels as chunks to explain the observations. As HCM proceeds with learning, it discovers visual correlations among the pixels and constructs increasingly complex visual parts.

**Learning Visual-Temporal Movement Hierarchies** Instead of seeing one image after another sampled from an independent, identically distributed distribution, real-world experiences contain correlations in both the visual and temporal dimensions. From observing object movements across space and time, the visual system learns structures from correlated visual and temporal observations, decomposes motion structure, and groups moving objects together as a whole [33]. To emulate this type of environment, an animated GIF of a squid swimming in the sea (Figure 5c) was used as a visual-temporal sequence to train HCM. As learning advances, HCM learns chunks spanning both the visual and temporal domains. There are visual-temporal chunks that mark the movements of a tentacle and the rising-up motion of a bubble. Additionally, visual chunks resemble a part of the visual's eye and face. The meaningful chunks in the visual-temporal domain suggest the grouping principle enables the plausible learning of movement sequences and aids the perception of objects as wholes and their corresponding parts.

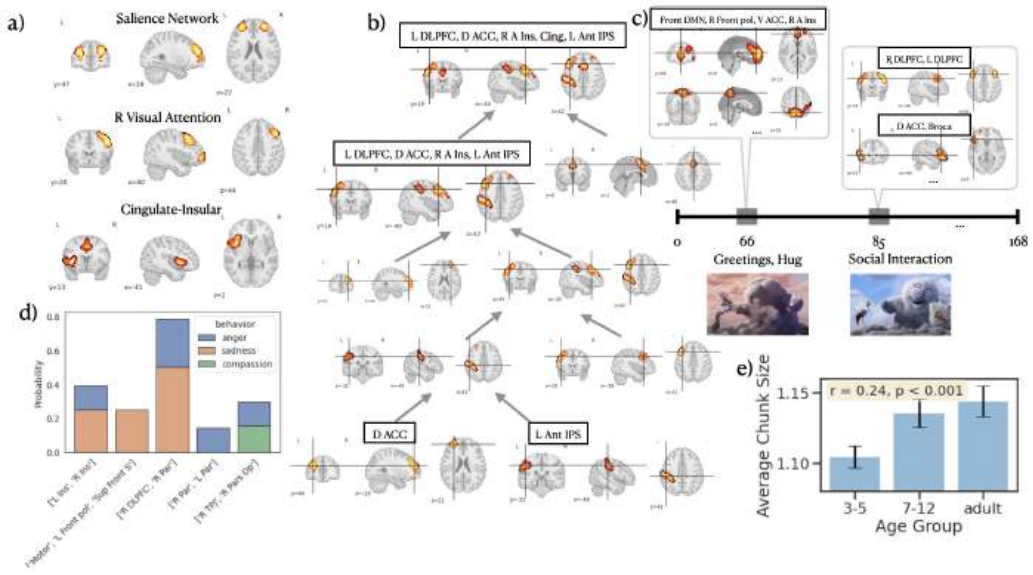


Figure 6: Application avenues of HCM on fMRI data. **a)** Example chunks of brain functional activation regions. **b)** HCM learns hierarchical functional network with bigger chunks emerging from its constituents. **c)** Chunk activation patterns responding to scene content **d)** Distinct response of retrieved chunks to tagged scenes. **e)** Average chunk size across age groups.

## 7 Learning Hierarchies of Brain Activation from Resting-state fMRI data

HCM learns hierarchies from structured sequential data. As brain activation has been suggested to be hierarchically structured [34], we demonstrate HCM's usefulness to learn structures in biological neural networks activating in response to complex stimuli by running HCM on a resting-state fMRI data set.

We used a developmental data set provided by the `nilearn` package with BSD License [35] and originally collected by [36] with its corresponding IRB approval. This data set contains the resting-state BOLD activity of 155 participants ranging from age 3 to 40, while watching the silent movie "Partly Cloudy". BOLD signal was extracted from functional brain regions defined by the MSDL Atlas [37], with confounds excluded and transformed into a rounded, normalized time series.

**HCM's Chunks Reflect Structural, Functional and Anatomical Connectivity** Figure 6 shows three typical examples of learned chunks for a randomly-chosen participant. The labels of functional regions come from the MSDL atlas [37]. The first example is the co-activation of D ACC and R A Ins. These two regions have been observed to co-activate in the presence of emotions, pain, and humor. They have been suggested to be a key hub of the salience network [38–41]. The second example chunk contains the activation of R DLPFC and R Front Pole. These regions belong to the visual attention network and are known to be anatomically connected [42–44]. A final example is the chunk of L Ins and Cing, which are also known to be anatomically and functionally connected [45]. Thus, the chunks discovered by HCM correspond to empirically-verified patterns of functional activity.

**HCM's Chunks Recover Hierarchical Activation Patterns** In fMRI data, hierarchies of chunk activation constructed by HCM reflect networks of functional regions. On the top of the hierarchy, the largest chunk contained L DLPFC, D ACC, R A Ins, Cing, and L Ant IPS (Figure 6b). Those regions are known to co-activate during cognitive tasks that demand attention, working memory, and control [41]. Chunks in the intermediate levels of the hierarchy reflect sub-networks of functional connectivity. Sub-chunks such as D ACC, R A Ins, L Ant IPS, and L DLPFC have been suggested to conjointly activate in cognitive effort-related activities [41]. Atomic chunks in the hierarchy such as D ACC and L Ant IPS activate individually sometimes without their parent chunks. Indeed, they have distinct functional signatures for affect processing [46, 47], and visual attention control [48]. Upon exposure to a time series in fMRI data, HCM constructs chunks from their constituents and arrives at a hierarchy of chunk relations, indicating nested network structures in the human brain.

**HCM's Chunks can be Matched with Stimulus Onsets** The retrieved chunks by HCM can be tagged with critical stimulus onsets. We tagged 19 critical moments involving social and emotional content in the movie. Figure 6c shows one example chunk activation upon stimulus onsets. Frontal DMN, right frontal pole, ventral anterior cingulate cortex, and right anterior insula activate together as a recurring unit after participants witness a scene with characters greeting and hugging each other. These regions have been suggested to be involved in social and cognitive processing [49]. Another example is the activation of areas known to be involved in emotion and language processing: D ACC and Broca [50], during a scene containing social interactions. In the meantime, the left and right prefrontal cortex, involved in theory-of-mind [36], also lights up.

We categorized the tagged moments into 3 groups of different emotional load: sadness, anger, and compassion. We then looked at the activation probability of retrieved chunks within the 6 seconds after watching those tagged scenes. Figure 6d shows a list of such chunks from one participant with their activation probability for each emotional category. For example, the left and right insula, known to be involved in affective processing [51], have a 0.4 activation probability after witnessing scenes of sadness or anger, but no activation after witnessing scenes of compassion. The same holds for R DLPFC and R Par that have been documented to activate in response to emotional conflict [52]. On the other hand, regions such as R TPJ and R pars opercularis that are involved in emotional reactions and theory of mind processes [53], activate in response to a scene of compassion or anger, but not to a scene of sadness. Thus, chunks of active brain regions can be related to complex stimuli, and regions activate selectively in response to one or more categories of emotional stimuli, but not others.

**HCM's Average Chunk size Correlates with Participants' Age** HCM can also be used to perform meaningful analyses at the population level. Specifically, we find that HCM's returned average chunk size per participant correlates significantly with age (Figure 6e). The older the participants are, the longer are the chunks found in their data ($r = 0.23$, $p \leq 0.001$). This discovery is in line with findings in the original study, which showed an increase in modularization of ToM and pain circuits across development [36].

To summarize, we applied HCM to learn chunks from a developmental fMRI data set. HCM enabled the discovery of spatially and temporally correlated activation chunks that are theoretically and empirically meaningful. The resulting chunks can be linked to complex stimuli and offer directly interpretable insights into the structure and function of brain activity.

## 8 Related Work

HCM extends upon decades of previous cognitive science and psychology research on chunking. In cognitive science, process models such as PARSER and competitive chunking were demonstrated to generate qualitatively similar chunks as in human sequence learning [54, 55]. HCM is a rational algorithm that learns the underlying chunks when the sequence is generated from a hierarchical chunking graph. Therefore, the chunking criterion is no longer a heuristic but a rational learning strategy that enables hierarchical structural discovery. On top of inheriting the merits of its predecessors, HCM generalizes the chunk learning principle to higher dimensional sequential domains such as visual-temporal sequential data.

HCM relates to several other lines of research. One is program induction. In program induction, explicit representations are acquired by searching for programmatic structures that best explain observational samples [22], and consolidating these offline with library learning [56]. However, domain expert knowledge is needed to specify the primitive programs; the relations and composition rules must adapt to the task settings and sensitively influences the quality of retrieved representations. Other approaches to structure learning include unsupervised parsing [57], which learns a stochastic and-or graph from sequential data. HCM is distinct in adapting its representation granularity to discover bigger chunks from data without pre-specifying the structure.

Another category of models to learn from sequential data are traditional sequence learning models including Hidden Markov Models (HMM), n-gram models and their variants to capture multi-scale sequential structure such as Hidden semi-Markov model [58] and hierarchical HMM [59]. The parameters of these models proliferate exponentially as a function of chunk length, implying memory inefficiency. Additionally, these models demand a structure specification before fitting parameters to the data. They also lack the adaptive recombination and reuse of pre-existing components. The

same issue is with neural network approaches to extract chunks from sequences [60–62]. Apart from lacking in interpretability, these models do not leverage the concatenation process observed in humans or reusing previously learned representations to construct new representations.

The principle of iterative merging of chunks has been used in compression algorithms, such as tokenizing methods in NLP, which were developed to optimize sequential data compression. Tokenizing methods such as Byte-Pair Encoding [63] iteratively merges the most frequent pairs of chunks to build a vocabulary of a text corpus. This objective is easy to compute but gives rise to ambiguous parses of the text (e.g. [AB, C] / [A, BC]). To minimize parse ambiguity, WordPiece [64] merges chunks that increase the likelihood of the corpus the most. However, the objective of WordPiece is expensive to compute. HCM circumvents the problem of computing the global sequence likelihood by instead maximizing the local chunk continuation likelihood. The computational efficiency of HCM makes it a plausible cognitive model of chunking as well as a promising method for NLP.

Probabilistic context-free grammars (PCFGs) are related to HCM in that they use trees as a representational form of sequences. The parse trees of PCFGs denote production rules, such as S $\rightarrow$ NP + VP. These production rules define how abstract syntactic units (non-terminals), such as a noun phrase and a verb phrase, are instantiated into a concrete string of words (terminals) to compose a sentence, such as 'we wrote the paper'. In comparison, the generative tree of HCM denotes statistical relationships among concrete chunks, such as 'we'-'wrote'-'the paper'. Extending HCM to represent abstraction is an exciting future direction, on which avenue the comparison to PCFGs will be instructive.

## 9 Discussion

Our work has its limitations. Currently, we fix the memory decay and the deletion threshold parameters to a priori plausible values. In future work, these parameters could be adapted online based on environment volatility. Another limitation is its scalability: at the moment, HCM learns representations from semi-high dimensional sequential data (i.e., currently between 1 to 625 dimensions). We are actively looking into generalizing this algorithm to higher dimensional data domains by combining it with existing neural network approaches or computer vision algorithms such as coherent point drift [65] or normalized cuts [66] to allow for the learning of ambiguous and high dimensional chunk exemplars. It is also possible to combine HCM with the compressed representation, such as the hidden activity of an auto-encoder to process and learn the structure from downstream representation. In this work, HCM learns one type of hierarchy of compound representations. However, we can show that HCM can be generalized to not only learn simple chunks but also chunks in projected spaces and thereby generalize between two chunks that contain the same motif (for example, "12221212" and "34443434"; see SI for detailed results). In the future, it might be worthwhile to further combine our approach with others amongst a taxonomy of representational hierarchies.

HCM also opens up other application directions. One direction is integrating HCM with deep neural network approaches as an interface between human understanding and distributed computation. Learning hierarchies of coherent activations from intermediate hidden units has the potential to reveal neural networks' underlying computation structure. Furthermore, it is also possible to equip HCM with additional top-down encoded representations, for example, by pre-training on other sequences or by adjusting the chunks by hand before the training starts. Another direction in neuroscience or behavioral research is to learn chunks of tagged animal movements that enable insights into the emergence of behavioral structure [67]. Finally, finding patterns that form as a cognitive unit is a vital task for infants to learn about the structures of the world and resembles the process of formulating a scientific theory from observation [68]. HCM can function as one means to come up with world models by observation, ready for experimental interventions or active learning to delineate the causal structure within [69].

## 10 Conclusion

We have proposed a hierarchical chunking model (HCM) that learns chunks from non-iid sequential data with a hierarchical structure. HCM starts out learning an atomic set of chunks to explain the sequence and gradually combines them into increasingly larger and more complex chunks. The output of the model is a dynamical graph that is a trace of the evolving representation. The resulting representations are easy to interpret, and flexibly reusable.

## References


[1] Pierre Perruchet, Bénédicte Poulin-Charronnat, Barbara Tillmann, and Ronald Peereman. New evidence for chunk-based models in word segmentation. *Acta psychologica*, 149:1–8, 2014.

[2] Stewart M McCauley and Morten H Christiansen. Computational investigations of multiword chunks in language learning. *Topics in Cognitive Science*, 9(3):637–652, 2017.

[3] Virginia B. Penhune and Christopher J. Steele. Parallel contributions of cerebellar, striatal and M1 mechanisms to motor sequence learning, 2012. ISSN 01664328.

[4] David A. Rosenbaum, Sandra B. Kenny, and Marcia A. Derr. Hierarchical control of rapid movement sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 1983. ISSN 00961523. doi: 10.1037/0096-1523.9.1.86.

[5] Geoffrey Hinton. Some demonstrations of the effects of structural descriptions in mental imagery*. *Cognitive Science*, 3(3):231–250, 1979. doi: https://doi.org/10.1207/s15516709cog0303_3. URL `https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog0303_3`.

[6] Timothy F. Brady, Talia Konkle, and George A. Alvarez. Compression in Visual Working Memory: Using Statistical Regularities to Form More Efficient Memory Representations. *Journal of Experimental Psychology: General*, 138(4), 2009. ISSN 00963445. doi: 10.1037/a0016797.

[7] Dennis E. Egan and Barry J. Schwartz. Chunking in recall of symbolic drawings. *Memory & Cognition*, 7(2), 1979. ISSN 0090502X. doi: 10.3758/BF03197595.

[8] Fernand Gobet, Peter C.R. Lane, Steve Croker, Peter C.H. Cheng, Gary Jones, Iain Oliver, and Julian M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5(6), 2001. ISSN 13646613. doi: 10.1016/S1364-6613(00)01662-4.

[9] George A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 1956. ISSN 0033295X. doi: 10.1037/h0043158.

[10] Iring Koch and Joachim Hoffmann. Patterns, chunks, and hierarchies in serial reaction-time tasks. *Psychological Research*, 63(1), 2000. ISSN 14302772. doi: 10.1007/PL00008165.

[11] Diana Mussgens and Fredrik Ullén. Transfer in motor sequence learning: Effects of practice schedule and sequence context. *Frontiers in Human Neuroscience*, 11 2015. doi: 10.3389/fnhum.2015.00642.

[12] Ludwig Josef Johann Wittgenstein. *Philosophical Investigations*. New York, NY, USA: Wiley-Blackwell, 1953.

[13] Eric Schulz, Francisco Quiroga, and Samuel J Gershman. Communicating compositional patterns. *Open Mind*, 4:25–39, 2020.

[14] Eric Schulz, Joshua B. Tenenbaum, David Duvenaud, Maarten Speekenbrink, and Samuel J. Gershman. Compositional inductive biases in function learning. *Cognitive Psychology*, 2017. ISSN 00100285. doi: 10.1016/j.cogpsych.2017.11.002.

[15] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services*, 1:1–10, 10 2017.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL `https://doi.org/10.1145/2939672.2939778`.

[17] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[18] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021. doi: 10.1109/TETCI.2021.3100641.

[19] B. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*, 2018.

[20] Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988. doi: 10.1016/0010-0277(88)90031-5.

[21] François Chollet. On the measure of intelligence, 2019. URL `https://arxiv.org/abs/1911.01547`.

[22] B. Lake, R. Salakhutdinov, and J. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332 – 1338, 2015.

[23] Shuchen Wu, Noémi Éltető, Ishita Dasgupta, and Eric Schulz. E pluribus unum but how? chunking as a rational solution to the speed-accuracy trade-off, Feb 2022. URL `psyarxiv.com/sjh27`.

[24] Claude Bonnet, Jordi Fauquet, and Santiago Ferrer. Reaction times as a measure of uncertainty. *Psicothema*, 20:43–8, 03 2008.

[25] Scott Jarvis and Aneta Pavlenko. Crosslinguistic influence in language and cognition. *Crosslinguistic Influence in Language and Cognition*, pages 1–287, 01 2007. doi: 10.4324/9780203935927.

[26] Johan Wagemans, Jacob Feldman, Sergei Gepshtein, Ruth Kimchi, James R Pomerantz, Peter A Van der Helm, and Cees Van Leeuwen. A century of gestalt psychology in visual perception: Ii. conceptual and theoretical foundations. *Psychological bulletin*, 138(6):1218, 2012.

[27] W. Metzger. *Laws of seeing*. MIT Press, 2006.

[28] Brian J Compton and Gordon D Logan. Evaluating a computational model of perceptual grouping by proximity. *Perception & Psychophysics*, 53(4):403–421, 1993.

[29] Dwight J Peterson and Marian E Berryhill. The gestalt principle of similarity benefits visual working memory. *Psychonomic bulletin & review*, 20(6):1282–1289, 2013.

[30] Don C Donderi. Visual complexity: a review. *Psychological bulletin*, 132(1):73, 2006.

[31] Raphaëlle Malassis, Arnaud Rey, and Joël Fagot. Non-adjacent dependencies processing in human and non-human primates. *Cognitive Science*, 42(5):1677–1699, 2018.

[32] Vicky Froyen, Jacob Feldman, and Manish Singh. Bayesian hierarchical grouping: Perceptual grouping as mixture estimation. *Psychological Review*, 122(4):575, 2015.

[33] Johannes Bill, Hrag Pailian, Samuel J. Gershman, and Jan Drugowitsch. Hierarchical structure is employed by humans during visual motion perception. *Proceedings of the National Academy of Sciences*, 117(39):24581–24589, 2020. ISSN 0027-8424. doi: 10.1073/pnas.2008961117. URL `https://www.pnas.org/content/117/39/24581`.

[34] Pedro Alves, Chris Foulon, Vyacheslav Karolis, Danilo Bzdok, Daniel Margulies, Emmanuelle Volle, and Michel Thiebaut de Schotten. An improved neuroanatomical model of the default-mode network reconciles previous neuroimaging and neuropathological findings. *Communications Biology*, 2:1–14, 10 2019. doi: 10.1038/s42003-019-0611-3.

[35] Alexandre Abraham, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gael Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8, 2014. ISSN 1662-5196. doi: 10.3389/fninf.2014.00014. URL `https://www.frontiersin.org/article/10.3389/fninf.2014.00014`.

[36] Hilary Richardson, Grace Lisandrelli, Alexa Riobueno-Naylor, and Rebecca Saxe. Development of the social brain from age three to twelve years. *Nat Commun*, 1027(9), 2018. doi: https://doi.org/10.1038/s41467-018-03399-2.

[37] Gael Varoquaux, Alexandre Gramfort, Fabian Pedregosa, Vincent Michel, and Bertrand Thirion. Multi-subject dictionary learning to segment an atlas of brain spontaneous activity. In *Proceedings of the 22nd International Conference on Information Processing in Medical Imaging*, IPMI'11, page 562–573, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 9783642220913.

[38] William W. Seeley. The salience network: A neural system for perceiving and responding to homeostatic demands. *Journal of Neuroscience*, 39(50):9878–9882, 2019. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.1138-17.2019. URL https://www.jneurosci.org/content/39/50/9878.

[39] Critchley HD. Neural mechanisms of autonomic, affective, and cognitive integration. *J Comp Neurol.*, 493(1), 2005. doi: 10.1002/cne.20749.

[40] Nick Medford and Hugo Critchley. Conjoint activity of anterior insular and anterior cingulate cortex: Awareness and response. *Brain structure & function*, 214:535–49, 06 2010. doi: 10.1007/s00429-010-0265-x.

[41] Bart Aben, Cristian Buc Calderon, Eva Van den Bussche, and Tom Verguts. Cognitive effort modulates connectivity between dorsal anterior cingulate cortex and task-relevant cortical areas. *Journal of Neuroscience*, 40(19):3838–3848, 2020. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.2948-19.2020. URL https://www.jneurosci.org/content/40/19/3838.

[42] Kathleen J Burman, David H Reser, Hsin-Hao Yu, and Marcello G P Rosa. Cortical input to the frontal pole of the marmoset monkey. *Cerebral cortex (New York, N.Y. : 1991)*, 8(21), 2011. doi: https://doi.org/10.1093/cercor/bhq239.

[43] Huaigui Liu, Wen Qin, Wei Li, Lingzhong Fan, Jiaojian Wang, Tianzi Jiang, and Chunshui Yu. Connectivity-based parcellation of the human frontal pole with diffusion tensor imaging. *The Journal of neuroscience*, 16(33), 2013. doi: https://doi.org/10.1523/JNEUROSCI.4882-12.2013.

[44] Michael Petrides and Deepak N. Pandya. Efferent association pathways from the rostral prefrontal cortex in the macaque monkey. *The Journal of neuroscience*, 27(43), 2007. doi: https://doi.org/10.1523/JNEUROSCI.2419-07.2007.

[45] Keri S. Taylor, David A. Seminowicz, and Karen D. Davis. Two systems of resting state connectivity between the insula and cingulate cortex. *Human Brain Mapping*, 30(9):2731–2745, 2009. doi: https://doi.org/10.1002/hbm.20705. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.20705.

[46] Francis Stevens, R.A. Hurley, and Katherine Taber. Anterior cingulate cortex: Unique role in cognition and emotion. *Journal of Neuropsychiatry and Clinical Neurosciences*, 23:121–125, 01 2011. doi: 10.1176/jnp.23.2.jnp121.

[47] Jue Wang, Ning Yang, Wei Liao, Han Zhang, Chao-Gan Yan, Yu-Feng Zang, and Xi-Nian Zuo. Dorsal anterior cingulate cortex in typically developing children: Laterality analysis. *Developmental Cognitive Neuroscience*, 15:117–129, 2015. ISSN 1878-9293. doi: https://doi.org/10.1016/j.dcn.2015.10.002. URL https://www.sciencedirect.com/science/article/pii/S1878929315000924.

[48] Sarah Vinette and Signe Bray. Variation in functional connectivity along anterior-to-posterior intraparietal sulcus, and relationship with age across late childhood and adolescence. *Developmental Cognitive Neuroscience*, 31, 04 2015. doi: 10.1016/j.dcn.2015.04.004.

[49] Marisa Loitfelder, Stephan CJ Huijbregts, Ilya Milos Veer, Hanna S Swaab, Mark A Van Buchem, Reinhold Schmidt, and Serge A Rombouts. Functional connectivity changes and executive and social problems in neurofibromatosis type i. *Brain connectivity*, 5(5):312–320, 2015.

[50] A Craig. How do you feel—now? the anterior insula and human awareness. *Nature reviews. Neuroscience*, 10:59–70, 02 2009. doi: 10.1038/nrn2555.

[51] Lucina Uddin, Jason Nomi, Benjamin Hébert-Seropian, Jimmy Ghaziri, and Olivier Boucher. Structure and function of the human insula. *Journal of Clinical Neurophysiology*, 34:300–306, 07 2017. doi: 10.1097/WNP.0000000000000377.

[52] Francesca De Luca, Manuel Petrucci, Bianca Monachesi, Michal Lavidor, and Anna Pecchinenda. Asymmetric contributions of the fronto-parietal network to emotional conflict in the word–face interference task. *Symmetry*, 12(10):1701, 2020.

[53] Richard P. Bagozzi, Willem J. M. I. Verbeke, Roeland C. Dietvorst, Frank D. Belschak, Wouter E. van den Berg, and Wim J. R. Rietdijk. Theory of mind and empathic explanations of machiavellianism: A neuroscience perspective. *Journal of Management*, 39(7):1760–1798, 2013. doi: 10.1177/0149206312471393. URL https://doi.org/10.1177/0149206312471393.

[54] Pierre Perruchet and Annie Vinter. Parser: A model for word segmentation. *Journal of Memory and Language*, 39(2):246 – 263, 1998. ISSN 0749-596X. doi: https://doi.org/10.1006/jmla.1998.2576. URL http://www.sciencedirect.com/science/article/pii/S0749596X98925761.

[55] Emile Servan-Schreiber and John Anderson. Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16:592–608, 07 1990. doi: 10.1037/0278-7393.16.4.592.

[56] Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B Tenenbaum. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. *arXiv preprint arXiv:2006.08381*, 2020.

[57] Kewei Tu, Maria Pavlovskaia, and Song-Chun Zhu. Unsupervised structure learning of stochastic and-or grammars. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/24681928425f5a9133504de568f5f6df-Paper.pdf.

[58] Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2009.11.011. URL https://www.sciencedirect.com/science/article/pii/S0004370209001416. Special Review Issue.

[59] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

[60] Katrin Ortmann. Chunking historical german. In *NODALIDA*, 2021.

[61] Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. Neural models for sequence chunking. In *AAAI*, 2017.

[62] Sun Si, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Jie Bao. Joint keyphrase chunking and salience ranking with bert, 04 2020.

[63] Philip Gage. A new algorithm for data compression. http://www.pennelynn.com/Documents/CUJ/HTML/94HTML/19940045.HTM. Accessed: 2022-07-29.

[64] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. 09 2016.

[65] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010. doi: 10.1109/TPAMI.2010.46.

[66] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, aug 2000. ISSN 0162-8828. doi: 10.1109/34.868688. URL `https://doi.org/10.1109/34.868688`.

[67] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[68] Alison Gopnik. The scientist as child. *Philosophy of Science*, 63(4):485–514, 1996. doi: 10.1086/289970.

[69] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] As this work introduces an algorithmic approach, it is not clear to us with the potential negative societal impact it can induce.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] A proof sketch is described in the main paper, while the full proof can be found in the supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] They will be included in the supplementary material and publicly available.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Details on the experiments will be included in the supplementary.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] Results obtained run without special computing resources.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] From our awareness, no personally identifiable information is present in any of the assets.
5. If you used crowdsourcing or conducted research with human participants...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We used existing data from experiments conducted in previous work. This information can be found in the original articles.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] This question is not applicable to this project.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Since no human experiment is conducted for this project and we used experimental data from other published and publicly available work, this question is not applicable to us here and can be found in the original articles.

# Supplementary Information: Learning Structure from the Ground up—Hierarchical Representation Learning by Chunking

**Shuchen Wu**[*]
Computational Principles of Intelligence Lab
Max Planck Institute for Biological Cybernetics
Tübingen, Germany
shuchen.wu@tuebingen.mpg.de

**Noémi Éltető**
Department of Computational Neuroscience
Max Planck Institute for Biological Cybernetics
Tübingen, Germany
noemi.elteto@tuebingen.mpg.de

**Ishita Dasgupta**
Computational Cognitive Science Lab
Department of Psychology
Princeton University
dasgupta.ishita@gmail.com

**Eric Schulz**
Computational Principles of Intelligence Lab
Max Planck Institute for Biological Cybernetics
Tübingen, Germany
eric.schulz@tuebingen.mpg.de

## A Definitions

An observational sequence is made up of discrete, integer valued, size-one elementary observational unit coming from an atomic alphabet set $\mathbb{A}_0$, where $0$ represents the empty observational unit.

One example of such an observational sequence $S$ is:

$$010021002112000...$$

The atomic alphabet set is $\mathbb{A}_0 = \{0, 1, 2\}$. The elementary observation units are '0', '1', and '2'.

**Definition 1 (*Chunk*)**

*A chunk is made from any combinations of non-empty observational units* $\mathbb{A}_0 \setminus \{0\}$.

Examples of chunks from the observational sequence can be '1', '21', '211', '12', '2112', ... etc. $0$ represents an empty observation in the sequence.

**Definition 2 (*Belief Set*)**

*A belief set is the set of chunks that HCM uses to parse sequences, denoted as* $\mathbb{B}$.

An example belief set that HCM has learned to parse sequence $S$ can be: $\mathbb{B} = \{0, 1, 21, 211, 12, 2112\}$.

[*]

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

**Definition 3 (*Parsing*)**

*Chunks are being parsed from the beginning of the sequence. At each parsing step, the biggest chunk in the belief set that matches the upcoming sequence is chosen to explain the observation. The end of the previous parse initiates the next parse.*

Using the belief set $\{0, 1, 21, 211, 12, 2112\}$ to parse the sequence $S$ results in the following partition. $\underline{0}\ \underline{1}\ \underline{0}\ \underline{0}\ \underline{21}\ \underline{0}\ \underline{0}\ \underline{2112}\ \underline{0}\ \underline{0}\ \underline{0}$.

**Definition 4 (*Completeness*)**

*We say that a belief set is* complete *if at any point when the model parses the sequence, the upcoming observations can be explained by at least one chunk in the belief set.*

In this work, we only refer to complete belief sets.

**Definition 5 (*Parsing Length $N_{\mathbb{B}}$*)**

*A parsing length $N_{\mathbb{B}}$ of a sequence is the length of resulting sequence after being parsed by chunks in $\mathbb{B}$.*

**Definition 6 ($N_{\mathbb{B}}(c)$)**

*$N_{\mathbb{B}}(c)$ denotes number of times chunk $c$ in the belief set $\mathbb{B}$ appears in the parsed sequence.*

$N_{\mathbb{B}}(c)$ for all of the chunks $c$ on a belief set $\mathbb{B}$ sums to the parsing length: $N_{\mathbb{B}} = \sum_{c \in \mathbb{B}} N_{\mathbb{B}}(c)$

**Definition 7 ($N_{\mathbb{B}}(x \to y)$)**

*The number of times chunk $x$ is being parsed following chunk $y$. $x$ and $y$ are both chunks in the belief set $\mathbb{B}$.*

For any chunk $x$ within any belief set $\mathbb{B}$, $N_{\mathbb{B}}(x)$ has the following relation with $N_{\mathbb{B}}(x \to y)$:

$$N_{\mathbb{B}}(x) = \sum_{y \in \mathbb{B}} N_{\mathbb{B}}(x \to y) \tag{1}$$

When the length of the sequence becomes infinite, it is easier to work with probabilities instead of counting the number of chunk occurrences.

**Definition 8 (*Probability space of a belief set*)**

*With a belief set $\mathbb{B}$, one can define a associated probability space $(\mathcal{S}_{\mathbb{B}}, \mathcal{F}_{\mathbb{B}}, \mathcal{P}_{\mathbb{B}})$. $\mathcal{S}_{\mathbb{B}}$ is the sample space representing all of the possible outcomes of a chunk parse. An event space $\mathcal{F}$ is the space for all possible sets of events. $\mathcal{F}$ contains all the subsets of $\mathcal{S}_{\mathbb{B}}$. Additionally, the probability function $P_{A_{\mathbb{B}}} : \mathcal{F}_{\mathbb{B}} \to \mathbb{R}$ is defined on the event space $\mathcal{S}_{\mathbb{B}}$. The probability function $P_{A_{\mathbb{B}}}$ satisfies the basic axioms of probability:*

- *$P_{A_{\mathbb{B}}}(E) \geq 0\ \forall E \in \mathcal{F}$. For any subset in the event space, the probability of an observation being in the subset is positive.*
- *$M, N \in \mathcal{F}$, and $M \cap N = \mathbb{E}$, then $P(M \cup N) = P(M) + P(N)$. For two non-intersecting subsets in the event space, the probability of observing any element that falls within the union of the two subsets is the sum of the probability of observing any event within one subset and the probability of observing any event from the other subset.*
- *$P(S) = 1$. The probability of observing any event that belongs to the sample space is one.*

In the limiting case when the sequence becomes infinitely long, we formulate the probability of parsing chunk $c \in \mathbb{B}$.

$$P_{\mathbb{B}}(c) = \lim_{N_{\mathbb{B}} \to \infty} \frac{N_{\mathbb{B}}(c)}{N_{\mathbb{B}}} \tag{2}$$

A learning model keeps track of the occurrence probability associated with each chunk in the belief set. For a current belief set, the model assumes that the chunks within the belief set occurs independently.

The probability of observing a sequence of chunks $c_1, c_2, ....c_N$ can be denoted as $P(c_1, c_2, ....c_N)$. The joint probability of observing any chunk in the generative process is:

$$P(c_1, c_2, ....c_N) = \prod_{c_i \in \mathbb{B}_d} P_{\mathbb{B}_d}(c_i) \tag{3}$$

Chunks as observation units serve as independent factors that disentangle observations in the sequence.

**Definition 9 (*Marginal Parsing Frequency $M_d$*)**

*A vector that stores the number of parses for each chunk $c$ in the belief set $\mathbb{B}_d$.*

$\boldsymbol{M}_d$ is a vector with size $|\mathbb{B}_d|$.

**Definition 10 (*Transition Frequency $T_d$*)**

*The set of transition frequency from any chunk $c_i \in \mathbb{B}_d$ to $c_j \in \mathbb{B}_d$*

**Definition 11 (*Chunk Hierarchy Graph $\mathcal{G}_d$*)**

*The relation between chunks and their constructive components in the generative model is described by a chunk hierarchy graph $\mathcal{G}_d$ with vertex set $V_{\mathbb{A}_d}$ and edge set $E_{\mathbb{A}_d}$. In this hierarchical generative model, $d$ is the depth of the graph and $\mathbb{A}_d$ is the set of chunks used as atomic units to construct the sequence. Each vertex in $V_{\mathbb{A}_d}$ is a chunk, and edges connect the parent chunk vertices to their child chunk vertices.*

## B Independence Test as a Chunking Criterion

Combining any two chunks $c_L$ and $c_R$ in the current belief set by ranking their joint occurrence probability may result in combining independently occurring chunks together. To distinguish this scenario of taking precedence over correlated and yet lower probability chunk pairs, we use Pearson's chi-square statistic for evaluating statistical independence to assess if the consecutive parses of $c_l$ and $c_r$ observed in $\boldsymbol{T}$ are independent. We use a $\chi^2$-test of independence to assess the correlation of consecutive occurrences of $c_L$ followed by $c_R$ in $\mathbb{B}$. Let $c_L$ be an indicator variable that is 1 when chunk $c_l$ is parsed and 0 otherwise, similarly we formulate $c_R$ as another indicator variable of parsing the chunk $c_r$. Observations of $c_L$ and $c_R$ in parses are categorical variables and can be represented as rows and columns of a contingency table. The number of observations that $c_L = 1$ or any other observations ($c_L = 0$) consists of the row entries, indicating observations of $c_L$, while the number of observations $c_R = 1$ and $c_R = 0$ make up the column entries. The table, therefore, consists of two rows and two columns.

The null hypothesis is the statistical independence of consecutive observations. Given the independence hypothesis, the expected frequency for observing $c_l$ followed by $c_r$ is $E[c_L, c_R] = Np(c_L)p(c_R)$, with $N$ being the total number of parses.

$$\begin{aligned} \chi^2 &= \sum_{c_L=\{0,1\}} \sum_{c_R=\{0,1\}} \frac{(O(c_L, c_R) - E[c_L, c_R])^2}{E[c_L, c_R]} \\ &= \sum_{c_L=\{0,1\}} \sum_{c_R=\{0,1\}} \frac{N(p(c_L, c_R) - p(c_L)p(c_R))^2}{p(c_L)p(c_R)} \end{aligned} \tag{4}$$

The degree of freedom for this test is 1. A $\chi^2$ value of less than or equal to 0.05 is used as a criterion for rejecting the null hypothesis of independence.

### B.1 Independence Test as a Halting Criterion

In the rational version of the chunking algorithm, the independence test is also employed to evaluate the strength of statistical correlation between chunks in the current belief set as a criterion to continue

or to halt the chunking process. In this case, the contingency table contains rows and columns corresponding to all possible chunks in the current belief set, and the $\chi^2$-statistic is calculated as:

$$\chi^2 = \sum_{c_L \in \mathbb{B}} \sum_{c_R \in \mathbb{B}} \frac{(O(c_L, c_R) - E[c_L, c_R])^2}{E[c_L, c_R]} = N \sum_{c_L \in \mathbb{B}} \sum_{c_R \in \mathbb{B}} \frac{(p(c_L, c_R) - p(c_L)p(c_R))^2}{p(c_L)p(c_R)} \tag{5}$$

The degrees of freedom are $(|\mathbb{B}| - 1)^2$, and a $p$-value of $0.05$ is used as a criterion to reject the null hypothesis and thereby used as an evidence to continue the chunking process.

We chose 0.05 as a decision criterion for rejecting the null hypothesis of two consecutively occurring chunks to be independent to be consistent with standard conventions in statistics. However, in applications, the strictness of parameters can be adapted to task domains. For example, for medical data, you might want to have only a few chunks and should, therefore, set alpha to be conservatively low. However, when using chunks for predictions in downstream tasks, you might want to have more of them, and should, therefore, set alpha to be liberally high.

## C Rational Chunking Algorithm

**Algorithm 1:** Rational Chunking Algorithm

**input :** Seq, maxIter
**output :** $\mathbb{B}_d$, $\hat{\mathcal{G}}_d$, $\boldsymbol{T}_d$, $\boldsymbol{M}_d$
$d \leftarrow 0$ $iter \leftarrow 0$;
$\mathbb{B}_d$, $\boldsymbol{M}_d$, $\boldsymbol{T}_d$ = getSingleElementSets(Seq); `/* minimally complete atomic set */`
**while** *!Test($\boldsymbol{M}_d$, $\boldsymbol{T}_d$) and iter* $\leq$ *maxiter* **do**
  $\boldsymbol{M}_d$, $\boldsymbol{T}_d$ = Parse(Seq, $\mathbb{B}_d$);
  $c_L, c_R \leftarrow None$;
  $MaxChunk, MaxChunkP \leftarrow None$;
  $PreCk = \{\}$;
  **for** $(c_i, c_j) \in \mathbb{B}_d \backslash \{0\} \times \mathbb{B}_d \backslash \{0\}$ **do**
    $P_d(c_i \oplus c_j) = CalculateJoint(\boldsymbol{M}_d, \boldsymbol{T}_d, c_i, c_j)$;
    $P_{d+1}(c_i \oplus c_j) = \frac{P_d(c_i \oplus c_j)}{1 - P_d(c_i \oplus c_j)}$;
    **if** $P_{d+1}(c_i \oplus c_j) \geq$ *MaxChunkP and* $c_i \oplus c_j \notin PreCk$ *and !Test($c_i$, $c_j$)* **then**
      $c_L \leftarrow c_i$, $c_R \leftarrow c_j$;
      $MaxChunkP \leftarrow P_{d+1}(c_i \oplus c_j)$;
      $MaxChunk \leftarrow c_i \oplus c_j$
    **end**
  **end**
  $c \leftarrow c_L \oplus c_R$;
  $\mathbb{B}_{d+1} \leftarrow \mathbb{B}_d \cup c$;
  $\hat{\mathcal{G}}_{d+1} \leftarrow$AugmentGraph$(\hat{\mathcal{G}}_d, (c_L, c), (c_R, c))$;
  $PreCk.add(c)$;
**end**

## D Online HCM and Generalization to Visual-Temporal Sequences

Online HCM learns a chunk hierarchy graph $\hat{\mathcal{G}}$ from visual-temporal sequences. The chunk hierarchy graph $\hat{\mathcal{G}}$ can be initialized as an empty graph or a pre-trained chunk hierarchy graph. $\boldsymbol{M}$ retains the frequency of each chunk in the belief set $\mathbb{B}$ and $\boldsymbol{T}$ retains the transition frequencies of visual-temporally adjacent chunks sorted by temporal lags. Temporal lag is the time difference between the end of the previous chunk and start of the next chunk. The pseudocode for the Visual-Temporal HCM is shown in Algorithm 2.

At each parsing step, online HCM does the following:

1. Identifies the chunks biggest in volume that explain observation from the time point when the last chunk ended to the current time point, and store them in the set of current chunks.

2. Identifies the currently ending chunks and their adjacent previous chunks and updates their marginal and transition counts.
3. Modifies the set of chunks used to parse the sequence based on their adjacency.
4. Entries in $\boldsymbol{M}$ and $\boldsymbol{T}$ are subject to memory decay at the rate of $\theta$. If any entry goes below the deletion threshold $DT$, their corresponding entries in $\boldsymbol{M}$, $\boldsymbol{T}$, $\mathbb{B}$ and $\hat{\mathcal{G}}$ are deleted.

If two parsed visual-temporally adjacent chunks violates the independence testing criterion and they are within the proximity of each other under a padding threshold, they are grouped together into a new chunk. The constituent parts of a chunk remains in the belief set, with the count frequency subtracted by the estimation of the joint occurrence frequency.

**Algorithm 2:** Online HCM

**input :** Seq, $\hat{\mathcal{G}}$, $\theta$, $DT$
**output :** $\hat{\mathcal{G}}$
$\boldsymbol{M}, \boldsymbol{T} \leftarrow \hat{\mathcal{G}}.M, \hat{\mathcal{G}}.T$;
PreviousChunkBoundaryRecord $\leftarrow []$; `/* Record Chunk Endings */`
ChunkTerminationTime.setall(-1);
**while** *Sequence not over* **do**
  CurrentChunks, ChunkTerminationTime =
   IdentifyTheLatestChunks(ChunkTerminationTime);
  ObservationToExplain $\leftarrow$ refactor(Seq, ChunkTerminationTime);
  **for** *Chunk in CurrentChunks* **do**
    **for** *CandidateAdjacentChunk in PreviousChunkBoundaryRecord* **do**
      **if** *CheckAdjacency(Chunk, CandidateAdjacentChunk)* **then**
        $\boldsymbol{M}, \boldsymbol{T}, \mathbb{B}, \hat{\mathcal{G}} \leftarrow$ LearnChunking(Chunk, CandidateAdjacentChunk. $\boldsymbol{M}, \boldsymbol{T}, \mathbb{B}, \hat{\mathcal{G}}$);
      **end**
    **end**
    ChunkTerminationTime.update(CurrentChunks)
  **end**
  PreviousChunkBoundaryRecord.add(CurrentChunks);
  Forgetting($\boldsymbol{M}, \boldsymbol{T}, \mathbb{B}, \hat{\mathcal{G}}$, $\theta$, $DT$, PreviousChunkBoundaryRecord);
**end**

To process and update chunks online, HCM iterates through the visual temporal sequence, identifies chunks, marks the termination time corresponding to each visual dimension and stores them in ChunkTerminationTime. As multiple visual temporal chunks can be identified to occur simultaneously, CurrentChunks stores the identified chunks that have not reached their ending points.

Once one or more chunks are identified to be ending at a time point, they are stored inside PreviousChunkBoundaryRecord and their finishing time is updated for each visual pixel in ChunkTerminationTime. Corresponding entries in $\boldsymbol{M}$ are updated. The chunks that finishes after the start of the current chunk is checked with each current chunk on whether there is a visual temporal adjacency in addition to a violation of the independence test.

If a pair of adjacent chunks $c_L$ and $c_R$ violate the independence testing criterion, they are combined into one chunk $c_L \oplus c_R$. A new entry is created in $\boldsymbol{M}$ with the joint occurrence frequency for $c_L \oplus c_R$, this occurrence frequency is subtracted from the marginal record of $c_L$ and $c_R$. Additionally, other combinations that result in the same chunk will accumulate toward the count of $c_L \oplus c_R$. The constituents' transition entries are set to 0. As a new chunk, $c_L \oplus c_R$ inherits the adjacency entries of $c_R$, and the marginal frequencies for $c_L$ and $c_R$ are each subtracted by 1.

## E Proof of Recoverability

As the belief set $\mathbb{B}$ keeps changing when one modifies the chunks in a sequence, so does the parsing length $N_{\mathbb{B}}$ and the probability associated with the belief set $P_{A_{\mathbb{B}}}$. This translates to a change of $N$ and a set of constraints on the probabilities defined on the augmented support set. We approach this problem in the following steps:

- Formulate the definition of probabilities based on $N$.
- Identify all relevant changes of $N$ before and after the chunk update.
- Translate this change of $N$ to the constraints on probability updates.

We derive the relation between the probabilities when two chunks $c_L$ and $c_R \in \mathbb{A}_d$ are concatenated together to form a new chunk $c_L \oplus c_R$ and update the alphabet to $\mathbb{A}_{d+1}$.

#### E.0.1 Summary N

Going from $\mathbb{A}_d$ to $\mathbb{A}_{d+1}$, $c_L$ and $c_R$ are both chunks in $\mathbb{A}_d$ and merged together as a new chunk to augment $\mathbb{A}_d$. The chunks in $\mathbb{A}_d$ can be divided into three groups, $c_L$, $c_R$, and $\mathbb{A}_d \setminus \{c_L, c_R\}$. The relation between $N_d$ and $N_{d+1}$ is:

$$N_{d+1} = \left[\sum_{c \in \mathbb{A}d - c_L - c_R} N_d(c)\right] + N_{d+1}(c_L) + N_{d+1}(c_R) + N_{d+1}(c_L \oplus c_R) \tag{6}$$

Additionally, $N_{d+1}(c_L) = N_d(c_L) - N_d(c_L \to c_R)$, $N_{d+1}(c_R) = N_d(c_R) - N_d(c_L \to c_R)$. Chunking reduces the number of times sub-chunks are being parsed when sub-chunks occur right after each other by twofold.

$$N_d = \sum_{c \in \mathbb{A}_d} N_d(c) = \left[\sum_{c \in \mathbb{A}_d - c_L - c_R} N_d(c)\right] + N_d(c_L) + N_d(c_R) \tag{7}$$

Comparing the above two equations we arrive at

$$N_d(c_L) + N_d(c_R) = N_{d+1}(c_L) + N_{d+1}(c_R) + 2N_{d+1}(c_L \oplus c_R)$$

We know that $N_d(c_L \to c_R) = N_{d+1}(c_L \oplus c_R)$ and therefore: $N_{d+1}(c_L) + N_{d+1}(c_R) = N_d(c_L) + N_d(c_R) - 2N_d(c_L \to c_R)$. The relation between the parsing counts $N_d$ and $N_{d+1}$ when switching from the alphabet set $\mathbb{A}_d$ to $\mathbb{A}_{d+1}$ by chunking $c_L$ and $c_R$ in $\mathbb{A}_d$ together is:

$$N_{d+1} = \left[\sum_{c \in A_d - c_L - c_R} N_d(c)\right] + N_d(c_L) + N_d(c_R) - N_d(c_L \to c_R) \tag{8}$$

$$N_{d+1} = N_d - N_d(c_L \to c_R) \tag{9}$$

#### E.0.2 Marginal N

To proceed into formulating the joint probability given a particular belief space, we need to formulate how the count of $N(c)$ for a chunk changes when the belief space when switching from $\mathbb{A}_d$ to $\mathbb{A}_{d+1}$, with the same division as before.

Of course, the count function should be fixed. However, the probability function associated with the chunks will change based on the update of the belief set. We use the update of the count function to find the relation between the probability updates.

For all $x$ in $\mathbb{A}_d - \{c_L, c_R, c_L \oplus c_R\}$: $N_{d+1}(x) = N_d(x)$, $N_{d+1}(c_R) = N_d(c_R) - N_d(c_L \to c_R)$, $N_{d+1}(c_L) = N_d(c_L) - N_d(c_L \to c_R)$, and $N_{d+1}(c_L \oplus c_R) = N_d(c_L \to c_R)$.

### E.1 Probability Density Switch when $\mathbb{A}_d$ expands to $\mathbb{A}_{d+1}$

The constraint is: the number of counts $N$ for all chunks defined for the support set $\mathbb{A}_d$ must remain the same for the support set $\mathbb{A}_{d+1}$, so that the definition of $P_{\mathbb{A}_d}$ for all relevant chunks within $\mathbb{A}_d$ remains the same when $\mathbb{A}_d$ expands to $\mathbb{A}_{d+1}$.

The probability of a chunk occurring in the alphabet set $\mathbb{A}_d$ is defined as: $P_{\mathbb{A}_d}(c) = \lim_{N_d \to \infty} \frac{N_d(c)}{N_d}$.

Because $N_d$ and $N_{d+1}$ are only a constant away, both go to infinity if one of them does, so there is a relation between the definition of probability $P_{\mathbb{A}_d}(c)$ and $P_{\mathbb{A}_{d+1}}(c)$. For any chunk $x$ in $\mathbb{A}_d$ that is not $c_L$ and $c_R$, $N_{d+1}(x) = N_d(x)$, $P_{A_{d+1}}(x) = \lim_{N_{d+1} \to \infty} \frac{N_{d+1}(x)}{N_{d+1}} = \lim_{N_d \to \infty} \frac{N_d(x)}{N_d - N_d(c_L \to c_R)}$.

That is, the probability of a chunk of this category at d and d+1 satisfies this relationship that $P_{A_{d+1}}(x) = P_{A_d}(x)\frac{\lim_{N_d\to\infty} N_d}{\lim_{N_{d+1}\to\infty} N_d - N_d(c_L\to c_R)}$.

For $c_L$ and $c_R$ in $A_{d+1}$: $P_{A_{d+1}}(c_L) = \lim_{N_{d+1}\to\infty}\frac{N_{d+1}(c_L)}{N_{d+1}}$, $P_{A_d}(c_L) = \lim_{N_{d+1}\to\infty}\frac{N_d(c_L)}{N_d}$, $P_{A_{d+1}}(c_R) = \lim_{N_{d+1}\to\infty}\frac{N_{d+1}(c_R)}{N_{d+1}}$, $P_{A_d}(c_R) = \lim_{N_d\to\infty}\frac{N_d(c_R)}{N_d}$.

Since $N_{d+1}(c_L) = N_d(c_L) - N_d(c_L \oplus c_R)$, $P_{A_{d+1}}(c_L) = \lim_{N_{d+1}\to\infty}\frac{N_d(c_L)-N_d(c_L\oplus c_R)}{N_{d+1}}$, $P_{A_{d+1}}(c_L) = \lim_{N_d\to\infty}\frac{N_d(c_L)-N_d(c_L\to c_R)}{N_d-N_d(c_L\to c_R)}$, $P_{A_{d+1}}(c_R) = \lim_{N_d\to\infty}\frac{N_d(c_R)-N_d(c_L\to c_R)}{N_d-N_d(c_L\to c_R)}$

Finally, $P_{A_{d+1}}(c_L \oplus c_R) = \lim_{N_{d+1}\to\infty}\frac{N_{d+1}(c_L\oplus c_R)}{N_{d+1}}$, $P_{A_d}(c_L \oplus c_R) = \lim_{N_d\to\infty}\frac{N_d(c_L\to c_R)}{N_d}$.

Since $N_d(c_L \to c_R) = N_{d+1}(c_L \oplus c_R)$, we have $P_{A_{d+1}}(c_L \oplus c_R) = \lim_{N_d\to\infty}\frac{P_{A_d}(c_L\oplus c_R)N_d}{N_{d+1}}$.

For summary probabilities: $N_{d+1} = N_d - N_d(c_L \oplus c_R) = N_d - N_d P_d(c_L \oplus c_R)$, and $\frac{N_{d+1}}{N_d} = 1 - P_d(c_L \oplus c_R)$.

### E.1.1 Marginal Probabilities

The next level marginal probability follows the constraints when the support set changes from $A_d$ to $A_{d+1}$: $P_{d+1}(x) = \frac{P_d(x)}{1-P_d(c_L\oplus c_R)}$, $P_{d+1}(c_R) = \frac{P_d(c_R)-P_d(c_L\oplus c_R)}{1-P_d(c_L\oplus c_R)}$, $P_{d+1}(c_L) = \frac{P_d(c_L)-P_d(c_L\oplus c_R)}{1-P_d(c_L\oplus c_R)}$, $P_{d+1}(c_L \oplus c_R) = \frac{P_d(c_L\oplus c_R)}{1-P_d(c_L\oplus c_R)}$.

## E.2 Hierarchical Generative Model

At the beginning of the generative process, the atomic alphabet set $A_0$ is specified. Another parameter, $d$, specifies the number of additional chunks that are created in the process of generating the hierarchical chunks. Starting from the alphabet $A_0$ with initialized elementary chunks $c_i$ from the alphabet, the probability associated with each chunk $c_i$ in $A_0$ needs to satisfy the following criterion:

$$\sum_{c_i\in A_0} P_{A_0}(c_i) = 1 \tag{10}$$

Meanwhile, $P(c_i) \geq 0, \forall c_i \in \mathbb{A}_0$.

We assume that at each step the marginal and transitional probability of the previous steps are known. The next chunk is chosen as the combined chunks with the biggest probability. The order of construction in the generative model follows the rule that the combined chunk with the biggest probability on the support set of pre-existing chunk sets is chosen to be added to the set of chunks.

$$c_L \oplus c_R = \underset{c_L, c_R\in\mathbb{A}_d\setminus\{0\}}{\arg\max} P_{\mathbb{A}_d}(c_L \oplus c_R) \tag{11}$$

Under the constraint that:

$$P_{\mathbb{A}_d}(c_L)P_{\mathbb{A}_d}(c_R) \leq P_{\mathbb{A}_d}(c_L \oplus c_R) \leq \min\{P_{\mathbb{A}_d}(c_L), P_{\mathbb{A}_d}(c_R)\} \tag{12}$$

This can be calculated from the transitional and marginal probability of the previous step.

$$c_L \oplus c_R = \underset{c_L, c_R\in\mathbb{A}_d\setminus\{0\}}{\arg\max} P_{A_d}(c_L \oplus c_R) = \underset{c_L, c_R\in\mathbb{A}_d\setminus\{0\}}{\arg\max} P_{A_d}(c_L)P_{A_d}(c_R|c_L) \tag{13}$$

In practice, after the chunks are specified in $\mathbb{A}_d$, the probability value associated with chunks in $\mathbb{A}_0$ are sampled from a flat Dirichlet distribution, which is then sorted so that the smaller sized chunks contain more of the probability mass and the null-chunk carries the biggest probability mass. Then, the above constraint is checked for the assigned probability on each of the newly generated chunk with their associated alphabet set $\mathbb{A}_i$. This process repeats until the probability drawn satisfies the condition for every newly created chunk.

At first, the set of chunks are $\mathbb{A}_0$, which is assigned each as an integer. Then $d$ additional recombination processes are carried out. In each process, two chunks are randomly chosen from the pre-existent alphabet set to recombine into a new chunk, until $d$ additional chunks are being created to augment the set of chunks from $\mathbb{A}_0$ to $\mathbb{A}_d$. The Dirichlet distribution is randomly generated in an unsorted fashion, and then the biggest probability mass is assigned to 0. Constraints are checked recursively.

**Theorem 1** (Marginal Probability Space Conservation)**.** *After the addition of $c_{d,i} \oplus c_{d,j}$ and the change of probability, $P_{\mathcal{A}_d}$ is still a valid probability distribution.*

**Proof:**

$$\begin{aligned}\sum_{c_{d,k}\in\mathbb{A}_d} P_{\mathbb{A}_d}(c_{d,k}) = & \sum_{c_{d,k}\in\mathbb{A}_{d-1}-c_{d-1,i}-c_{d-1,j}} P_{\mathbb{A}_{d-1}}(c_{d-1,k})+ \\ & + P_{\mathbb{A}_{d-1}}(c_{d-1,i}) - P_{\mathbb{A}_{d-1}}(c_{d-1,j}|c_{d-1,i})P_{\mathbb{A}_{d-1}}(c_{d-1,i}) \\ & + P_{\mathbb{A}_{d-1}}(c_{d-1,j}) + P_{\mathbb{A}_{d-1}}(c_{d-1,j}|c_{d-1,i})P_{\mathbb{A}_{d-1}}(c_{d-1,i}) \\ & = 1\end{aligned} \tag{14}$$

□

**Theorem 2** (Measure Space Preservation)**.** *Given that at the end of the generative process with depth $d$ one ends up having an alphabet set $\mathbb{A}_d$, the probability space defined on $\mathbb{A}_i$, which includes the marginal and joint probability of any chunk and combinations of chunks in $\mathbb{A}_i$, $i = 0, 1, 2, \dots d$, which are predecessor alphabet sets of $\mathbb{A}_d$, all values in the set $\mathbb{M}_d$ and $\mathbb{T}_d$ remain the same no matter how the future support set changes according to the generative model.*

**Proof:** By induction.

- Base case: starting from the initialized alphabet set $\mathbb{A}_0$, the probability of $P_{A_0}(c), c \in \mathbb{A}_0$, and the probability of $P_{A_1}(xy), x, y \in \mathbb{A}_0$, for all valid c, x, y, when the alphabet is $\mathbb{A}_1$. Going from $\mathbb{A}_0$ to $\mathbb{A}_1$, $N_0(c)$, $N_0$ and $N_0(x \to y)$ does not change, therefore $P_{A_0}(c)$ and $P_{A_0}(x \to y)$ at the alphabet $\mathbb{A}_1$ is the same as that when the alphabet is $\mathbb{A}_0$.
- Induction Step: starting from the initialized alphabet set $\mathbb{A}_d$, the probability of $P_{A_d}(c), c \in \mathbb{A}_d$, and the probability of $P_{A_d}(xy), x, y \in \mathbb{A}_d$, for all valid c, x, y, when the alphabet is $\mathbb{A}_{d+1}$. Going from $\mathbb{A}_d$ to $\mathbb{A}_{d+1}$, $N_d(c)$, $N_d$ and $N_d(x \to y)$ does not change, therefore $P_{A_d}(c)$ and $P_{A_d}(x \to y)$ at the alphabet $\mathbb{A}_{d+1}$ is the same as that when the alphabet is $\mathbb{A}d$.

□

**Theorem 3.** *The order of $P_{A_i}(xy), x, y \in A_i$ for any $i = 0, 1, 2, \dots d$ at any previous belief space is preserved throughout the update.*

**Proof:** At the end of the generative process with depth $d$, one ends up having such an alphabet set: $\mathbb{A}_d$. The probability space defined on $\mathbb{A}_i$, which includes the marginal and joint probability of any chunk and combinations of chunks in $\mathbb{A}_i$, $i = 0, 1, 2, \dots d$ is preserved, hence the order is preserved.

□

The generative process can be described by a graph update path. The specification of the initial set of atomic chunks $\mathbb{A}_0$ corresponds to an initial graph $\mathcal{G}_0$ with the atomic chunks as its vertices. At the i-th iteration, as the generative graph goes from $\mathcal{G}_{\mathbb{A}_i}$ to graph $\mathcal{G}_{\mathbb{A}_{i+1}}$, two none zero chunks $c_L$, $c_R$ chosen from the pre-existent set of chunks $\mathbb{A}_i$ and are concatenated into a new chunk $c_L \oplus c_R$, augmenting $\mathbb{A}_i$ by one to $\mathbb{A}_{i+1}$. The vertex set also increments from $V_{\mathbb{A}_i}$ to $V_{\mathbb{A}_{i+1}} = V_{\mathbb{A}_i} \cup c_L \oplus c_R$. Moreover, two directed edges connecting the parental chunks to the newly-created chunk are added to the set of edges: $E_{\mathbb{A}_i}$ to $E_{\mathbb{A}_i} = E_{\mathbb{A}_i} \cup (c_L, c_L \oplus c_R) \cup (c_R, c_L \oplus c_R)$. The series of graphs created during the chunk construction process going from $\mathcal{G}_{\mathbb{A}_0}$ to the final graph $\mathcal{G}_{\mathbb{A}_d}$ with $d$ constructed chunks can be denoted as a graph generating path $P(\mathcal{G}_{\mathbb{A}_0}, \mathcal{G}_{\mathbb{A}_d}) = (\mathcal{G}_{\mathbb{A}_0}, \mathcal{G}_{\mathbb{A}_1}, \mathcal{G}_{\mathbb{A}_2}, ..., \mathcal{G}_{\mathbb{A}_d})$.

### E.3 Learning the Hierarchy

The rational chunking model is initialized with one minimally complete belief set, the learning algorithm ranks the joint probability of every possible new chunk concatenated by its pre-existing belief set, and picks the one with the maximal occurrence joint probability on the basis of the current set of chunks as the next new chunk to enlarge the belief set. With the one-step agglomerated belief set, the learning model parses the sequence again. This process repeats until the chunks in the belief set pass the independence testing criterion.

**Theorem 4** (Learning Guarantees on the Hierarchical Generative Model)**.** *As $N \to \infty$, the chunk construction graph learned by the model $\hat{\mathcal{G}}$ is the same as the chunk construction graph of the generative model: $\hat{\mathcal{G}} = \mathcal{G}$, which entails that they have the same vertex set: $\hat{\mathbf{V}} = \mathbf{V}_{\mathcal{G}}$ and the same*

*edge set:* $\hat{\mathbf{E}} = \mathbf{E}_{\mathcal{G}}$. *Additionally, the belief set learned by the chunk learning model* $\mathbb{B}_d = \mathbb{A}_d$, *and the marginal probability evaluated on the learned belief set* $\boldsymbol{M}_{\mathbb{B}_d}$ *associated with each chunk is the same as the marginal probability imposed by the generative model on the generative belief set* $\boldsymbol{M}_{\mathbb{A}_d}$.

**Proof:** Given that all of the empirical estimates are the same as the true probabilities defined by the generative model, we prove that starting with $\mathbb{B}_0$, the learning algorithm will learn $\mathbb{B}_D = \mathbb{A}_D$. $\mathbb{A}_D$ is the belief set imposed by the generative model. We approach this proof by induction.

**Base Step:** As the chunk learner acquires a minimal set of atomic chunks that can be used to explain the sequence at first, the set of elementary atomic chunks learned by the model is the same as the elementary alphabet imposed by the generative model, i.e. $\mathbb{B}_0 = \mathbb{A}_0$. Hence, the root of the graph, which contains the nodes without their parents, is the same, $\hat{\mathcal{G}} = \mathcal{G}$; put differently, $\hat{\mathbf{V}}_0 = \mathbf{V}_0$

Additionally, the learning model approximates the probability of a specific atomic chunk as $\hat{P}_{\mathbb{A}_0}(a_i)$. As $n \to \infty$, for all chunks $c$ in the set of atomic elementary chunks in $\mathbb{B}_0$, the empirical probability evaluated on the support set is the same as the true probability assigned in the generative model with the alphabet set $\mathbb{A}_0$:

$$\hat{P}_{\mathbb{B}_0}(c) = \lim_{n\to\infty} \frac{N_0(c)}{N_0} = P_{\mathbb{A}_0}(c) \tag{15}$$

**Induction hypothesis**: Assume that the learned belief set $\mathbb{B}_d$ at step $d$ contains the same chunks as the alphabet set $\mathbb{A}_d$ in the generative model.

The HCM, by keeping track of the transition probability between any pairs of chunks, calculates $\hat{P}_{\mathbb{B}_d}(c_i|c_j)$ for all $c_i, c_j$ in $\mathbb{B}_d$. Afterwards, it finds the pair of chunks $c_i, c_j$, such that the chunk created by combining $c_i$ and $c_j$ together contains the maximum joint probability violating the independence test as candidate chunks to be combined together.

$$\hat{P}_{\mathbb{B}_d}(c_i \oplus c_j) = \sup_{c_i, c_j \in \mathbb{B}_d} \hat{P}_{\mathbb{B}_d}(c_i)\hat{P}_{\mathbb{B}_d}(c_j|c_i) \tag{16}$$

We know that in the generative step the suprimum of the joint probability with the support set $\mathbb{A}_d$ is being picked to form the next chunk in the representation graph, so each step of the process at step $d$ satisfies the condition that:

$$P_{\mathbb{A}_d}(c_i \oplus c_j) = \sup_{c_i, c_j \in \mathbb{A}_d} P_{\mathbb{A}_d}(c_i)P_{\mathbb{A}_d}(c_j|c_i) \tag{17}$$

Since $P_{\mathbb{A}_d}(c_j|c_i) = \hat{P}_{\mathbb{B}_d}(c_j|c_i)$, $P_{\mathbb{A}_d}(c_i) = \hat{P}_{\mathbb{B}_d}(c_i)$, the chunks $c_i$ and $c_j$ chosen by the learning model will be the same ones as those created in the generative model.

**End step**: The chunk learning process stops once an independence test has been passed, which means that the sequence is better explained by the current set of chunks than any of the other possible next-step chunk combinations. This is the case once the chunk learning algorithm has learned a belief set $\mathbb{B}_d$ that is the same as the generative alphabet set $\mathbb{A}_d$. At this point $\hat{\mathcal{G}} = \mathcal{G}$ □

# F Experiment Detail

## F.1 Chunk Recovery and Convergence

To test the model's learning behavior on this type of sequential data, random graphs of chunk hierarchies with an associated occurrence probability for each chunk are specified by the hierarchical generative process. To do so, an initial set of specified atomic chunks $\mathbb{A}_0$ and a pre-specified level of depth (new chunks) $d$ is used to initiate the generation of a random hierarchical generative graph $\mathcal{G}$. In total, there are $|\mathbb{A}_0| + d$ number of chunks in the generative alphabet $\mathbb{A}$, with chunk $c$ having an occurrence probability of $P_{\mathbb{A}}(c)$ on the sample space $\mathbb{A}$. Once a hierarchical generative model is specified, it is then used to produce training sequences with varying length $N$ to test the chunk recovery.

To compare representation learned by the rational chunking model with the ground truth generative model $\mathcal{G}$, a discrete version of Kullback–Leibler divergence is used to evaluate learning performance:

$$KL(P||Q) = \sum_{c\in\mathbb{A}} P_{\mathbb{A}}(c) \log_2(\frac{P_{\mathbb{A}}(c)}{Q_{\mathbb{A}}(c)}) \tag{18}$$

$P_{\mathbb{A}}(c)$ is defined by the generative model. $Q(c)$ is the learned probability of chunk $c$. To evaluate $Q_{\mathbb{A}}(c)$, the learned representation is used to produce "imagined" sequences of length 1000. After that, the occurrence probability of each chunk $c$ in $\mathbb{A}(c)$ is used to evaluate $Q$, comparing the HCM's learned representation with the ground truth.

For comparison, we used the same sequence used for training HCM to train a 3-layer recurrent neural network (one embedding layer with 40 hidden units, one LSTM layer with drop-out rate = 0.2, and one fully connected layer, batch size = 5, sequence length = 3, epoch = 1, so that the data used for training is the same as N) of the training sequence, and used it to generate predictive sequences with of length 1000. The predicted sequence is parsed in the unit of the generative alphabet, producing a discrete distribution on the same support set of the generative model. This distribution is used to calculate KL. The rational chunking model is trained on the the sequence $S$ with increasing sizes (from 100 to 3000 with steps of 100) produced by the hierarchical generative graph. For each depth $d$, 5 random chunk hierarchies with the same depth is randomly assigned. The sequence generated by these random chunking graphs are then used to train both HCM and RNN.

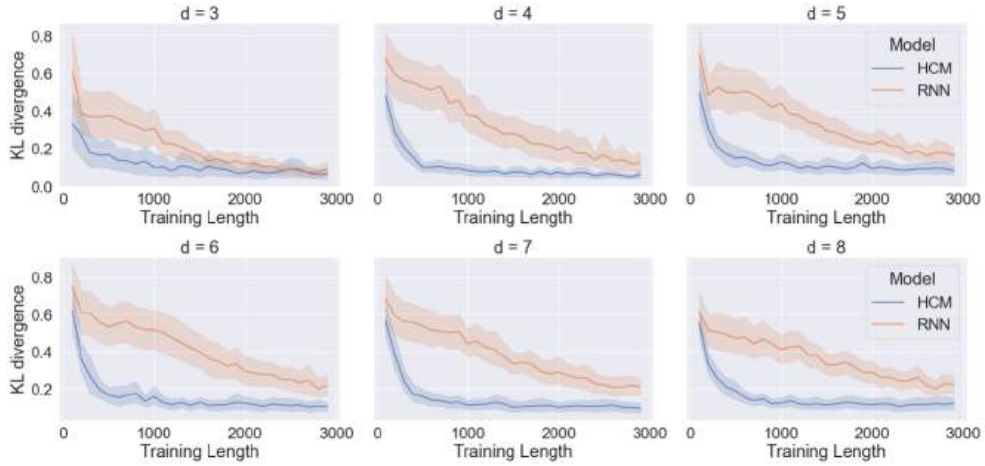


Figure 1: Learning Comparison Between HCM and RNN with Varying Graph Depth

### F.2 Transfer Between Environments with Overlapping and Interfering Structure

After training on a sequence, HCM acquires an interpretable representation. Knowing what the model has learned enables us to directly know what type of hierarchical environment would facilitate or interfere with the learned representations.

More formally, two HCM models might have acquired different hierarchical chunking graphs $\mathcal{G}_i$ and $\mathcal{G}_j$ from their past experience. These might lie on the graph construction path $(\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_d)$. The HCM with a chunk hierarchy graph 'closer' to the ground truth chunk $\mathcal{G}_d$ on the path, takes fewer iteration to arrive at $\mathcal{G}_d$. This also applies when the chunk hierarchies starting out are not along the graph construction path but only showing partial overlap. In other words, if $D(\mathcal{G}_i, \mathcal{G}_d) \leq D(\mathcal{G}_j, \mathcal{G}_d)$ then representation learned by HMC with graph structure $\mathcal{G}_i$ is more facilitative than that with graph structure $\mathcal{G}_j$.

Similarly, the chunk hierarchy $\mathcal{G}_i$ learned by an HCM might facilitate its performance in a new environment where $\mathcal{G}_i$ lies along the graph construction path to the true $\mathcal{G}_d$, i.e. there is partial overlap between the chunk hierarchies.

We took a graph with the trained representation and make it learn from sequences generated from a transfer and interfering environment. In the mean time, we used naive models separately to learn representations from the facilitative and interfering environment with an increasing sequence length from 50 to 1000. For all of the training models, the forgetting rate is set to $0.996$, with the

deletion threshold being $0.01$. An imaginative sequence with length 1000 is used to evaluate the discrepancies between the non-naive learner, and the naive learner, with respect to the corresponding facilitative/interfering environment.

Note that with the model that started out from a representation learned in an interfering environment, it learns representation from the new environment as shown in figure 2, albeit slower than the naive model.

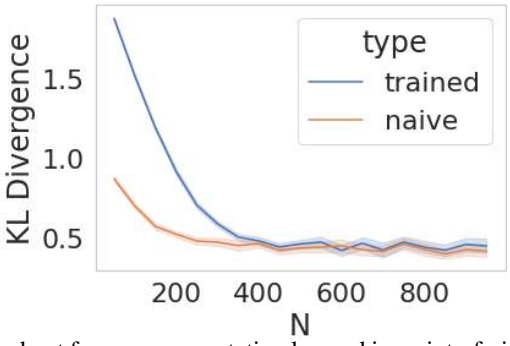


Figure 2: The model started out from a representation learned in an interfering environment converges in learning eventually, albeit slower than the naive model.

### F.3 Visual Hierarchical Chunks

The visual hierarchical chunks are crafted as binary arrays. The dark pixels are encoded as 1 and the background 0. Each image in the generative hierarchy is 25 dimensional (5 x 5) in the visual domain and size 1 in the temporal domain. An empty array is included to denote no observation. The alphabet $\mathbb{A}$ of the generative model include all 14 images in the generative hierarchy. In the generative hierarchy, higher level chunks are a composition of the lower level chunks. The occurrence probability for each generative visual chunk is drawn from a flat Dirichlet distribution with the empty observation retaining the highest mass, to emulate the sparsity of observation signals.

$$f(x_1, ..., x_k; \alpha_1, ..., \alpha_K) = \frac{1}{\mathbf{B}(\mathbf{a})} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{19}$$

Where the beta function when expressed using gamma function is: $\mathbf{B}(\mathbf{a}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_i^K \alpha_i)}$, and $\mathbf{a} = (\alpha_1, .., \alpha_K)$. The parameters $(\alpha_1, .., \alpha_K)$ with $K = |\mathbb{A}|$ are all set to one.

To generate the training sequence, $P_{\mathbb{A}}(c)$, $c_1, ..., c_K \in \mathbb{A}$ is assigned by the sampled distribution. Each image $c$ in the hierarchy are sampled independently with probability $P_{\mathbb{A}}(c)$and appended to the end of the sequence. As a result, there are visual correlations in the sequence defined by the hierarchy, but temporally, each image slice is sampled independently. In total, the sequence is made up of 2000 images.

We use online HCM to learn representation from visual-temporal sequences (forgetting rate = $0.996$, deletion threshold = $0.01$). HCM learn to construct representations from simple to complex (representation snapshots are collect at $t = 10$, $t = 100$, and $t = 1000$ respectively). Figure 3 shows the construction images from simple to complex. Parent chunks are made from the concatenation of their children chunks.

### F.4 GIF Movement

Here, we take a GIF file of a squid jumping in the sea with bubbles rising in the background. The entire animation is made up of 10 frames of $25 \times 25$ images. Each unique color of the GIF is mapped to an integer, with the background having a value of 0. In this way, the animation sequence

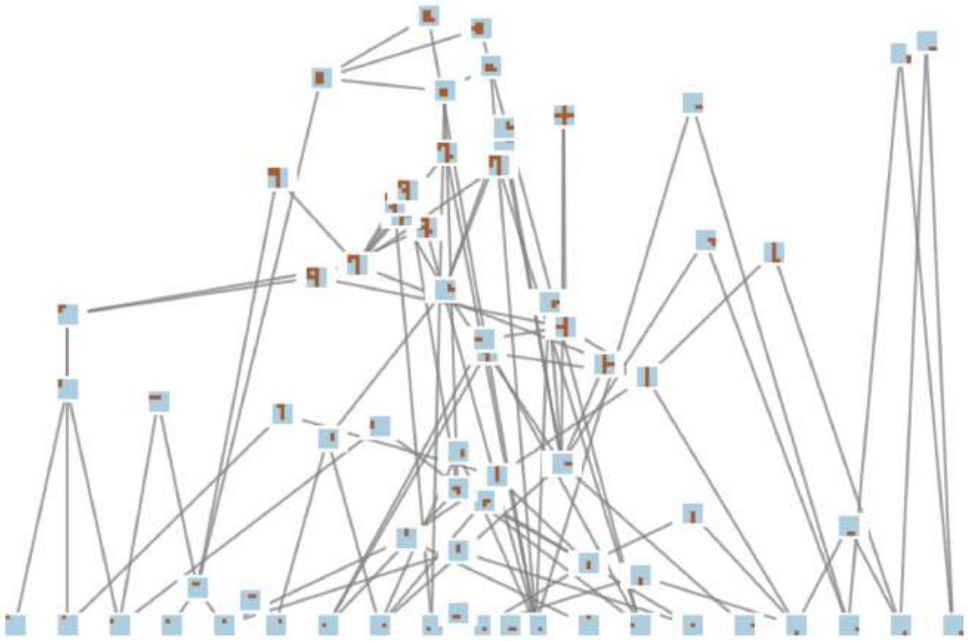

Figure 3: Construction of Hierarchy for Visual Chunks

becomes an integer array with size $T \times H \times W$. $T$ is the temporal dimension, $H$ and $W$ are the respective spatial dimension of the sequence. In this way, the gif file is converted into a tensor with size $10 \times 25 \times 25$. The entire movement is repeated 100 times and trained on the online version of HCM (forgetting rate = $0.996$, deletion threshold = $0.01$). Images are taking from the chunk learning graph of HCM at the end of the training process.

### F.5 Human Experiment

In the dataset from Wu et al. (2022), 47 participants are recruited from Amazon Mechanical Turk for a sequence learning experiment. Specifically, they conduct a serial-reaction-time task. In this task, participants are instructed to press the corresponding key on the keyboard upon observation of consecutive sequential instructions. Participants are rewarded based on a combination of speed and accuracy. Particularly, the training sequence is made up of sampling from the chunk [1,2,3] and [4] independently without pauses in between. Participants chunking behavior inferred from the reaction-time speed up is provided. If participants would be confident that some particular sequential instructions will show up, then they will be more confident to predict within-chunk items compare to between-chunk items and thereby speed up their reaction time.

The average chunk size across the training sequence is evaluated by averaging the size with a window of 30 chunks. Longer chunks imply that the predictive horizon, i.e. how confident participants can predict the upcoming sequential instructions, increases with practice.

To evaluate chunk learning of RNN on the same sequence, the probability estimate of each instruction choice in RNN is compared with the human data by evaluating the predicted negative log probability of the upcoming sequential instruction as a proxy of reaction time, since reaction time is often modeled as the negative log of choice probability. This reaction time is then grouped into within and between-chunk reaction time using mixtures of Gaussian classification method as in Wu et al. (2022).

We run online HCM on the same training sequence (forgetting rate = $0.90$, deletion threshold = $0.1$). We recorded the sequence of chunks in addition to the probability of chunk activation as calculated

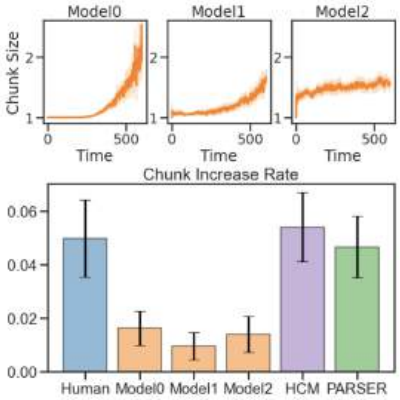

Figure 4: Comparison of chunk increase rate across three RNN models. Model0 is the architecture used in the main paper.

from the marginal frequencies. The average chunk size evaluated on the chunk sequence is used as a measure to compare with humans and RNNs. Additionally, the probability of within-chunk reaction time is set to $1 - 4\epsilon$ with $\epsilon = 0.05$ denoting the probability of choosing instructions outside of the predicted chunks.

### F.5.1 Size of RNN

We compared the chunk size increase rate across three RNN models varying in size as in Figure 4. Model0 is the RNN that we used in the experiment, which has the dimension of 40 embedding dimensions, and 3 layers, each with 40 LSTM units and a dropout rate of 0.2, followed by a fully connected feed-forward layer.

Model1 reduces the size of Model0 by half. Model1 has 20 embedding dimensions, and 3 layers, each with 20 LSTM units and the same dropout rate, followed by a fully connected feed-forward layer.

Model2 is about two times the size of Model0, with the same 40 embedding dimensions, and 5 layers LSTM neurons, each layer has 40 hidden units and a dropout rate of 0.2, followed by a fully connected feed-forward layer.

In short, Model1 is half of the size of Model0, and Model2 is double the size of Model0. Across all three RNN architectures, the chunk size only increased very slowly with increasing sequence length.

### F.6 fMRI dataset

The data comes from the brain development dataset (fMRI), including the measurement of 50 children (ages 3 - 13) and 33 young adults (ages 18 - 39). The experiment measures the resting state activities of subjects in the scanner, watching the PIXAR movie 'Partly Cloudy'. The data is down sampled to 4mm resolution, with a repetition time (TR) of 2 secs. Each session translates to 168 TRs in total. Signals of the fMRI BOLD activity are extracted using the MSDL labeled atlas of brain spontaneous activity Varoquaux et al. (2011) that segments regions of the brain and defines a functional parcellation of the brain's localized regions. In this way, the brain activities of each participant are extracted into a time series with 39 non-overlapping functional dimensions. Confounds from the original data file is extracted from the signal. The preprocessing pipeline, including the transformation from 4D images to 2D masked array, is obtained using the nilearn package offered by Abraham et al. (2014) heavily based on scikit-learn Pedregosa et al. (2011).

Upon exposure to a time series, online HCM (forgetting rate = 1.0, deletion threshold = 0.1) constructs chunks from their constituents and arrives at a nested hierarchy of chunk relations. The independent chunk activation frequencies $n$ within the hierarchy are identified upon another independent parse of the sequence.

In the movie, 19 scenes are tagged with their corresponding content. After running HCM on the data, one can obtain the chunk activation information after every tagged scene for each participant.

The 19 scenes are then categorized into 5 coarse categories: anger, pain, social, compassion, and sadness. The chunk activation probability is evaluated as the probability of one chunk being identified within 3 TRs after a tagged scene happens. Thereby, one can arrive at the chunk activation probability for each subject with the five tagged emotional categories.

### F.6.1 More examples

Figure 5 shows more examples of scene-tagged brain chunk activities.

An additional example of hierarchical structure learned by HCM that describes the nested hierarchical relationship between brain activation regions that are related to social and emotion recognition as well as theory-of-mind circuits is shown in Figure 6.

Figure 7 shows the conditional probability of scene contents given the activation of size-2 chunks. Brain functional regions such as 'D ACC' and 'R A Ins' show a multi-faceted prospect in reacting to scenes with a diverse range of categories, whereas chunks such as 'Striate' and 'R Par' activate only to the scene when it contained greetings and a hug.

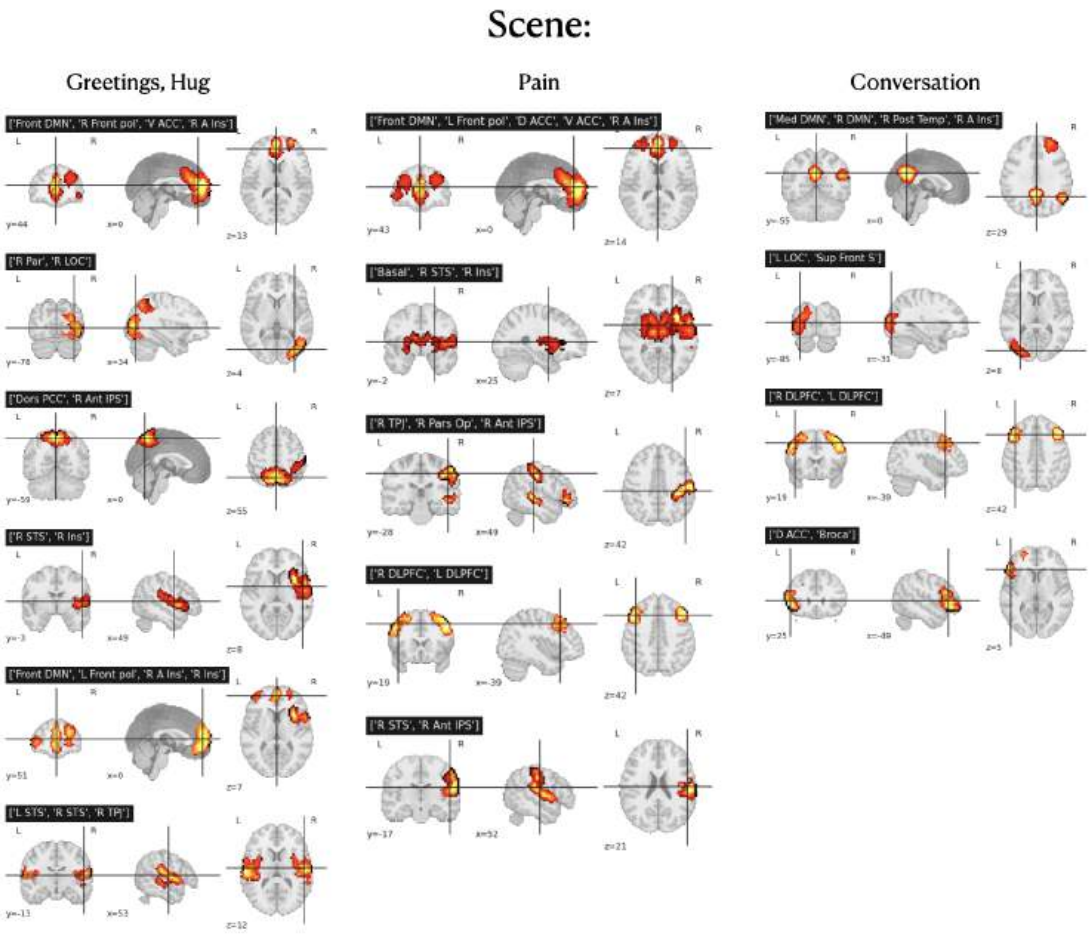


Figure 5: Additional Examples of Scene-Tagged Chunks of Brain Area Activation.

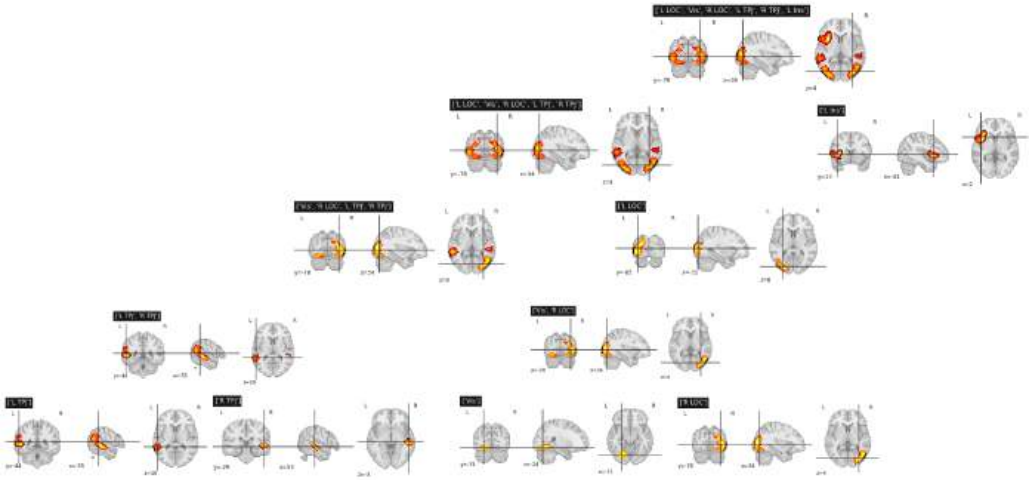

Figure 6: Additional example of hierarchical relationship between activation chunks of brain areas.

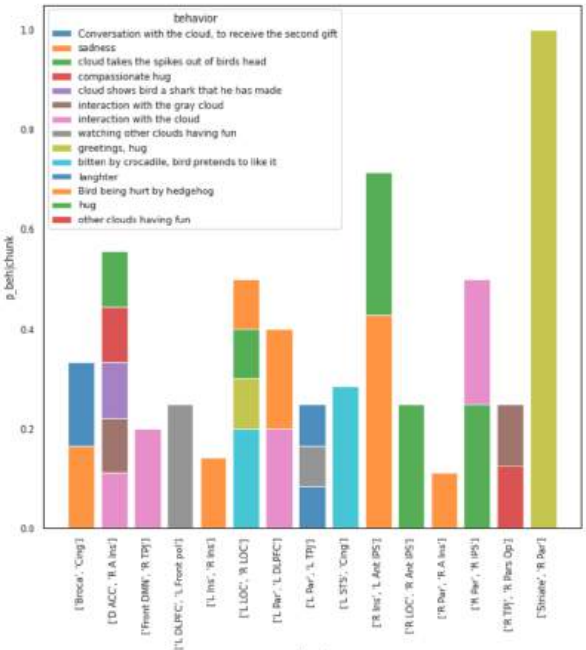


Figure 7: Conditional probability chunk activation given scene content.

## G Translating representation learned by HCM into $n$th order Markov Chain

Given an HCM that has learned a set of chunks $\mathbb{B}$. The storage of this representation demands the storage of $|\mathbb{B}|$ number of frequencies for each learned chunk entry, and the storage of the transition probability in a matrix with size $|\mathbb{B}| \times |\mathbb{B}|$.

Translating this HCM representation into an $n$th order Markov chain in the most parsimonious way would demand the storage of each individual state specific to each chunk. Thereby, a Markov chain

that contains the full information as in the chunks and transitions between chunks learned by HCM requests the number of states $n = \sum_{c \in \mathbb{B}} |c|$, where $|c|$ denotes the size of each chunk. So this Markov chain needs to store the probability of $\sum_{c \in \mathbb{B}} |c|$ number of states and a transition probability matrix of $\sum_{c \in \mathbb{B}} |c| \times \sum_{c \in \mathbb{B}} |c|$, which is a matrix much bigger than $|\mathbb{B}| \times |\mathbb{B}|$.

With regard to chunk size, for a learned graph with hierarchy depth $d$, the worst case scenario of the maximum chunk size will be $2^d$. The upper bound of the maximum chunk size grows exponentially with depth $d$. Therefore, the number of states for an equivalent Markov chain will be bounded by $|\mathbb{B}| \times 2^d$.

# H Memory Analysis

We analyze the memory demand to store each chunk in the case of optimal encoding, and provide examples of why chunking is beneficial for encoding by formulating an Expected Unit of Explanatory Power (EUEP) measure.

Given a set of chunks $\mathbb{B}$, each chunk $c \in \mathbb{B}$ with size $|c|$. The observational sequence $S$ is tiled by chunks. The minimal code length $I(c_i)$ assigned to each chunk $c_i \in \mathbb{B}$ to distinguish one from another is bounded by the optimal code length $-\log_2 P(c_i)$.

$$I(c_i) = -\log_2 P(c_i) \tag{20}$$

A chunk that occurs quite frequently contains a low information content, and correspond to a small code length, hence it is easier for the agent to encode this chunk.

The average length of sequence spent per bit of information to store a specific chunk $c_i$ is: $\frac{|c_i|}{I(c_i)}$.

The bigger this length per bit ratio is, the more efficient storage is optimized to encode a chunk $c_i$ with occurrence probability $P(c_i)$.

We denote the expectation of this unit length per code across all possible chunks as **Expected Unit Explanatory Power** (EUEP):

$$EUEP = \sum_{c_i \in \mathbb{B}} p(c_i) \frac{|c_i|}{I(c_i)} = \sum_{c_i \in \mathbb{B}} p(c_i) \frac{|c_i|}{-\log_2 P(c_i)} \tag{21}$$

Note $I(c_i)$ denotes the information content for a specific event.

## H.1 Example

Given the following sequence $S$:

$$11122221112222$$

We compare two belief sets $\mathbb{B}_1 = \{111, 2222\}$ with $c_1$ being 111 and $c_2$ being 2222 and the second belief set $\mathbb{B}_2 = \{1, 2\}$ with $c_1$ and $c_2$ being the single atomic units in the sequence.

In the first case, $S$ will be parsed by $\mathbb{B}_1$ as $c_1, c_2, c_1, c_2$, whereas in the second case, the parsing by $\mathbb{B}_2$ will become $c_1, c_1, c_1, c_2, c_2, c_2, c_2, c_1, c_1, c_1, c_2, c_2, c_2, c_2$.

For the first example: $P(c_1) = \frac{1}{2}$, $P(c_2) = \frac{1}{2}$. Their corresponding information contents are:

$$I(c_1) = -\log_2(P(c_1)) = -\log_2(0.5) \tag{22}$$

$$I(c_2) = -\log_2(P(c_2)) = -\log_2(0.5) \tag{23}$$

The expectation of unit length explaining this sequence per bit are:

$$EUEP = \sum_{c_i \in \mathbb{B}_1} p(c_i) \frac{|c_i|}{I(c_i)} = 0.5 \times \frac{|111|}{-\log_2(0.5)} + 0.5 \times \frac{|2222|}{-\log_2(0.5)} = 3.5 \tag{24}$$

For the second example:

The information content for each chunk is:

$$I(c_1) = -\log_2(P(c_1)) = -\log_2(6/14) \tag{25}$$

$$I(c_2) = -\log_2(P(c_2)) = -\log_2(8/14) \tag{26}$$

The expectation of unit length explaining this sequence per bit are:

$$EUEP = \sum_{c_i \in \mathbb{B}_2} p(c_i) \frac{|c_i|}{I(c_i)} = 6/14 \times \frac{|1|}{-\log_2(6/14)} + 8/14 \times \frac{|2|}{-\log_2(8/14)} = 1.05 \tag{27}$$

From the above two examples, the first chunking method $B_1$ with chunks $\{c_1 = 111, c_2 = 2222\}$ is more efficient compared to the second chunking method $B_2$ with chunks $\{c_1 = 1, c_2 = 2\}$. It explains $3.5$ sequential units per bit of encoding compared to $1.05$ sequential units per bit.

# I Sensitivity Analysis

## I.1 Sensitivity to Noisy Observations

One interesting question is how sensitive is HCM's learning performance to noisy observations. To examine HCM's sensitivity to noisy observations, we generate sequences from random hierarchical generative graphs with an increasing level of depth (from 3 to 8), taking 5 sample graphs of each level of depth. Each of the random graphs is used to generate training sequences increasing from length 100 to 900.

To simulate noisy observations, we add an $\epsilon$ probability of switching to an alternative atomic unit for each unit in the sequence. That is, for each sequential element, there is a $(1 - \epsilon)$ probability that the element stays the same, and a $\epsilon$ probability of flipping to any other atomic unit with equal probability. We took an exponentially increasing level of $\epsilon$ ranging from $0$ to $0.1$, and trained the rational HCM algorithm on these noise-perturbed sequences.

Shown in Figure 8 is the learning performance with an increasing depth of the random generative model. The learned representation is evaluated on the underlying ground truth in the generative model. Learning converges in all cases. Thus, HCM's performance is fairly robust to low levels of noise.

## I.2 Sensitivity to Phase Shift

Another question we investigated was the performance of the model in response to a phase shift of the sequence, in other words, where the start of the sequence is. To investigate this question, we generated random hierarchical chunking model with depth $d = 5$ of random hierarchical graphs. For each generative model, we shifted the sequence rightwards $n$ steps, ranging from 0 to 19, and evaluated the learning performance with an increasing length of the sequence. In Figure 9, we plot the sensitivity of of HCM's learning convergence to phase shift. There was no systematic influence of phase on learning performance.

# J Comparison with PARSER

To position HCM better in relation to other models, we compared HCM's fitting on human data with the most related cognitive algorithm, PARSER.

Figure 10 shows the comparison of chunk size increase between HCM, RNN, PARSER and human behavior. HCM and PARSER continue to learn and build up longer chunks as they go through the sequence. Evaluating the average rate of chunk growth also showed that both PARSER and HCM are more similar to participants' than the RNNs. The negative log-probabilities of sequence elements generated by the HCM, RNN and PARSER were both significantly related to human reaction times (that reflect the certainty of their internal predictions). Yet the relationship was substantially stronger between HCM ($\beta = 16.74$, $p \leq 0.001$) and human participants than PARSER ($\beta = 11.070$, $p \leq 0.001$) compared to that of the RNN ($\beta = 9.24$, $p \leq 0.001$).

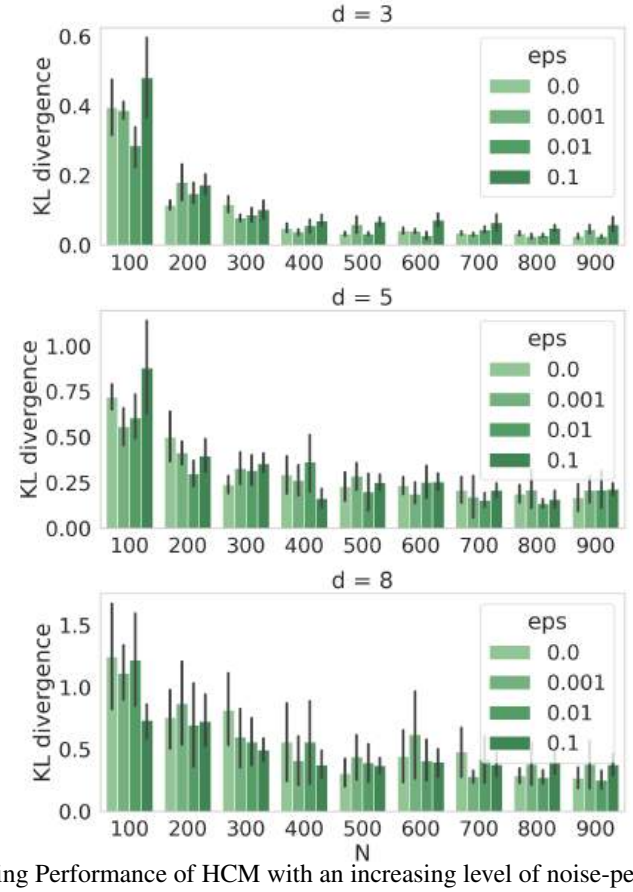


Figure 8: Learning Performance of HCM with an increasing level of noise-perturbed sequence.

## K Learning Motifs

There are situations when there is an underlying structure governing seemingly disparate sequences, for example, the sequences "12221212" and "34443434". We show a method to use HCM to learn such underlying structure. We denote such structures as motifs, and formulate a projecting function that maps a sequence from the observational space to some projected space, followed by illustrations of examples showing how the motifs can be learned by HCM. Finally, we show an experiment demonstrating HCM's motif learning ability.

**Definition 12 (*Projecting Function $f(S)$*)**

*A projecting function $f : \mathbb{O} \to \mathbb{P}$ maps a sequence in observation space to a projected space.*

The projection almost always maps from a higher dimensional space to a lower dimensional space because the intention is to discover common, overlapping parts that are shared between disparate sequences in the observation space.

**Definition 13 (*Motif*)**

*A motif $m$ is a chunk in the projected space, made up of concatenation of elements in the projected space.*

**Definition 14 (*Injection Function $g(m) \to c$*)**

*An injection function maps from motifs back to chunks in observational space $f : \mathbb{P} \to \mathbb{O}$.*

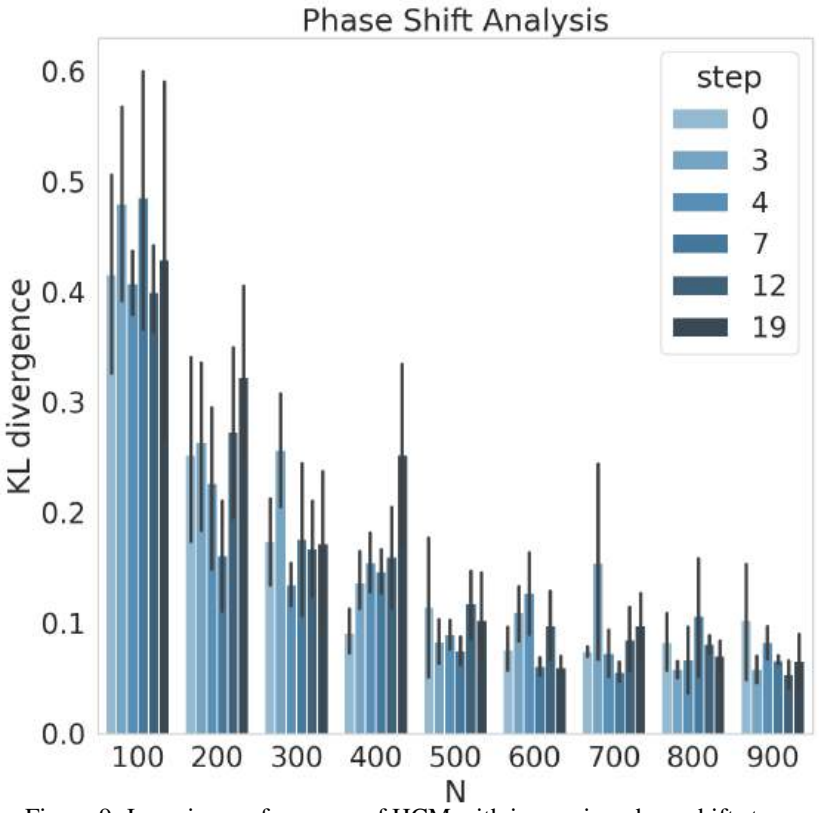

Figure 9: Learning performance of HCM with increasing phase shift steps.

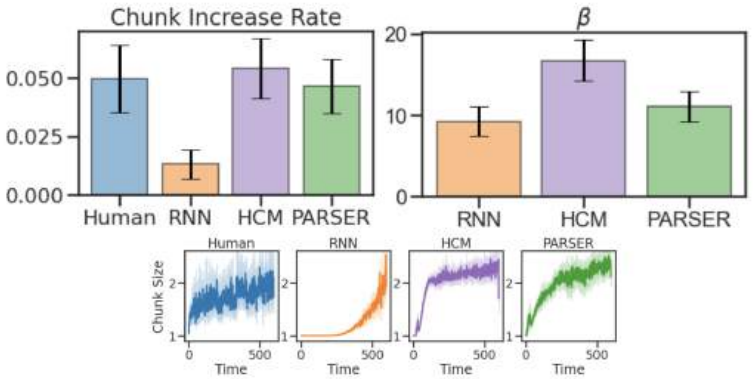

Figure 10: Model comparison for human SRT data.

### K.1 Example

We here demonstrate an example of observational sequences with any underlying motif, and illustrate why motif learning is useful for encoding or prediction. The projecting function, when applying to sequences with an underlying motif, is used so that similar motifs between projected representations can be clustered and identified as a whole.

Let's say the model has observed the following sequences: $ABBAAABBB$, $CDDCCCDDD$, and $EFFEEEFFF$: The observation set is $\mathbb{O} = \{A, B, C, D, E, F\}$, which are the set of observations that entail a chunk.

Now, the projecting function maps the first two distinct atomic sequential units separately as $X$ and $Y$, i.e. $f(A) \to X$, $f(B) \to Y$, $f(C) \to X$, $f(D) \to Y$, $f(E) \to X$, $f(F) \to Y$.

When this projection function is applied to every single chunk, then $f(ABBAAABBB)$, $f(CDDCCCDDD)$ and $f(EFFEEEFFF)$ all map to the same sequence $XYYXXXYYY$ in the projected space. The probability of observing such sequence in the projected space sums up the individual sequences in the observational space $P(XYYXXXYYY) = P(ABBAAABBB) + P(CDDCCCDDD) + P(EFFEEEFFF)$.

### K.2 Simulation Demonstration

We show a simulation to demonstrate that HCM learns motifs. In this experiment, HCM learns 40 trials of sequences. Each sequence is 12 units in length. There is an underlying structure in the motif space that generates sequences of every trial. In each trial, the sequence is made up of letters sampled from the set $\{R, B, G, T, P, Y\}$. Shown in Figure 11 as an example, sequences generated from a projected motif space can be BBBGBGBGGGGB as the first trial, and RRRYRYRYYYYR as the second trial, etc.

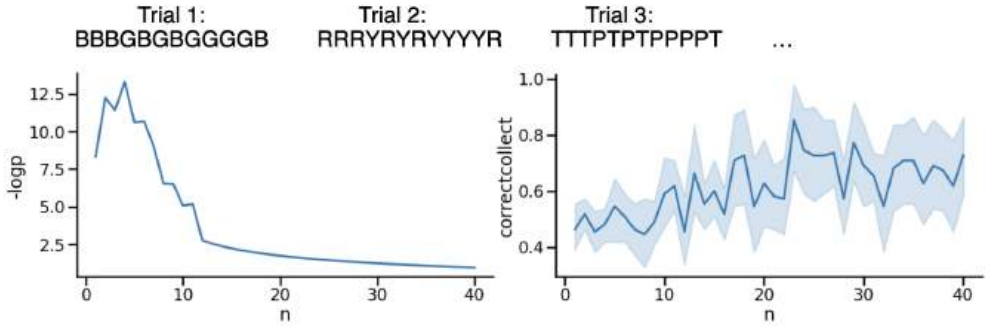


Figure 11: Learning projected motif chunks.

HCM learns from these sequences one after another. In each trial, the projecting function that maps the first two distinct elements to $X$ and $Y$ is applied to the sequence. HCM gradually learns and constructs motifs composed of $X$s and $Y$s in the projected space by constructing atomic units in the motif space and combining motifs together to bigger motifs. This way, HCM learns a belief set $\mathbb{B}$ of motifs. Each motif contains an estimated probability of occurrence, and thereby enables the evaluation of sequence information content $-\log p$ calculated by the probability of motifs that are used to parse a sequence, when the sequence is displayed to HCM. Smaller $-\log p$ implies less information content in the sequence.

The left plot of Figure 11 shows the information content contained in every sequential trial. The right plot of Figure 11 shows the imagination accuracy, which is the case that when HCM generates a sequence with the same length, the percentage of agreement with the sequence with such underlying motif. Both plots show convergence as the number of trials $n$ increases. Within 40 trials of sequences, HCM learns the underlying structure generating the sequence. Furthermore, given a novel sequence such as $TTTPTPTPPPPT$, the observation of $T$ and $P$ as distinct entities enable HCM to predict the entire sequence after observing the only the first $4$ elements, even if this sequence has never appeared before.

Code and data for the experiments are available with this link: `https://github.com/swu32/HCM`

## References


Alexandre Abraham, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gael Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8, 2014. ISSN 1662-5196. doi: 10.3389/fninf.2014.00014. URL `https://www.frontiersin.org/article/10.3389/fninf.2014.00014`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Gael Varoquaux, Alexandre Gramfort, Fabian Pedregosa, Vincent Michel, and Bertrand Thirion. Multi-subject dictionary learning to segment an atlas of brain spontaneous activity. In *Proceedings of the 22nd International Conference on Information Processing in Medical Imaging*, IPMI'11, pp. 562–573, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 9783642220913.

Shuchen Wu, Noémi Éltető, Ishita Dasgupta, and Eric Schulz. E pluribus unum but how? chunking as a rational solution to the speed-accuracy trade-off, Feb 2022. URL `psyarxiv.com/sjh27`.

# Two types of motifs enhance the recall and generalization of long sequences

communications psychology

Article

https://doi.org/10.1038/s44271-024-00180-8

# Two types of motifs enhance human recall and generalization of long sequences

Check for updates

Shuchen Wu [1] ✉, Mirko Thalmann[2] & Eric Schulz[2]

Whether it is listening to a piece of music, learning a new language, or solving a mathematical equation, people often acquire abstract notions in the sense of motifs and variables—manifested in musical themes, grammatical categories, or mathematical symbols. How do we create abstract representations of sequences? Are these abstract representations useful for memory recall? In addition to learning transition probabilities, chunking, and tracking ordinal positions, we propose that humans also use abstractions to arrive at efficient representations of sequences. We propose and study two abstraction categories: projectional motifs and variable motifs. Projectional motifs find a common theme underlying distinct sequence instances. Variable motifs contain symbols representing sequence entities that can change. In two sequence recall experiments, we train participants to remember sequences with projectional and variable motifs, respectively, and examine whether motif training benefits the recall of novel sequences sharing the same motif. Our result suggests that training projectional and variables motifs improve transfer recall accuracy, relative to control groups. We show that a model that chunks sequences in an abstract motif space may learn and transfer more efficiently, compared to models that learn chunks or associations on a superficial level. Our study suggests that humans construct efficient sequential memory representations according to the two types of abstraction we propose, and creating these abstractions benefits learning and out-of-distribution generalization. Our study paves the way for a deeper understanding of human abstraction learning and generalization.

When the iconic notes strike: GGGE♭, FFFD,—Beethoven's Fifth Symphony comes immediately to our mind. As the music progresses, we note the change of motif to GGGB or GGGC, variations in forms and voices, one at each step. Our ability to effortlessly identify those forms of abstract motifs endows us with an ability to learn mathematics, languages, and music. From representing "x" as a variable to perceiving 'noun' as a category including "cats", "dogs", and "elephants", these abstract motifs automatically come to our mind and help us to memorize sequences and generalize to novel situations. How do we abstract motifs from perceiving sequences? What advantages does this ability confer in terms of memory representations and transfer? More importantly, how do we construct an abstract representation during learning?

The literature suggests that we have the capability to learn multi-faced aspects of sequences. In what is known as grammatical judgment tasks in artificial grammar learning, after familiarizing participants to a set of grammatically valid sequences generated by a finite state language[1,2], participants acquire the ability to distinguish unseen grammatical from ungrammatical sequences[3,4]. Further research suggested that sequence learning extends beyond learning first-order transition probabilities. Frequently occurring fragments shared between the test and training sequences influence test judgment[5] and are more likely to be judged as grammatical[6–8]. Such phenomenon can be explained by models that learn repeated sequence fragments as chunks[9–12].

Beyond learning sequence fragments and transition probabilities, a few studies suggest the early cognitive capability to acquire sequential patterns on an abstract level: After familiarizing infants early as 7-month-old to sequences such as AAB and CCD, they were likelier to direct their gaze toward novel sequences sharing the same structure, such as DDF, rather than a different structure, such as KTK. Such ability to capture what was named as 'abstract algebraic structure'[13] in sequences cannot be explained by learning transition probabilities or chunks. Meanwhile, another abstract pattern has been hypothesized by linguists: we acquire sequence knowledge on a symbolic level[14–16]. This ability is a prerequisite to learning phase structures on the level of symbols such as noun phrase = determinant +

[1]Max Planck Institute for Biological Cybernetics, Tübingen, Germany. [2]Helmholtz Institute for Human-Centered AI, Münich, Germany.
✉e-mail: shuchen.wu@tuebingen.mpg.de

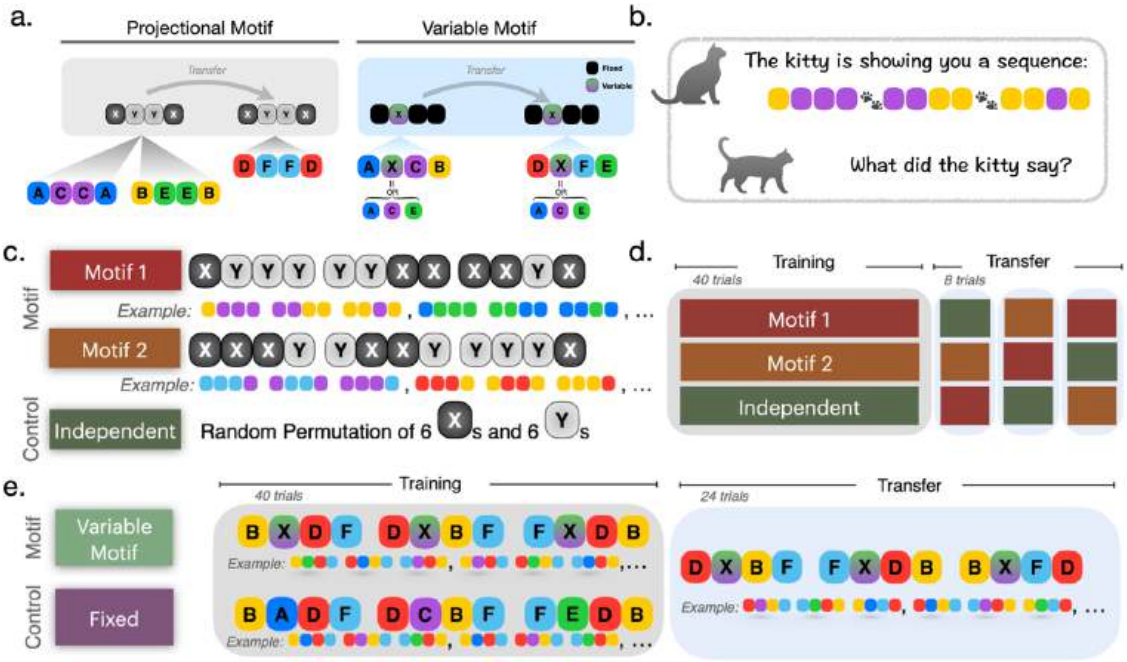


**Fig. 1 | Taxonomy of motifs and experimental design. a** A taxonomy of sequence motifs. Projectional motifs refer to patterns of sequences in a projected space that are mapped from the concrete sequence space by a projection function. In the example being shown, the projection function finds the distinct items in the sequence and maps sequential observation into a binary sequence XYYX, with X being the first unique item appearing and Y being the second. Variable motifs refer to a pattern of invariant (dark box) and variant (gradient-colored box) sequential elements. The variable motif contains at some position a variable–a symbol representing a sequential component that can vary. Such a variable is identified when any of the sequential components it represents is identified. In this example, the variable X represents the possible occurrence of A, C, or E. We hypothesized that participants could learn both types of motifs through practice and exploit their knowledge of both motif types in memorizing and generalizing to novel sequences. **b** We study motif learning in a sequence recall task. Participants are instructed to remember a sequence of 12 colors. To make the sequence more digestible, the colors are displayed one after another in three groups of four items separated by the display of pair of paws after each group. **c** Experiment 1 studies learning projectional motifs. Participants are divided into three groups. Two motif groups (Motif 1 and Motif 2) and one control group (Independent). **d** Each group is first trained on their respective motif or random sequences (Independent) and then tested on randomly interleaved transfer blocks of three types. There are no overlapping sequences between all transfer blocks and training blocks. **e** Experiment 2 studies learning variable motifs. The variable motif group is trained on sequences with an underlying variable motif. That is, the second position of each subsequence display is randomly drawn among three colors (purple, blue, or green). The fixed group is trained to recall fixed sequences. Both groups are then subsequently tested on novel sequences sharing the variable motif.

noun, and helps us to judge the grammaticity of very unlikely-occurring sentences[17].

In this work, we zoom in, refine, and categorize different forms of abstract sequential structures. We define and differentiate between two algebraic abstractions: "projectional motifs," which are patterns derived from sequences using a projectional function, and "variable motifs," which include patterns involving concrete and variable elements. We move beyond grammaticity judgements and examine the role of motifs on sequence memory and recall. We test the learning of these abstract motifs in a much longer sequences than previous work, demanding participants to gradually build up their knowledge of the motif while learning.

We study the effect of memorizing projectional and variable motifs in sequences by asking the following questions: 1. Are sequences constructed according to an underlying motif memorized more accurately than randomly generated sequences, and 2. Are novel sequences sharing the same motif recalled more accurately than random sequences? We ask these questions in two experiments, each studying one proposed motif type. Furthermore, we hypothesized that identifying structures as motifs helps to simply memory representations of long sequences. We implemented this assumption in our computational model, which continuously finds recurring motifs from distinct sequences. The model learns motifs as abstract representations incorporating components from transition probabilities, chunks, and motifs to reduce memory complexity. We look at the learning and transfer abilities of participants in comparison to the model.

## Methods

### A taxonomy of sequence motifs

We define sequence motifs as underlying sequence patterns that are not on the superficial item level but only detectable after performing transformations on sequences of items. We define and study two types of sequence motifs: projectional and variable. An illustration of the two motif types is shown in Fig. 1.

**Projectional motif** refers to a pattern in a projected space shared amongst distinct sequences. Some transformation functions can map the superficial sequential content to a projected space as illustrated in Fig. 1a, a projectional motif denoted as XYYX appears in distinct sequences ACCA and sequence BEEB shares (with X being the first appearing unique item in the sequence, and Y being the second appearing item). In relation to Beethoven's Fifth in the introduction, the music phrases GGGE♭ and FFFD contain a projectional motif XXXY.

**Variable motif** refers to a pattern containing invariant and variant parts of the sequence. A sequence with a variable motif contains at some position a variable—a symbol representing a quantity that can vary in its identity. An example is illustrated in Fig. 1a. The variable "X," described by a gradient-colored box, is an unknown entity representing the possible occurrence of A, C, or E. The same underlying variable motif appears in sequence AXCD and sequence DXFE in the example. They share the same structure of having a varying entity at the location of "X" and unchanging entities at the rest of the sequence positions. In relation to the example in the introduction: in Beethoven's Fifth, a variable motif underlies the music phrase GGGE♭, GGGB, and GGGC, which progresses with the symphony.

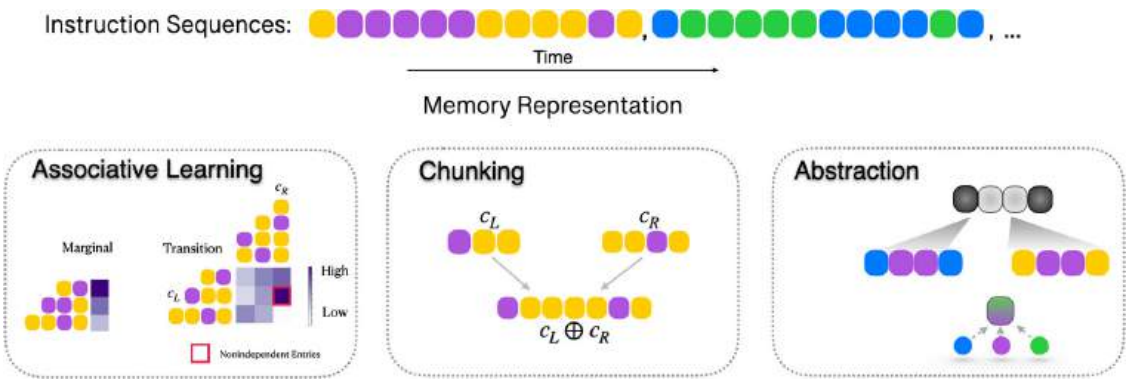


**Fig. 2 | Motif learning model.** Upon observations of instruction sequences, the model acquires the transition frequencies between the learned chunks, combines previously learned chunks into new ones, and looks for abstract representations to compress its sequence memory.

Here, we construct a model that learns abstraction in the case of projectional and variable motifs to reduce representation complexity. The model first tracks the transition probabilities in an abstract space and then gradually chunks sequential elements together. We will test the predictions of our model in two experiments.

### Motif learning model

We put forward a model (Fig. 2) that learns to memorize long sequences via a combination of three strategies: associative learning, chunking, and motif abstraction.

**Associative learning**. When a sequence is presented to the model, the model keeps track of the observational frequencies and the transition frequencies between subsequently presented items. Once an item has been identified, its occurrence frequency will increment by 1, and so will the transition frequency between the current item and the previously identified item. Meanwhile, all frequencies are subject to memory decay via multiplying the count of both the marginal and transition frequency entries by a decay parameter $\theta < 1$.

**Chunking**. Apart from associative learning, the model also remembers sequences by chunking. This part of the model is based on the hierarchical chunking model described in our earlier work (HCM[10]). The model stores learned chunks in long-term memory. These chunks are used in addition to observation and transition frequencies to parse the instruction sequence. The model keeps track of the marginal frequencies of chunks and the transitional frequencies between chunks. A new chunk is created by combining two correlated consecutively occurring chunks into a longer chunk. The combined chunk is then added to the memory of the model. This feature enables the model to learn longer and longer sequences with practice. A picture of how memory chunks are acquired during learning is: at the beginning of the training block, the model stores no sequence segments, and therefore, the model parses the first instruction sequence as 12 sequences of unitary length. These unitary sequential chunks are stored in memory as distinct units. As the model learns to combine previously learned chunks into larger chunks, these larger chunks are, in turn, used to parse the upcoming instruction sequences. During the parsing process, the memory chunk of the largest size, consistent with the upcoming instruction sequence, is identified. In this way, the longer the sequence segments the model has learned, the fewer segments are needed to parse the instruction sequence, and the further the model can predict the sequence. In this way, the model builds up a stable memory representation of sequences over practice by combining pre-existing stable representation of memory sequences in long-term memory[18,19].

We also formalize memory chunks based on their occurrence probabilities, consistent with memory models with memory strength increasing with practice[11]. The lower bound on the number of bits needed to encode this chunk $c$ to be distinguished from other chunks in memory is $-\log_2 P(c)$. The more probable that a chunk occurs in the instruction sequence, the less the memory encoding cost.

**Abstraction**. When the same sequence is presented repeatedly, subparts of the sequence will gradually chunk to combine into bigger sequence segments. However, this process is slow because it requires many repetitions to form chunks. This is especially problematic when the instruction sequence repeats rarely, as each unique sequence has only a small probability in the sequence observation space, and the number of repetitions would have to be increased for the chunking process to build up a memory of the whole sequence. We propose the learning of projectional and variable motifs as two mechanisms to reduce the complexity of memory representations.

**Abstraction via learning projectional motifs**. The model identifies two unique items to describe the sequence and assigns X to the first occurring item and Y to the second item. In this case, X and Y represent separate entities in the projectional motif space. This will be one way that the sequence can be transformed into a lower-dimensional space, in which only two dimensions exist.

Once observational sequences are projected onto a lower dimensional projectional motif space, the model learns the sequence via associative learning and chunking and builds up memory representations of sequences by combining correlated consecutively occurring chunks in the projectional motif space.

For example, upon seeing ACCC, BDDD, and FEEE sequences, the model will map all three sequences onto the same sequence in the projectional motif space: XYYY. Originally, there needed to be six dimensions to describe the observational sequence, each representing the binary indicator of observing each letter. The abstraction process enables all three sequences to be described by the same pattern in an abstract projectional space. Without abstraction, if each of the three sequences occurs uniformly likely, then the minimal encoding length to distinguish between the three subsequences shall be $-\log P(\frac{1}{3})$. But once the projectional motif has been identified, it explains all observational sequences and demands significantly less encoding memory of $-\log P(1)$.

**Abstraction via learning variable motifs**. Under the demand of learning to remember long sequences, an alternative way to compress sequence representation is to learn variables. A variable is an abstract sequence entity that entails a set of concrete sequence entities/chunks. The model identifies the variable identity whenever any of its entailing entities is identified.

The abstraction model discerns variables by analyzing the structure of the transition matrix. Specifically, the model identifies structural patterns within a series of sequential observation chunks that share a common precursor and successor. For instance, if the model observes that entity A

transitions to B, C, and E, and further notes that B, C, and E each transition to F (as reflected in the transition matrix), it will recognize a new variable encompassing B, C, and E. This variable becomes identifiable when any of the elements B, C, or E are detected.

Once a variable entity has been learned, it is parsed and identified as one entity to join forces with associative learning and chunking. In this way, the variable helps the learning agent discover an overarching pattern in the sequence, which would otherwise demand more sequence observations to be learned as separate memory chunks.

The mechanism of variables naturally leads to sequence compression. For example, assume the following subsequences: BADF, BBDF, and BCDF have been observed to occur equally likely; each subsequence demands a minimal encoding complexity of $-\log P(1/3)$. As soon as a variable X is identified to entail A, E, or C, then the chunk BXDF would suffice to explain all three observational instances, and this chunk demands a minimal encoding length of $-\log P(1)$.

The model learns memory pieces by combining chunking and associative learning. On top of that, sequence abstraction processes, including projectional transformation and identifying variables, help the model to locate recurring motifs in the abstract space, capable of explaining a larger number of sequence observations and thereby learning faster and compressing further.

A natural benefit of learning abstract motifs is generalization to novel, unseen sequences sharing the same motif structure. The previously learned projectional or variable motifs can be reused to remember novel sequences, facilitating novel sequence acquisition and compression.

The model predicts that participants looking for the minimal complexity representation to learn sequences should behave in the following ways:

- When there are underlying projectional or variable motifs in the sequence, participants' representation of the sequence shall decrease in complexity when more sequences are presented with the same motif type.
- Participants who benefit from learning motifs from training sequences will exploit their previously learned motif structure.
- In the case of projectional motif, motif structure that has been learned before will be exploited to memorize a novel sequence that has never been observed by participants.
- When participants learn the representation of a variable and extrapolate it as a sequential unit to be combined with the unvarying part of the sequence, the variable as a concept will be reused when novel sequences share the same variable.

We will test these predictions in detail in the following two experiments.

### Ethics statement

Informed consent was obtained from all participants before participation, and the experiments were performed following the relevant guidelines and regulations approved by the ethics committee of the University of Tuebingen (Ethik-Kommission an der Medizinischen Fakultät der Eberhard-Karls-Universität und am Universitätsklinikum Tübingen), under the study title: Experimente zum Sequenz- und Belohnungslernen, with application number 701/2020BO.

Participants' data were analyzed anonymously. Upon agreement to participate in the study, they consented to a data protection sheet approved by the data protection officer of the MPG (Datenschutzbeauftragte der MPG, Max-Planck-Gesellschaft zur Förderung der Wissenschaften).

### Paradigm

Specifically, six equally distanced squares are horizontally placed on the display. Each assumes a distinct color: blue, yellow, magenta, red, green, and teal and corresponds to one legitimate key on the keyboard: S, D, F, J, K, and L. Participants were instructed to place their fingers stationarily on these designated keys throughout the task (left index finger on D, left middle finger on S, left ring finger on A, right index finger on J, right middle finger on K, and right ring finger on L). To control for finger familiarity biases, a random mapping from keyboard position to color is generated for each participant.

Before the start of each trial, all colors were initially covered by dark shades. The sequence was then presented sequentially by revealing each color for 800 ms followed by a brief re-covering of dark shades for another 200 ms before the next display color. The colored sequence was presented in three groups of four, separated by pauses of 800 ms accompanied by the display of a pair of paws, akin to the structure of a three-prose-poem with four words in each prose and pauses in between.

Following the sequence display, participants were prompted to recall the instructed sequence by pressing the corresponding key. Upon the press of each key, the shade covering the corresponding color would disappear and the color would be revealed for 200 ms. At the end of each group, a pair of paws would appear to signify the completion of one subsequence. At the end of the third recall group, participants received immediate feedback on their recall accuracy and recall time which marks the completion of one trial. Participants were instructed to prioritize both speed and accuracy and received a performance-based bonus based on both factors. Before the official trials, participants completed a practice trial to familiarize themselves with the task. There was no preregistration of this study.

### Recruitment of participants

We recruited 135 participants for Experiment 1 from Prolific, an online crowd-sourcing experimental platform. Out of all participants, thirty-seven were female, ninety-eight were male. Participants' ages ranged from 18 to 67, with an average of 32 and a median of 28. The experiment took an average of 45.06 minutes to complete. As compensation, participants received a base pay of £4 and another performance-dependent bonus up to £4. The average hourly pay for the study was £11.60.

We recruited 120 participants for Experiment 2 from Prolific, out of which thirty-four were female, eighty-six were male. Participants' ages ranged from 19 to 63, with an average of 31.2 and a median of 28. The experiment took an average of 47.55 minutes to complete. As compensation, participants received a base pay of £4 and another performance-dependent bonus up to £4. The average hourly pay for the study was £10.89.

Across both experiments, we did not collect participants' race/ethnicity data.

### Payment

For both experiments, participants receive feedback about their trial-wise bonus, which is dependent on a mixture of their sequence recall accuracy and reaction time and is ceiled to the maximum bonus divided by the number of trials. The reaction time bonus becomes the maximum when the recall reaction time is less than 2000 ms, and is set to 0 when the recall reaction time exceeds 10000 ms. For reaction time in the middle, the bonusfast is calculated as $bonusfast = bonusmax - (10000 - trialrt)/(10000 - 2000) \times maxtrialbonus$. In this way, a reaction time between the two limits will yield a steady bonus increase.

The trial-wise bonus for accuracy is calculated as follows: when the recall accuracy is perfect, the bonusacc is set to maxtrialbonus. And when the recall accuracy is below 50%, which corresponds to more than 6 of the recalled sequences in a false order or a false recalled item, then the bonusacc for this trial is set to 0. A recall accuracy in between will yield a bonusacc calculated as $bonusacc = bonus_{,,}ax \times (trialacc - 0.5)/(1 - 0.5)$.

Finally, the trial bonus is calculated as an average of the reaction time bonus and the recall accuracy bonus $trialbonus = 0.5 \times bonusfast + 0.5 \times bonusacc$.

At the end of the experiment, trial-wise performance-dependent bonus was summed up to the total amount of bonus that participants will receive.

### Filtering

We applied the same filtering criteria on the training blocks for all groups as a basis to exclude participants: mean reaction time < 10,000 ms (that is

10 seconds to press a sequence of 12 made of two distinct colors), mean recall accuracy ≥ 50%. On top of that, we measured whether participants were learning by inspecting reaction time decrease, as a violation of a decrease in rt would be an indication of distraction during the study. Data distribution was assumed to be normal but this was not formally tested. When applying a linear regression model regressing trial number on reaction time on participant's data during the training blocks, the reaction time should on average, decrease, which translates to having a significant ($p < 0.05$) of a negative beta coefficient. No filtering criteria were applied to the transfer blocks. Filtering excludes 21.4% (29) of participants out of 135. After filtering, 37 participants are left in group m1, 41 in m2, and 28 in group independent. The average accuracy was 0.80 ± 0.22, and the average reaction time was 5446 ± 3723 ms.

For experiment 2, we excluded participants who took on average more than 20 seconds to recall a sequence during the training block (since experiment 2 employs more colors than experiment 1, we also relaxed this exclusion criteria accordingly). Since the motif condition is harder than the control condition, we applied different exclusion criteria for the two groups, and excluded participants with an average sequence recall accuracy below 50% in the fixed group (as they have to recall the same sequence repeatedly), and below 20% in the variable motif group. Additionally, we excluded people who do not have a significant reaction time decrease ($p < 0.05$) during the training block—an indicator of not learning during the task. The exclusion criteria apply only to the training blocks and no participants are excluded based on their transfer block performance. 23 participants were excluded given that they have violated any of the above-mentioned criteria. After exclusion, 45 participants out of 120 remained in group m1, and 52 remained in group control. The average accuracy was 0.70 ± 0.28, and the average reaction time was 8094 ± 6209 ms.

**Sequence recall**. The model receives the same instruction sequences to participants as its training sequences, except that the middle pauses were removed. To recall, the initial item of the sequence is used as a primer for the model to recall subsequent sequential items. Based on the sequence segments stored in the model, it samples from the set of sequence segments that are consistent with the sequence prime while giving priority to sampling larger segments. Once the first sequential segment is sampled, the segment becomes the previous item to sample the next segment, which is based on the transition given the occurrence of the previous segment. The recall complexity is evaluated by calculating the sampled probability of the recalled sequence. $P(c_1, c_2, c_3) = P(c_1)P(c_2|c_1)(c_3|c_2)$, calculated from the marginal and conditional frequencies are both stored in the model.

**Random effect structure of regression analysis**. To obtain the maximal random effect structure justified by design without inflating the Type I error rate[20], while balancing the loss of statistical power[21], we systematically select models across multiple possible random effect structures and report the best model that is supported by data. Specifically, when fitting linear mixed effect logistic regression on keypress correctness, we compared across random intercept per participant, random slope per serial position, and trial ID, and always reported the best fitting model that includes any subset of the three random effects.

### Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

## Results

In Experiment 1, we tested whether people can learn and transfer sequences described by a projectional motif as shown in Fig. 1. In Experiment 2, we tested whether participants remember novel sequences better when these sequences share the same variable structure as shown in Fig. 1.

Taken together, we implemented learning structured motifs as a memory compression strategy in a computational model. The model exhibits similar learning and transfer behavior to participants in two sequence recall experiments testing each motif type.

### Experiment 1: projectional motifs

Experiment 1 tested how projectional motifs could help memorization and transfer by instructing participants to memorize long sequences. In a sequence recall task, participants were instructed to play a memory game and to memorize 12 consecutively displayed colors by a cartoon cat. After the instruction, they had to recall the sequence by pressing the keys corresponding to the colors.

Unbeknownst to the participants, the instruction sequences contained underlying motifs. As shown in Fig. 1, the motifs consisted of two distinct variables, X and Y, and individual motifs were constructed by arranging patterns of Xs and Ys. All sequences contained an equal amount of 6 Xs and 6 Ys to control for stimulus-specific habituation effects. Each participant was randomly assigned to one of the two motif groups (Motif 1; Motif 2), or to a control group (Independent). Motif 1 followed the pattern XYYY YYXX XXYX, while Motif 2 adhered to the format XXXY YXXY YYYX. In the motif groups, the underlying motif remained consistent across trials. Conversely, in the Independent group, a permutation of 6 Xs and 6 Ys was generated for each trial. The instruction sequences were finalized by mapping X and Y to two distinct colors.

The task was divided into training and transfer blocks. The training block comprised 40 trials, after which participants proceeded to three randomly ordered transfer blocks, each testing for Motif 1, Motif 2, and the Independent sequences with 8 trials. The transfer phase occurs immediately following the training phase without explicit notification. In all trials, participants were instructed to recall sequences by consecutively typing keyboard keys corresponding to the displayed item until the length of the instruction sequence was reached. Within a trial, the response of individual key presses is recorded. The number of key-press errors is calculated by evaluating the hamming distance (the minimum number of substitutions required to change one string into the other) between the recalled sequence and the instruction sequence. The trial-recall accuracy was calculated by evaluating the proportion of positions at which the corresponding keys are the same. After participants finish recall, the trial-wise accuracy is displayed in addition to the bonus corresponding to the current trial. Participants are not informed about their specific mistakes or the position where they have made the mistake. To ensure that no sequences in transfer blocks appeared in the training block, the six colors were divided into two sets: the training set with four colors and the transfer set with the remaining two colors.

#### Model prediction

Reducing representation complexity through projectional motifs. In the case of projectional motifs, a rational agent that looks for minimal complexity representations shall acquire the unchanging motifs during learning since motifs in the abstract projectional space explain more instances of sequences compared to memorizing concrete sequence instances.

Our hypothesis posits that an underlying motif within training sequences in a projectional space will enhance memory and out-of-distribution transfer. In this context, a sequence of length n can be conceptualized as a point within an n-dimensional space, and out-of-distribution refers to the capacity to transfer the representation to sequences never encountered during training. We anticipate improved learning and memorization performance during training for both motif groups and positive transfer when the two groups are tested on motifs of the same type.

#### Training

Behavioral results. We first compared sequence recall accuracy amongst the three groups in the training block as shown in Fig. 3a. We fitted a linear mixed-effects regression model onto participants' trial-wise sequence recall accuracy, assuming a random intercept over participants and excluded trials that were immediate repetitions. We observed a significant effect of group ($\chi^2(2) = 10.85$, $p = 0.004$, Conditional $R^2 = 0.22$), suggesting that participants in the Motif 1 group

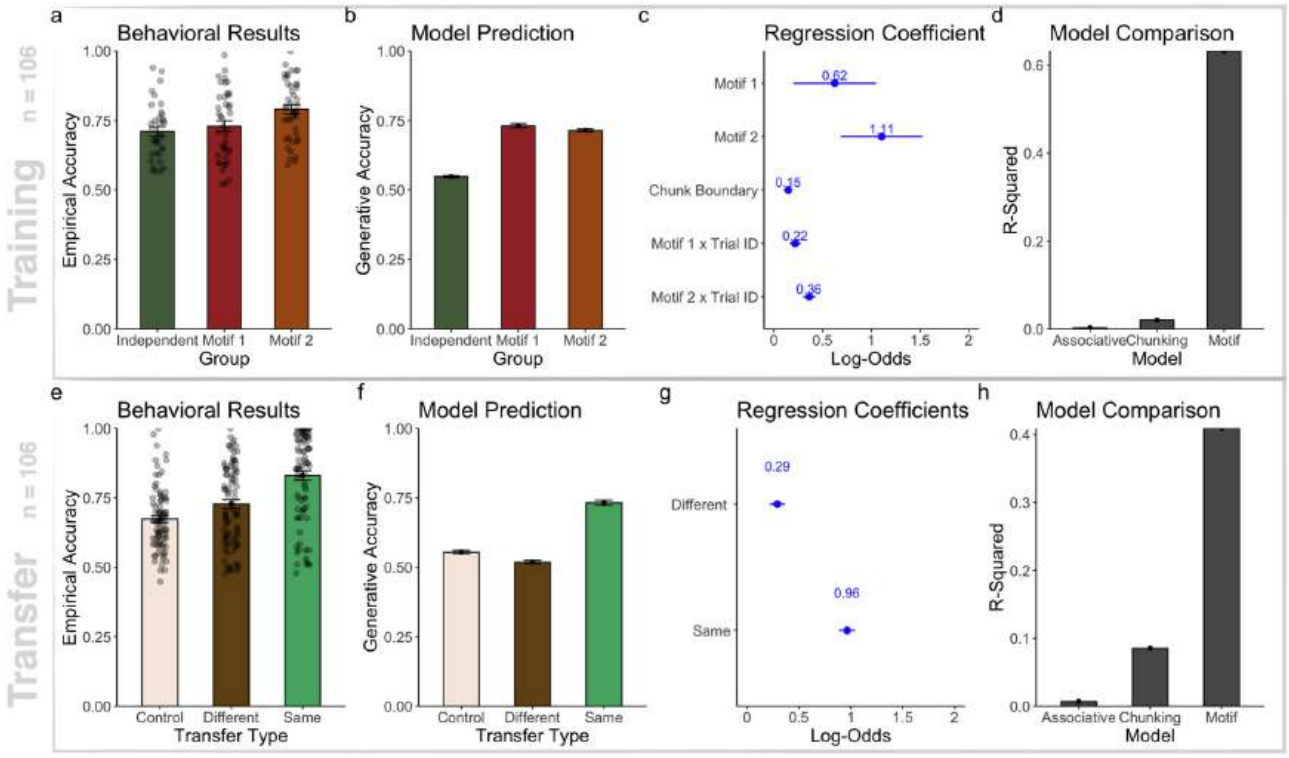


**Fig. 3 | Model simulation and behavioral results for learning and transferring projectional motifs. a** Recall accuracy is higher during the training block in the motif groups than in the control group. **b** Model prediction of sequence recall accuracy training on participants' instruction data. **c** Regression coefficients of the linear mixed-effect model predicting recall accuracy during the training block. **d** Generative accuracy as simulated by the motif learning model correlates with the empirically observed sequence recall accuracy across groups during the training trials. **e** Behavioral results of group-wise recall accuracy across three categories of transfer. Same: Motif 1—Motif 1 and Motif 2—Motif 2; different: Motif 1—Motif 2, and Motif 2—Motif 1; control: Independent—Motif 1, and Independent—Motif 2. **f** Simulation transfer results. **g** Beta coefficients of the logistic regression predicting recall accuracy during the transfer blocks. **h** Correlation between the simulated recall accuracy and participants' recall accuracy.

($\hat{\beta} = 0.02, se = 0.02, t(109) = 0.7, p = 0.46$, 95% CI = −0.03 to 0.07) and the Motif 2 group ($\hat{\beta} = 0.07, se = 0.02, t(109) = 3.18, p = 0.002$, 95% CI = 0.02 to 0.13) recalled sequences more accurately during the training blocks than those in the Independent group.

Model simulation. We compared the behavioral results with the model predictions. We used the same sequences instructed to the participants to train the motif learning model, which creates memory representations of sequence motifs from the observational sequences in an abstract space. We then generated sequences based on the representations learned by the model up to the current time point. We came up with generative accuracy as a surrogate for sequence recall accuracy. The generative accuracy was the edit distance between a generative sequence sampled from the model and the instruction sequence in a particular trial. Figure 3b shows the average generative accuracy of the model. We observed a significant effect of group ($\chi^2(2) = 216.23$, $p < 0.001$, Conditional $R^2 = 0.13$), suggesting that participants in the Motif 1 group ($\hat{\beta} = 0.18, se = 0.01, t(118) = 22.15, p<0.001$, 95% CI = 0.16 to 0.20) and the Motif 2 group ($\hat{\beta} = 0.17, se = 0.01, t(119) = 20.23, p<0.001$, 95% CI = 0.15 to 0.18) recalled sequences more accurately during the training blocks than the independent group. Similar to participants, the model remembered sequences with underlying motifs more accurately.

Regression coefficient. Apart from having higher average recall accuracy, both motif groups improved their recall accuracy faster. As shown in Fig. 3c, we analyzed participants' recall key-press correctness by fitting a logistic regression model assuming a random intercept of each participant and a random slope over individual serial positions (explanation on random effect structure selection in method section Random Effect Structure of Regression Analysis). We observed an effect for both Motifs (for Motif 1: $\beta = 0.62, se = 0.21, z = 2.88, p = 0.003$ 95% CI = 0.20 to 1.05; for Motif 2: $\beta = 1.10, se = 0.21, z = 5.17, p < 0.001$, 95% CI = 0.69 to 1.53). Apart from that, we observed an interaction effect between the trial number and group ($\chi^2(2) = 51.69, p < 0.001$). Participants in the Motif 1 group improved their recall accuracy at a faster rate than participants in the Independent group ($\beta = 0.21, se = 0.03, z = 7.55, p < 0.001$, 95% CI = 0.16 to 0.28); the same effect was present for the Motif 2 group ($\beta = 0.36, se = 0.03, z = 11.74, p < 0.001$, 95% CI = 0.30 to 0.42). Thus, people improved faster on remembering sequences with fixed motifs than sequences without.

Model comparison. We compared the recall accuracy of the motif learning model with two alternative models: an associative learning model and a chunking model. The motif learning model constructs memory pieces by combining chunking, associative learning, and abstraction via learning projectional motifs. The chunking model contains the same components except for abstraction. The associative learning model learns the first-order transition between observed sequential items. We gave the same instruction sequence to all three models and thereby arrived at an average recall accuracy for each model on each proceeding experimental trial. To do so, we used the same sequences instructed to the participants to train all models. After updating memory components from each trial of sequences, the memory components of the model are used to generate sequences that emulate recall. Then, the model recall accuracy on a particular trial is calculated as the percentage of matching items in the recalled sequence by the models and the instruction sequence. After that, we calculated the group accuracy progression (averaging across participants) for both the model-simulated performance and the participants' performance. The average

generative accuracy per trial of the models is compared to the average recall accuracy per trial of the participants.

We then regressed the generative accuracy of each model onto empirical accuracy and evaluated the goodness of fit by computing the R-squared value. The R-squared measure determines the proportion of variance in the behavioral results that the model prediction can explain and shows how well the data fit the regression model. As shown in Fig. 3d, the motif learning model ($R^2$ = 0.63, 95% CI = 0.55 to 0.71) explained more variance in the behavioral result than a chunking model ($R^2$ = 0.02, 95% CI = 0 to 0.12) that did not abstract. This suggests that abstracting the sequence via projecting the sequence onto the motif space is a critical component that captures human behavior in this task.

Comparing the motif learning model to an associative learning model shows that abstraction alone isn't enough to explain the results. The associative learning model factors in marginal and transition probabilities in the sequences but doesn't learn chunks. Additionally, it explains very little of the variance in human behavior, with $R^2$ = 0.003, 95% CI = 0 to 0.06, compared to the motif learning model. This result suggests that learning the association between items in the projected motif space is insufficient; combining the previously memorized memory chunks into longer memory chunks is also vital to explaining human learning progress.

**Transfer**. We then assessed whether training on motifs affected participants' ability to memorize novel sequences in the transfer blocks.

Behavioral results. We compared participants' performance in the transfer blocks grouped by three transfer types relative to the training block types: Same (Motif 1 - Motif 1, Motif 2 - Motif 2), Different (Motif 1 - Motif 2, Motif 2 - Motif 1), and Control (Independent - Motif 1, Independent - Motif 2). Shown in Fig. 3e, we observed a significant effect of transfer type ($\chi^2(2) = 91.43$, $p < 0.001$, Conditional $R^2 = 0.63$) on recall accuracy. Participants remembered novel sequences with the same motifs more accurately compared to control ($\hat{\beta} = 0.16, se = 0.01, t(168) = 10.78, p<0.001$, 95% CI = 0.13 to 0.19). Surprisingly, we also observed that participants benefited from transferring to a different motif type compared to control ($\hat{\beta} = 0.05, se = 0.01, t(168) = 3.69, p<0.001$, 95% CI = 0.02 to 0.08). Consistent with our hypothesis, training on sequences with motifs helps participants learn novel sequences sharing the same motifs. Participants' reaction time data is also analyzed and visualized in Supplementary References and Figure S1, S2.

Model prediction. Similarly, we evaluated the recall accuracy of the motif learning model on the transfer blocks. Figure 3f shows the generative accuracy of the motif learning model grouped by transfer types. Similar to participants, the model recalled novel sequences with motifs better after it had been trained on the same motif ($\chi^2(2) = 265.43$, $p < 0.001$), compared to having been trained on neither motif ($\beta = 0.18$, $se = 0.01$, $t = 16.69$, $p < 0.001$, 95% CI = 0.16 to 0.2). Different from the participant: it is harder for the model to transfer to an alternative motif type ($\beta = -0.04$, $se = 0.01$, $t = -3.35$, $p < 0.001$, 95% CI = -0.06 to -0.02) than the control. We inspect this discrepancy further in the discussion section.

Regression coefficients. We looked at participants' correctness of recall key presses by fitting a logistic regression model, assuming a random intercept of participants and random slope over individual serial positions and trial numbers (Fig. 3g). We found that the transfer types affect the recall key press correctness ($\chi^2(2) = 679.46$, $p < 0.001$, Conditional $R^2 = 0.23$). Participants who have been tested on the same motif as they had been trained on (m1–m1 and m2–m2) ($\beta = 0.96$, $\sigma = 0.04$, $z = 24.16$, $p < 0.001$, 95% CI = 0.89 to 1.04) are more likely to recall the correct item compared to control. This result resonates with our linear mixed-effect analysis on recall accuracy. Interestingly, participants tested on a motif different from their training motif also did better than the control ($\beta = 0.29$, $\sigma = 0.04$, $z = 7.90$, $p < 0.001$, 95% CI = 0.22 to 0.36). We discuss the implications of this finding further in the discussion section. Additional regression coefficients that confirm the practice effect, recency effect, and chunk boundary effect are reported in the Supplementary Reference file.

Model comparison. We then compared the resemblance to human behavior between the motif learning model, the associative learning model, and the chunking model (Fig. 3h) during the transfer blocks. Since all three models change their representation when the training schedule switches from training to the transfer blocks, we can compare the generative accuracy of the models to participant recall accuracy. This feature allows us to regress the generative accuracy of each of the three models onto empirical recall accuracy per transfer trial and evaluate the R-squared of the regression as a goodness-of-fit measure.

The motif learning model ($R^2$ = 0.41, 95% CI = 0.26 to 0.55) explains more variance of participants' transfer performance compared to the chunking model ($R^2$ = 0.08, 95% CI = 0.003 to 0.27), suggesting that projecting sequences in a projected motif space, an abstraction process, is critical to capture human behavior in this task. The motif learning model also explains more variance than the associative learning model ($R^2$ = 0.05, 95% CI = 0 to 0.10). Associative learning only is insufficient to capture participants' transfer behavior.

## Experiment 2: variable motifs

Experiment 2 tested the learning and transfer of variable motifs in the sequence recall paradigm. A training block of 40 trials was followed by a transfer block of 24 trials. Participants were split into two groups: the variable motif group (motif) and the fixed group (control). The variable motif group was instructed to remember sequences with variable motif B X D F, D X B F, F X D B (1). X represents a variable and randomly assumes a letter amongst A, C, and E with equal probability with every occurrence. The fixed group was instructed to remember unchanging sequences assuming the form: B A D F, D C B F, F E D B.

During the test block, both groups were instructed to remember a novel sequence with an embedded variable X: D X B F, F X D B, B X F D. The location and entailment of X were the same as the training sequence with variables, but we changed the fixed part of the sequence.

Similar to Experiment 1, the transfer phase proceeds immediately following the training phase without explicit notification. Sequence recall instruction, accuracy evaluation, and feedback are identical to Experiment 1.

**Model prediction**. We hypothesize that when participants are instructed to memorize sequences with a component that varies, identifying variable entities and memorizing them in conjunction with the unvarying part of the sequence should facilitate transfer. That is, when participants encounter novel sequences sharing the same variable entity but different unvarying parts, they should memorize novel sequences with overlapping variables better compared to the control group.

### Training

Behavioral results. Figure 4a shows the average sequence recall accuracy of the variable motif group and the fixed group. We fitted participants' sequence recall accuracy with a linear mixed-effects regression model, assuming a by-participant random intercept. The result showed a significant effect of group ($\chi^2(1) = 50.012$, $p < 0.001$, Conditional $R^2 = 0.42$). The fixed group recalled sequences more accurately than the variable motif group ($\hat{\beta} = -0.22, se = 0.03, t(95) = -0.806, p<0.001$, 95% CI = −0.28 to −0.17). A changing part of the instruction sequence hindered recall.

Model prediction. We trained the variable motif learning model on the same instruction sequences seen by participants. For sequences with the variable motif, the model learned memory representation manifested in chunks and variables. To do so, the model condensed observations of disparate instances of A, C, and E into one variable entity and concatenates the variable entity with the already-acquired fixed sequence parts in its memory. In this way, the motif learning model learned to represent instruction sequences with variable motifs as a chunk with embedded variable entities. Hence, the

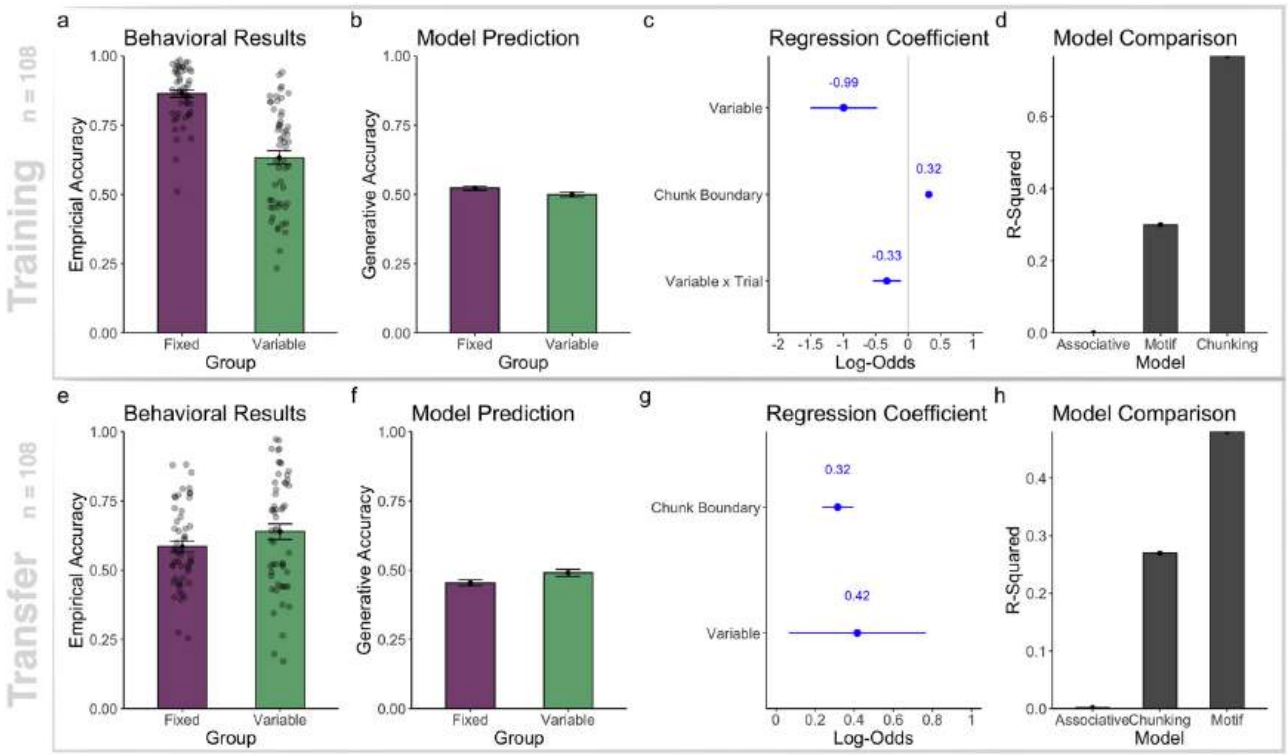


**Fig. 4 | Model simulation and behavioral results for learning and transferring variable motifs. a** Recall accuracy across groups during the training blocks. **b** Simulated recall accuracy during the training blocks. **c** Beta coefficient of a linear mixed effect logistic regression on recall key press correctness during the training blocks. **d** Correlation between simulated generative accuracy and participants' recall accuracy. **e** Recall accuracy across groups during the transfer blocks. **f** Regression coefficients of logistic regression performed on recall keypress correctness during the transfer block. **g** Correlation between simulated transfer generative accuracy and participants' sequence recall accuracy. **h** Correlation between training improvement (average recall accuracy difference between the last five training trials and the first five training trials) and the average recall accuracy during the initial 5 trials of the transfer block.

memory contains both concrete and abstract sequence parts as a low-complexity sequence representation. For control sequences, the model constructed memory pieces by chunking. During recall, sampling entailment chunks of a variable entity introduces memory recall error ($\chi^2(1) = 3.72$, $p < 0.001$, Conditional $R^2 = 0.03$). The motif learning model recalled sequences with variable motifs less accurately than fixed sequences ($\hat{\beta} = -0.02, se = 0.01, t(106) = -1.93, p<0.05$, 95% CI = -0.04 to 0).

Regression coefficient. We then studied factors that influenced the keypress correctness via fitting a logistic mixed-effects regression, assuming a per-participant random intercept and a random slope per serial position (Conditional $R^2$ = 0.32). Shown in Fig. 4c, the regression coefficient suggested that the variable motif group was more prone to recall mistakes than the fixed group ($\beta = -0.99, se = 0.26, z = -3.78, p < 0.001$, 95% CI = −1.51 to -0.48). Apart from that, the variable motif group learned sequences slower than the fixed group ($\beta$ = −0.33, $se$ = 0.11, $z$ = −2.98, $p$ = 0.002, 95% CI = −0.54 to −0.11). Training on sequences with variables decreased participants' probability of recalling the correct key and slowed down learning. Overall, the regression analysis was consistent with our predictions.

Model Comparison. We again compared the motif learning model with an associative learning model and a chunking model by evaluating the R-squared value regressing simulation recall accuracy onto empirical recall accuracy in the same way as in Experiment 1. Figure 4d shows the goodness-of-fit model comparison on the training blocks.

The associative learning model ($R^2$ = 0.0005, 95% CI = 0–0.08) explained very little variance in participants' recall accuracy progression during learning, suggesting that just learning the first-order transition probability was insufficient to explain participants' learning curve on memorizing sequences with variables. Having a chunking component that builds up recall memory pieces together was essential to explain participants' learning progression. Meanwhile, we observed that the chunking model ($R^2$ = 0.76, 95% CI = 0.65 to 0.86) explained more variance of recall accuracy progression than the variable learning model ($R^2$ = 0.39, 95% CI = 0.14–0.47), possibly because the average chunking process becomes more predictive of participants' recall accuracy than the average variable learning process, as participants may have learned variables in idiosyncratic ways that are not captured by the variable discovery process of the model but are described better by a chunking model.

### Transfer

Behavioral results. We hypothesized that participants transfer variable representations from the training to the test block. Shown in Fig. 3e is the average recall accuracy of the two groups across all transfer trials. We used an independent-sample t-test to assess the performance difference between the two groups, and a two-sided t-test to assess the superiority of the variable group compared to the fixed group in sequence recall. We observed a significant difference ($t(2317.4) = 4.99$; $p < 0.001$; $95\% CI = [0.033, 0.076]$) in recall accuracy between the motif group ($M$ = 0.64) and the control group ($M$ = 0.58), supporting our hypothesis that the variable group performs better at transfer than the fixed group. Participants' reaction time data is also analyzed and visualized in Supplementary References and Figure S3.

Model prediction. As per model simulation shown in Fig. 4f, generative accuracy was higher for the model trained on variable sequences than those trained on fixed sequences ($\hat{\beta} = 0.04, SE = 0.02, t(106) = 2.11, p = 0.03$) ($\chi^2(1) = 4.47, p = 0.03$, Conditional $R^2 = 0.03$). This transfer advantage results from the variable learning model reusing the previously learned variables to parse and chunk in conjunction with the novel sequence part. In other words, the model trained on sequences with variables learned to ignore a

certain part of the novel sequences to afford memorizing the unchanging sequence part.

Regression coefficients. We fitted a mixed-effect logistic regression on participants' recall key press correctness in the transfer block, assuming a per-participant random intercept and a logit link function (Conditional $R^2$ = 0.30). Shown in Fig. 4g, we observed a positive effect of train condition ($\beta$ = 0.42, $se$ = 0.18, $z$ = 2.33, $p$ = 0.02, 95% CI = 0.07 to 0.77). Training on sequences with variable motifs helped participants recall novel sequences sharing the same variable motif better than the control group trained on fixed sequences, which was consistent with our model's prediction.

Model comparison. We compared the motif learning model with the chunking and associative learning model on the transfer block. Shown in Fig. 4h, we observed that the motif learning model that reuses its previously learned variables to memorize novel sequences explains the most human recall accuracy variance ($R^2$ = 0.48, 95% CI from 0.33 to 0.65) than the chunking ($R^2$ = 0.26, 95% CI from 0.13 to 0.44) and the associative learning model ($R^2$ = 0.001, 95% CI from 0.0 to 0.16). This aspect suggests that reusing previously learned variables to memorize novel sequences captures a part of the human sequence memory variance when they transfer to novel sequences.

Training improvement correlates with transfer performance. We also assessed the effect of training improvement on transfer performance for both experimental groups. The improvement measure is evaluated on individual participants' sequence average recall accuracy between the last five trials at the end of the training block, subtracted by the first five trials at the beginning of the training block. This difference reflects the average improvement over the training period for every participant. We observed a significant interaction between training improvement and group ($RSS$ = 2.44, $F(1)$ = 10.42, $p$ = 0.001) affecting transfer recall accuracy. Participants who improved more during training on variable motifs performed better during the initial transfer blocks, compared to control ($\beta$ = 0.53, $se$ = 0.17, $t$ = 3.22, $p$ = 0.002). Training improvement on variable motifs facilitated transfer to sequences sharing the same variables.

## Discussion

We effortlessly perceive and extract motifs in music, acquire grammatical structure from languages, and use mathematical variables to find out about the unknown. Already during early childhood, we can learn abstract concepts as soon as we learn concrete concepts[22,23]. Linguistics suggest that the conceptual metaphor — mapping similar structural concepts of a known thing to construct an understanding of an unknown concept — plays a vital role in human understanding and reasoning[24,25]. Having seen a solution to a problem, people can solve problems in a similar conceptual relational space[26]. Abstraction as a principle has demonstrated its usage in mathematics and machine learning. Mathematicians have used abstraction as a mapping principle to transfer deductions from one formal system to a new formal system[27]. Abstraction has long been postulated as a crucial requirement for intelligent agents to solve problems in diverse situations[28]. Reinforcement learning studies suggest that state or action abstraction makes the representation more compact, easier to plan, and generalize flexibly to different environments and across tasks[29–33]. Yet, current artificial intelligence systems do not explicitly abstract in the way that humans do[34]. Hence, understanding how humans arrive at abstraction more generically has wide and profound implications in the study of artificial and natural intelligence.

As the key to generalization, transfer, and planning, our ability to abstract from perceptual observations — which has not received sufficient attention relative to its importance in intelligence — urges us to take a closer look at how abstraction arises from sequential perceptual sequences. In the current work, we have proposed two specific sequence abstraction types: projectional motifs — patterns derived from sequences through a projectional function, and variable motifs — patterns that combine both concrete and variable elements. We studied the process of abstract motif learning in sequences, tested the learning and transfer of both motifs in a sequence recall paradigm, and proposed a model that abstracts sequences to compress sequence representations with projectional and variable motifs. We found that our model explained human behavior well.

Previously, associative learning models have been shown to explain human judgment of grammatical versus agrammatical strings in artificial grammar learning tasks[3,4,35,36]. Our model comparison between associative learning and motif learning suggests that associative learning alone is insufficient to explain human abstraction learning and transfer in sequence recall. As an alternative account of sequence learning, chunking models including PARSER[9], HCM[10], CCN and TRACX[11,12] acquire repeated patterns from sequences as chunks. Model comparison between the chunking model and motif learning model suggests that the chunking model captures a part of variable motif learning but not variable motif transfer, nor the learning and transfer of projectional motifs. Expanding the space of chunking from concrete sequences to abstract spaces is vital to capture the motif learning and transfer effects observed in our experiments. In experiment 1, during training, when memorizing sequences with projectional motifs, the chunking model does not align with our observation of human behavior because the model learns chunks on the surface value of the sequences. In comparison, the motif learning model learns chunks in the projectional space of the motifs. While both models learn chunks by a merging mechanism that combines preexisting memorized sequence sub-parts into novel chunks of memorized sequence sub-parts, this chunk-building efficiency correlates with the number of repetitions of the memorized sequence chunks. The motif learning model, having memorized chunks in the motif space, has more opportunities to hone in its memory thanks to the frequent repetition of sequences in the projectional motif space. This model comparison suggests that humans facing this task exert learning behavior that resembles memorization in the projectional motif space rather than memorization of the concrete sequence space.

The inflexibility of memorizing subsequences on the surface value further disadvantages the chunking model in this experiment's transfer phase. Although the chunking model might have learned to compress sequences in chunks in the training phase, the fact that the memory chunks lie in the concrete sequence spaces makes the model inflexible to transfer any learned chunks to the transfer sequences. In comparison, the motif learning model learns chunks in the projectional motif space, which is shared between training and transfer.

In the variable motif learning experiment, the chunking model explains better than the motif learning model on learning variable motifs during the training phase but not the transfer to new ones. This is possible because learning chunks on concrete sequences captures a part of the participant's learning behavior. However, during transfer, the concrete sequence chunks are too stiff to adapt to novel sequences sharing the same variables. The model comparison suggests that the ability of the variable motif learning model to recycle the variable as an entity to construct new chunks is critical to capture the transfer behavior for humans in experiment 2.

## Related Work

A range of cognitive tasks examine learning of surface example structure in text strings. In the artificial grammar learning paradigm, participants learn a subset of the grammatical sequences generated from finite state languages[1]. After observation, they are asked to discriminate grammatical versus ungrammatical (inconsistent with the finite state language) sequences in a test phase. It was observed that participants can generally identify grammatical sequences in the test phase with above-chance accuracy[2]. Previous modeling work suggests that learning the associative transition probabilities between items in the string can replicate participants' performance in the task[3,4,35,36]. Our model comparison between associative learning and motif learning suggests that associative learning alone cannot explain human abstraction learning and transfer in sequence recall.

On top of the first-order transition structure, past research also suggests that people learn explicit structures as frequently occurring fragments

in sequences. Literature suggests that the similarity between the test and training strings influences test judgment[5]. Specifically, test strings that contain overlapping string fragments with the training string are more likely to be judged as grammatical[6–8]. This phenomenon can be explained by chunking models such as PARSER[9], HCM[10], CCN and TRACX[11,12], which learn repeated patterns from sequences as chunks. Our analysis suggests that although the chunking model resembles participants' learning progression during variable motif learning, it fails to capture variable motif transfer or the learning and transfer of projectional motifs. Expanding the space of chunking from concrete sequences to abstract spaces is vital to capturing the motif learning and transfer effects observed in our experiments.

Other works that relate sequence learning and mental compression have used an outlier detection task[37]: participants detect violation upon hearing a binary auditory sequence. Past work has shown that a language-of-thought model's minimal description length of binary sequences relates to human psychological complexity[37,38]. A sequence recall task differs from an outlier detection task in that it directly probes human ability to recall the sequences to be memorized and, therefore, is not limited to testing human prediction of the subsequent element. Our work further relates mental compression with sequence motif learning. Rather than a static account of sequence complexity, the abstraction learning model proposes a discovery process of actively constructing sequence motifs during practice.

The learning of explicit rules bridges between literature and has also been considered in the field of category learning[39]. After presenting participants with observation instances of artificial objects coming from artificially defined category memberships, participants categorize novel objects as belonging to one category or the other. It was observed that both rules and statistics of the categories influence judgment, as atypical examples take longer to be categorized[40,41]. Most theories of rule-based category learning assume that rules and similarities operate on the level of explicit perceptual representations. Relating to our work, we suggest that regularities from observational examples can also be manifested on an even more abstract level, such as the variability structure or projectional space. Assuming that the rule-discovery process operates on projected representations, previous rule-based models are similar to our currently put-forward motif-learning model. If we assume that models originally thought to operate on perceptual representations can also operate on projected representations, different interpretations of the current results become possible. For example, if every presented sequence is stored in abstract space, motifs could also be considered as prototypical abstract sequences[42,43]. Similarly, motifs could also be considered as assemblies of similar multidimensional abstract exemplars[44]. Our results cannot contribute to the debate about people's strategy to learn categories and regularities. The motif learning model is compatible with any of these strategies. However, the current results show that people can transform and use representations in an abstract, projected space to detect regularities, over and above the algebraic rules put forward previously[13] (see Experiment 2).

On the level of learning non-explicit sequence patterns: previous work[13] showed that seven-month-old children could extract an abstract rule when exposed to sequences with simple grammar (e.g., ABA). After exposure, the infants were more likely to direct their gaze toward novel sequences sharing the same structure, such as KTK, rather than toward a different structure, such as DDF. Our experiment further examines the implication of learning projectional motifs in sequence memorization and recall.

The notion of learning variable motifs relates to the symbolic acquisition of language knowledge[14,15], endorsing the view that occurrence frequency cannot be the only basis of grammatical or syntactical language learning, as we can judge very unlikely-occurring sentences to be grammatical[17]. Language acquisition involves learning phase structure, such as a noun phrase usually consistent with a determinant followed by a noun[16]. suggests that abstract patterns on the level of symbols, such as nouns and verbs, operate to utter grammatically valid sentences without an enlisted preoccupied output. The acquisition and utterance of language structure involves the acquisition of operations on the symbolic level.

Previous work has postulated that similarities and rule knowledge are two ends of the same continuum and may have separated learning origins[45]. Moreover, abstraction learning tends to occur after learning the surface-level structure[46]. Perceptual and abstract properties can concurrently occur during the learning process[47]. Our model captures the process of bimodal learning by learning both the surface-level fragments and the deep-level structure and demonstrates its resemblance to human behavior in a sequence recall task that both associative and chunk learning fall short of explaining.

## Limitations

Our work has limitations. In Experiment 1, learning one motif facilitated participants' transfer to a different motif (3e). The same was not true for the model: learning one motif impaired its ability to transfer to the other different motif. The model's ability to recall a new motif is hindered when it has already learned one motif. This occurs because the recall process involves sampling subsequences acquired since the start of training, and the previously learned chunks from the training motif may still get sampled during the recall process which interferes with recall accuracy. This effect is consistent with the proactive interference effect in the literature that memory for previously presented lists impairs memory for later presented lists[48–51]. In contrast, in our experiment, it seems as if humans are establishing a fresh context for structure discovery when encountering a new motif which is not captured by the current model[52–54]. This phenomenon can be attributed to yet an additional layer of contextual abstraction that the model does not capture. Namely, training on sequences with motifs guides people to look for motifs in subsequent sequences. This observation that structural prior prime participants to search for structure in another form resonates with previous findings on structured multi-armed bandit tasks, where a learning-to-learn effect was observed[55]. Future work could extend the current modeling framework to accommodate the flexibility of transferring across motifs. For example, one option would be to introduce a mechanism that updates the prior about the probability of having underlying structure in the sequence. And consequentially, having trained on a motif helps a model to update the structural prior and infer an alternative structural form with a higher likelihood than no structures in the sequence.

In this work, we compared model fit via generative accuracy, which reflects the model's internal memory representation acquired from instruction sequences up to trial n, as it is evaluated on the recalled sequence generated by the model in comparison to the instruction sequence presented on that trial. This method provides one aspect of model fit. Future work could look at other aspects of behavioral-model comparison. One example could be to evaluate the likelihood of participants' recalled sequence given the models and compare the likelihood as a measure of model fit. Alternatively, the complexity of participant generated sequences as parsed by the models can be compared with reaction time data, as less complex sequences would be recalled faster.

Finally, most of our analysis compare model predictions with human behavior on an aggregated level. We encourage future investigations to examine participants' idiosyncratic learning and transfer strategies. Apart from that, our work defines and investigates two particular types of abstraction. We encourage future work to extend the investigation and look at more forms of abstraction or automatic ways of discovering abstraction such as hierarchical clustering and chunking on recursive abstract levels.

## Conclusion

A vital role of abstraction is to facilitate sequence compression and generalization, and we proposed a motif learning model based on this principle. Our model builds up a sequence memory via chunking motifs in an abstract space in search of a low-complexity sequence representation, facilitating memorization and transfer. We developed a sequence recall task to examine whether the two proposed motif types aid in learning and generalization. Our findings suggest that both motifs facilitate sequence memorization and generalization to novel, unseen sequences. Humans showed similar behavior to the model in learning and generalization of both abstraction types.

This suggests that sequence compression via abstraction is a plausible mechanism to explain human performance in sequence memory tasks. Our work paves the way for a better understanding of how people construct abstract representations from observational sequences for efficient compression and transfer.

## Data availability

The data collected is also available at: https://github.com/swu32/motif_learning.

## Code availability

The data collected and code used for analyzing this study can be found in this github repository: https://github.com/swu32/motif_learning.

Received: 15 January 2024; Accepted: 16 December 2024;
Published online: 07 January 2025

## References

1. Chomsky, N. & Miller, G. A. Finite state languages. **1**, 91–112.
2. Dulany, D. E., Carlson, R. A. & Dewey, G. I. A case of syntactical learning and judgment: How conscious and how abstract? *J. Exp. Psychol.: Gen.* **113**, 541–555 (1984).
3. Gomez, R. L. & Gerken, L. Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge. *Cognition* **70**, 109–135 (1999).
4. Gómez, R. L. Variability and detection of invariant structure. *Psycholog. Sci.* **13**, 431–436 (2002).
5. Brooks, L. Salience of item knowledge in learning artificial grammars. *J. Exp. Psychol.: Learn., Mem., Cognition* **18**, 328–344 (1992).
6. Perruchet, P. & Pacteau, C. Synthetic grammar learning: Implicit rule abstraction or explicit fragmentary knowledge? *J. Exp. Psychol.: Gen.* **119**, 264–275 (1990).
7. Knowlton, B., Squire, L. & Gluck, M. Probablistic classification learning in amnesia. *Learn. Mem. (Cold Spring Harb., N. Y.)* **1**, 106–20 (1994).
8. Knowlton, B. & Squire, L. Artificial grammar learning depends on implicit acquisition of abstract and exemplar-specific information. *J. Exp. Psychol. Learn., Mem., cognition* **22**, 169–81 (1996).
9. Perruchet, P. & Vinter, A. Parser: A model for word segmentation. *J. Mem. Lang.* **39**, 246–263 (1998).
10. Wu, S., Elteto, N., Dasgupta, I. & Schulz, E. Learning Structure from the Ground up—Hierarchical Representation Learning by Chunking. In Koyejo, S. et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 35, 36706–36721 (Curran Associates, Inc., 2022). https://proceedings.neurips.cc/paper_files/paper/2022/file/ee5bb72130c332c3d4bf8d231e617506-Paper-Conference.pdf.
11. Servan-Schreiber, E. & Anderson, J. Learning artificial grammars with competitive chunking. *J. Exp. Psychol.: Learn., Mem., Cognition* **16**, 592–608 (1990).
12. French, R. M., Addyman, C. & Mareschal, D. TRACX: A recognition-based connectionist framework for sequence segmentation and chunk extraction. *Psycholog. Rev.* **118**, 614–636 (2011).
13. Marcus, G. F., Vijayan, S., Bandi Rao, S. & Vishton, P. M. Rule learning by seven-month-old infants. *Sci. (NY)* **283**, 77–80 (1999).
14. Boole, G.*The Laws of Thought (1854)* (The Open court publishing company, London, 1854).
15. Marcus, G. F. The algebraic mind: Integrating connectionism and cognitive science (2001). https://api.semanticscholar.org/CorpusID:142639115.
16. Marcus, G. Children's overregularization of english plurals: A quantitative analysis. *J. child Lang.* **22**, 447–459 (1995). Funding Information: [*] I thank Steven Pinker, Fei Xu and two anonymous reviewers for comments on an earlier draft. This research was funded by an NDSE Graduate Fellowship to Marcus, NIH Grant HD 18381 to Steven Pinker (MIT), and grants from NIMH (training grant T32 MH18823) and the McDonnell-Pew Program in Cognitive Neuroscience to MIT's Department of Brain and Cognitive Sciences. Address for correspondence: Gary Marcus, Department of Psychology, Tobin Hall, University of Massachusetts, Amherst, MA 01003, USA. E-mail: marcus@psych.umass.edu.
17. Chomsky, N. *Aspects of the Theory of Syntax*. (MIT press, 2014).
18. Ebbinghaus, H.*Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie* (Duncker & Humblot Leipzig, 1885).
19. Lewandowsky, S. & Murdock Jr, B. B. Memory for serial order. *Psycholog. Rev.* **96**, 25–57 (1989).
20. Barr, D. J., Levy, R., Scheepers, C. & Tily, H. J. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *J. Mem. Lang.* **68**, 255–278 (2013).
21. Matuschek, H., Kliegl, R., Vasishth, S., Baayen, H. & Bates, D. Balancing type i error and power in linear mixed models. *J. Mem. Lang.* **94**, 305–315 (2017).
22. Ohlsson, S. & Lehtinen, E. Abstraction and the acquisition of complex ideas. *Int. J. Educ. Res.* **27**, 37–48 (1997).
23. Gentner, D. & Hoyos, C. Analogy and Abstraction. *Top. Cogn. Sci.* **9**, 672–693 (2017).
24. Lawler, J. M. Metaphors we live by. *Language* **59**, 201–207 (1983).
25. Hofstadter, D. R. Analogy as the core of cognition. *The analogical mind: Perspectives from cognitive science* 499–538 (2001).
26. Duncker, K. On problem-solving. *Psychol. Monogr.* **58**, i–113 (1945).
27. Giunchiglia, F. & Walsh, T. A theory of abstraction. *Artif. Intell.* **57**, 323–389 (1992).
28. Konidaris, G. On the necessity of abstraction. *Curr. Opin. Behav. Sci.* **29**, 1–7 (2019).
29. Abel, D., Hershkowitz, D. E. & Littman, M. L. Near Optimal Behavior via Approximate State Abstraction 9.
30. Eckstein, M. K. & Collins, A. G. E. Computational evidence for hierarchically structured reinforcement learning in humans. *Proc. Natl. Acad. Sci.* **117**, 29381–29389 (2020).
31. Jetchev, N., Lang, T. & Toussaint, M. Learning grounded relational symbols from continuous data for abstract reasoning (2013).
32. Luciw, M. & Schmidhuber, J. Low complexity proto-value function learning from sensory observations with incremental slow feature analysis. In *Proceedings of the 22nd International Conference on Artificial Neural Networks and Machine Learning - Volume Part II*, ICANN'12, 279–287 (Springer-Verlag, Berlin, Heidelberg, 2012). https://doi.org/10.1007/978-3-642-33266-1_35.
33. Silver, T. et al. Inventing relational state and action abstractions for effective and efficient bilevel planning. ArXiv abs/2203.09634 https://api.semanticscholar.org/CorpusID:247595182 (2022).
34. Chollet, F. On the measure of intelligence. ArXiv abs/1911.01547 https://api.semanticscholar.org/CorpusID:207870692 (2019).
35. Saffran, J. R., Newport, E. L. & Aslin, R. N. Word segmentation: The role of distributional cues. *J. Mem. Lang.* **35**, 606–621 (1996).
36. Saffran, J. R., Johnson, E. K., Aslin, R. N. & Newport, E. L. Statistical learning of tone sequences by human infants and adults. *Cognition* **70**, 27–52 (1999).
37. Planton, S. et al. A theory of memory for binary sequences: Evidence for a mental compression algorithm in humans. *PLOS Comput. Biol.* **17**, e1008598 (2021).
38. Dehaene, S., Al Roumi, F., Lakretz, Y., Planton, S. & Sablé-Meyer, M. Symbols and mental programs: a hypothesis about human singularity. *Trends Cogn. Sci.* **26**, 751–766 (2022).
39. Ashby, F. G. & Townsend, J. T. Varieties of perceptual independence. *Psycholog. Rev.* **93**, 154–179 (1986).
40. Allen, S. W. & Brooks, L. R. Specializing the operation of an explicit rule. *J. Exp. Psychol. Learn., Mem., Cognition* **120**, 3–19 (1991).
41. Rips, L. J.*Similarity, typicality, and categorization* (Cambridge University Press, 1989).

42. Homa, D., Sterling, S. & Trepel, L. Limitations of exemplar-based generalization and the abstraction of categorical information. *J. Exp. Psychol.: Hum. Learn. Mem.* **7**, 418 (1982).
43. Smith, J. D. & Minda, J. P. Prototypes in the mist: The early epochs of category learning. *J. Exp. Psychol.: Learn., Mem., Cognition* **24**, 1411 (1998).
44. Nosofsky, R. M. Attention, similarity, and the identification–categorization relationship. *J. Exp. Psychol.: Gen.* **115**, 39 (1986).
45. Pothos, E. M. The rules versus similarity distinction. *Behav. Brain Sci.* **28**, 1–14 (2005).
46. Goldwater, M. B., Don, H. J., Krusche, M. & Livesey, E. J. Relational discovery in category learning. *J. Exp. Psychol.: Gen.* **147**, 1–35 (2018).
47. Shanks, D. R. & John, M. F. S. Characteristics of dissociable human learning systems. *Behav. Brain Sci.* **17**, 367–395 (1994).
48. Wickens, D. D. Encoding categories of words: An empirical approach to meaning. *Psychological Rev.* **77**, 1–15 (1970).
49. Wickens, D. D., Born, D. G. & Allen, C. K. Proactive inhibition and item similarity in short-term memory. *J. Verbal Learn. Verbal Behav.* **2**, 440–445 (1963).
50. Watkins, O. C. & Watkins, M. J. Buildup of Proactive Inhibition as a Cue-Overload Effect.
51. Watkins, M. J. & Watkins, O. C. Cue-overload theory and the method of interpolated attributes. *Bull. Psychonomic Soc.* **7**, 289–291 (1976).
52. Dennis, S. & Humphreys, M. A context noise model of episodic word recognition. *Psycholog. Rev.* **108**, 452–78 (2001).
53. Farrell, S. Temporal clustering and sequencing in short-term memory and episodic memory. *Psycholog. Rev.* **119**, 223–271 (2012).
54. Brown, G., Neath, I. & Chater, N. A temporal ratio model of memory. *Psycholog. Rev.* **114**, 539–76 (2007).
55. Schulz, E., Franklin, N. T. & Gershman, S. J. Finding structure in multi-armed bandits. *Cogn. Psychol.* **119**, 101261 (2020).


## Acknowledgements

We thank Peter Dayan, Felix Wichmann, and Susanne Haridi for helpful discussions. This work was supported by the Max Planck Society. The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

## Author contributions

Conceptualization: Shuchen Wu, Mirko Thalmann, Eric Schulz. Formal analysis: Shuchen Wu, Mirko Thalmann. Software: Shuchen Wu. Visualization: Shuchen Wu. Writing - original draft: Shuchen Wu. Writing - review & editing: Shuchen Wu, Mirko Thalmann, Eric Schulz.

## Funding

Open Access funding enabled and organized by Projekt DEAL.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s44271-024-00180-8.

**Correspondence** and requests for materials should be addressed to Shuchen Wu.

**Peer review information** *Communications psychology* thanks Lorenzo Ciccione and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editors: Jennifer Bellingtier and Antonia Eisenkoeck. A peer review file is available.

**Reprints and permissions information** is available at http://www.nature.com/reprints

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2025

# Supplementary References

## 0.1 Experiment 1

### *0.1.1 Training*

**Regression Coefficient** Other regressors that showed significant effects are serial position, trial ID, chunk boundary, and the number of repetitions. Serial position is the n-th item recalled in a trial, significantly affecting recall correctness ($\chi^2(1) = 697.92$, $p < 0.001$). The further the position of a sequence recall, the more likely that participants will be making a mistake ($\beta = -0.29$, $se = 0.03$, $z = -11.60$, $p < 0.001$). This result is consistent with the primacy effect widely observed in the serial recall literature[1] as mistake probability increases with the serial position. Apart from that, trial ID, i.e., the number of practice trials ($\chi^2(1) = 810.02$, $p < 0.001$), also increases the log-odds of recalling correctly ($\beta = 0.08$, $se = 0.02$, $z = 4.83$, $p < 0.001$), confirming a practice effect over training blocks.

We also observed that the sub-sequence boundary (at the first, fourth, fifth, eighth, ninth, and twelfth item of the sequence) affects recall correctness ($\chi^2(1) = 34.767$,$p < 0.001$). Items located at the beginning and the end of the displayed sub-sequence are more likely to be recalled correctly compared to the items within each sub-sequence ($\beta = 0.15$, $se = 0.02$, $z = 6.64$, $p < 0.001$). This observation resonates with the literature suggesting participants have more accurate memory and recall performance at the boundaries of serially ordered sub-sequences than between[2–4]. Additionally, the number of exact repetitions ($\chi^2(1) = 158.19$,$p < 0.01$) increases the log odds of correct recall press ($\beta = 0.07$, $se = 0.01$, $z = 5.10$, $p < 0.001$).

### *0.1.2 Transfer*

**Regression Coefficient** Apart from transfer types, the recall keypress correctness decreases with the recall sequence position ($\chi^2(1) = 322.3$, $p < 0.001$). The further participants are into recall, the more likely they will make mistakes ($\beta = -0.08$, $\sigma = 0.008$, $z = -10.78$, $p < 0.001$). The decrease in recall accuracy is consistent with the primacy effect in memory literature: items that occur early in a sequence tend to be remembered and recalled more accurately[1]. We also observed a practice effect: trial ID affects the log odd ratio of pressing the right key ($\chi^2(1) = 86.07$,$p < 0.001$) ($\beta = 0.09$, $se = 0.01$, $z = 5.94$, $p < 0.001$). Apart from that, chunk boundary effect was also observed: the subchunk boundaries generally exhibit a higher recall accuracy than the interchunk items ($\chi^2(1) = 25.17$,$p < 0.001$) ($\beta = 0.17$, $\sigma = 0.03$, $z = 5.17$, $p < 0.001$), resonating with existing findings that chunk boundaries are remembered more accurately than within-chunk items[3].

## 0.2 Experiment 2

### *0.2.1 Training*

**Regression Coefficient** Other regressors that showed significant effects are serial position ($\beta = -0.68$, $se = 0.04$, $z = -16.37$, $p < 0.001$, 95% CI = -0.77 to -0.60), confirming the recency effect; Trial ID ($\beta = 0.64$, $se = 0.13$, $z = 5.04$, $p < 0.001$), confirming the practice effect; the number of repetitions ($\beta = 0.05$, $se = 0.01$, $z = 4.41$, $p < 0.001$); and chunk boundary ($\beta = 0.32$ $se = 0.02$, $z = 13.29$, $p < 0.001$).

### *0.2.2 Transfer*

**Regression Coefficient** Similar to the training block, we observed a recency effect ($\beta = -0.64$, $se = 0.02$, $z = -31.09$, $p < 2e-16$, 95% CI = -0.69 to -0.61), practice effect ($\beta = 0.32$, $se = 0.02$, $z = 13.30$, $p < 0.001$, 95% CI = 0.28 to 0.37), repetition effect ($\beta = 0.05$, $se = 0.03$, $z = 1.85$, $p = 0.06$, 95% CI = 0.00 to 0.12), and chunk boundary effect($\beta = 0.31$, $se = 0.04$, $z = 7.91$, $p < 0.001$, 95% CI = 0.24 to 0.39), confirming a viable expectation over experimental manipulation.

## 0.3 Reaction Time Analysis

As shown in Supplementary Figure S1 average reaction time for the three training groups decreases with practice, and reaction time converges at the end of the training block for all three groups. Supplementary Figure S1 b shows the reaction time to press the recall sequence within each recall trial. Shown in Supplementary Figure S1 c is the average reaction time across the three training groups. The average reaction time to recall the sequence does not differ significantly amongst the three groups, as indicated via fitting a linear mixed effect regression model onto participants' recall time, assuming a random intercept over individual participants and a random slope over serial positions ($\chi^2(2) = 4.32$, $p = .11$).

Other regressors that showed significant effects are serial position, trial ID, chunk boundary, and repetitions, as shown in Supplementary Figure S2. Serial position, the n-th item recalled in a trial, affects reaction time. The further the position of a sequence recall, the shorter the reaction time ($\beta = -126.76$, $se = 20.51$, $t = 103.30$, $p < 0.001$). Trial ID, i.e., the number of practice trials, also reduces reaction time ($\beta = -117.413$, $se = 5.86$, $t = -20$, $p < 0.001$), confirming a practice effect over the training phase. Immediate repetitions of the previous sequence also drives reaction time faster ($\beta = -61.86$, $se = 9.28$, $t = -6.67$, $p < 0.001$). Reaction time of the first item in each subsequence position is much higher than other serial positions in the sequence ($\beta = 573.54$, $se = 7.19$, $t = 79.73$, $p < 0.001$), reflecting the structure of the task.

Shown in Supplementary Figure S1 d is the average reaction time during the transfer phase: for the groups trained on motifs, transfer type affects their transfer performance ($\chi^2 = 174.05$, $p < 0.001$). When the motif groups transfer to the test blocks,

their reaction time to recall the sequence and execute the sequence presses is higher for transferring to the same motif compared to transferring to an independent block ($\hat{\beta} = -109.34$, $se = 9.49$, $t(30417) = -11.514$, $p =< 0.0001$). When the motif group transfers to a different motif, the reaction time speed up is not significantly higher than the transfer to an independent block ($\hat{\beta} = -2.22$, $se = 9.49$, $t(30417) = -0.23$, $p = 0.81$). Having trained on sequences with motifs, participants recall sequences faster when transferring to a sequence with the same motif but not necessarily to a different motif.

For experiment 2, we also fitted a linear mixed effect regression model onto participants' recall time, assuming a random intercept over individual participants and a random slope over trial ID. As shown in Supplementary Figure S3, regressors that showed significant effects during the training block are serial position, trial ID, and chunk boundary. Serial position, the n-th item recalled in a trial, affects reaction time. The further the position of a sequence recall, the shorter the reaction time ($\beta = -97.68$, $se = 4.09$, $t = -23.84$, $p < 0.001$). Trial ID, i.e., the number of practice trials, also reduces reaction time ($\beta = -134.271$, $se = 19.62$, $t = -6.84$, $p < 0.001$), confirming a practice effect over the training phase. Number of repetitions drives reaction time faster ($\beta = -7.15$, $se = 2.23$, $t = -3.20$, $p = 0.002$). Reaction time of the first item in each subsequence position is much higher than other serial positions in the sequence ($\beta = 653.003$, $se = 9.46$, $t = 69.02$, $p < 0.001$), reflecting the structure of the task.

During the transfer phase, a linear mixed effect regression on recall time, assuming a random intercept over individual participants and a random slope over trial ID and serial position shows serial position ($\beta = -140.66$, $se = 19.02$, $t = -7.39$, $p < 0.001$), and chunk boundary as affecting the reaction time ($\beta = 773.94$, $se = 18.95$, $t = 40.85$, $p < 0.001$).

## 0.4 Model Specification

**Algorithm 1** Motif Learning

**Require:** *seq*: learning sequences
**Require:** *cg*: dictionary of learned chunks
**Require:** *threshold_chunk*: boolean flag for learning chunks
**Require:** *abstraction*: boolean flag for learning variables

*chunk_record* $\leftarrow \{\}$ ▷ Initialize chunk record
$t \leftarrow 0$
**while** not *seq_over* **do**
    $current_chunks, cg, seq, chunk_record \leftarrow identify_latest_chunks(cg, seq)$
    $cg \leftarrow learning_and_update(current_chunk, chunk_record, cg, threshold_chunk = True)$
    **if** *abstraction* **then**
        $cg \leftarrow abstraction_update(current_chunks, cg)$
    **end if**
    $cg.forget()$ ▷ multiple all frequency record by $\theta$
**end while**
**return** $cg, chunk_record$

The model initiates with a dictionary $cg$, which holds chunks (sub-sequences) and the transition between chunks. When an instruction sequence is presented to the model, it consecutively parses the sequence via the chunks in the dictionary that contain the biggest size. At each parsing step, the model updates the frequencies of each parsed item and transition frequencies between the previously parsed item and the current one. After parsing a chunk, the boolean flag *thresholdchunk* and *abstraction* control the model to create new chunks or to learn new variables.

*thresholdchunk* is a boolean flag that indicates whether the algorithm will learn and combine new chunks based on the input sequence (True) or just parse the sequence with existing items in the dictionary (False). In case it is true, then the algorithm checks if the currently identified chunk and the previously identified chunk have been conjunctively activated more than a minimum threshold in the transition matrix ($N = 3$). On top of that, a hypothesis test ($\chi^2$) is conducted to assess whether the consecutively parsed chunks are correlated with significance level $p < 0.05$. If so, then a new chunk is created by combining the previous with the current and incorporating it into the chunking graph $cg$. This procedure also includes cases where the current chunk contains variables within.

*abstraction* is another boolean flag that controls the learning of variables. When this flag is on, the model constructs new variables from chunks that share common ancestors and common descendants, indicating these chunks share similar occurrence contexts. A new variable is created if it connects a set of chunks with a combined frequency above a threshold ($freq_T = 6$). At the end of each sequence parse, the algorithm performs a "forgetting" step, which multiplies all chunk occurrences and transition frequencies by $\theta = 0.996$.

**Abstraction Learning** When simulating learning projectional motifs, we turn on the *thresholdchunk* and learn sequences

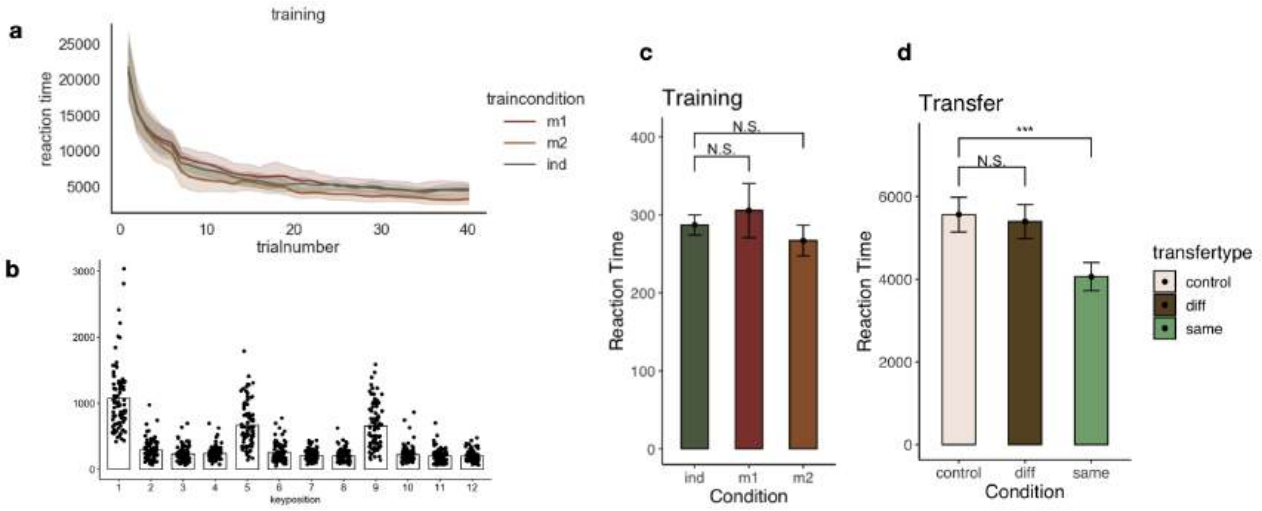


**Figure S1.** Reaction time analysis. a. Average reaction time across training trials. b. Average reaction time across recall sequence position. c. Average reaction time during the training block. d. Average reaction time during the transfer block across three transfer types. Same: Motif 1 – Motif 1 and Motif 2 – Motif 2; different: Motif 1 – Motif 2, and Motif 2 – Motif 1; control: Independent – Motif 1, and Independent – Motif 2.

on the projectional motif space. When simulating learning variable motifs, we set both the *learn* and *abstraction* flag to be true.

**Chunking** When simulating the chunking model, we turn on the *thresholdchunk* flag and turn off the *abstraction* flag.

**Associative Learning** When simulating the associative learning model, we turn off both the *thresholdchunk* flag and the *abstraction* flag. Thereby, no new chunks are created, and the model will learn the transition and occurrence frequencies of the atomic sequential elements.

**Recall** The recall function simulates the process of sequential recall from a chunk graph, starting with a primed item and proceeding through associative transitions. Given a priming first item of the sequence, the model samples a chunk consistent with the primed first item. Subsequent chunks are sampled based on transition probabilities from the previously recalled chunk (prev). The process repeats until the length of the recalled sequence reaches the desired sequence length $seql = 12$.

## Supplementary References


1. Oberauer, K. Understanding serial position curves in short-term recognition and recall. *J. Mem. Lang.* **49**, 469–483, DOI: https://doi.org/10.1016/S0749-596X(03)00080-9 (2003).
2. Lewandowsky, S. & Jr, B. Memory for serial order. *Psychol. Rev.* **96**, 25–57, DOI: 10.1037/0033-295X.96.1.25 (1989).
3. Farrell, S. Temporal clustering and sequencing in short-term memory and episodic memory. *Psychol. Rev.* **119**, 223–271, DOI: 10.1037/a0027371 (2012).
4. Cowan, N., Saults, J., Elliott, E. M. & Moreno, M. V. Deconfounding Serial Recall. *J. Mem. Lang.* **46**, 153–177, DOI: 10.1006/jmla.2001.2805 (2002).

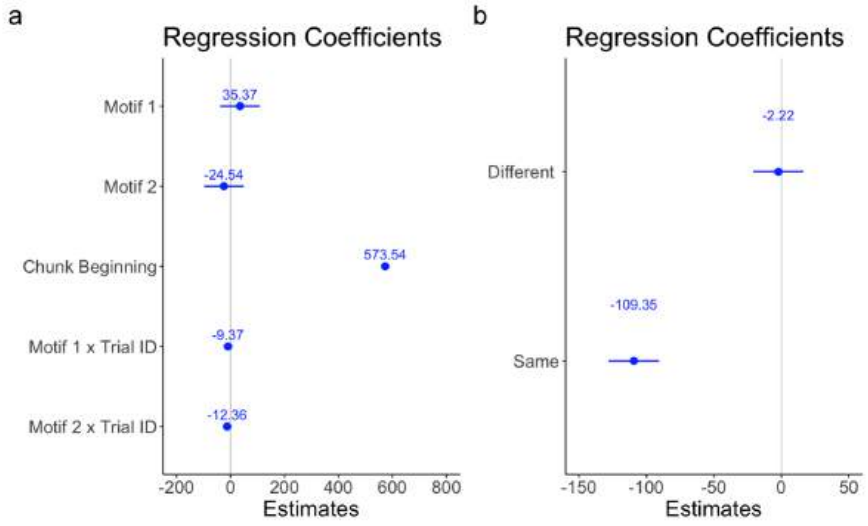


**Figure S2.** Reaction time analysis. a. regression coefficient of experiment 1 during training. b. transfer

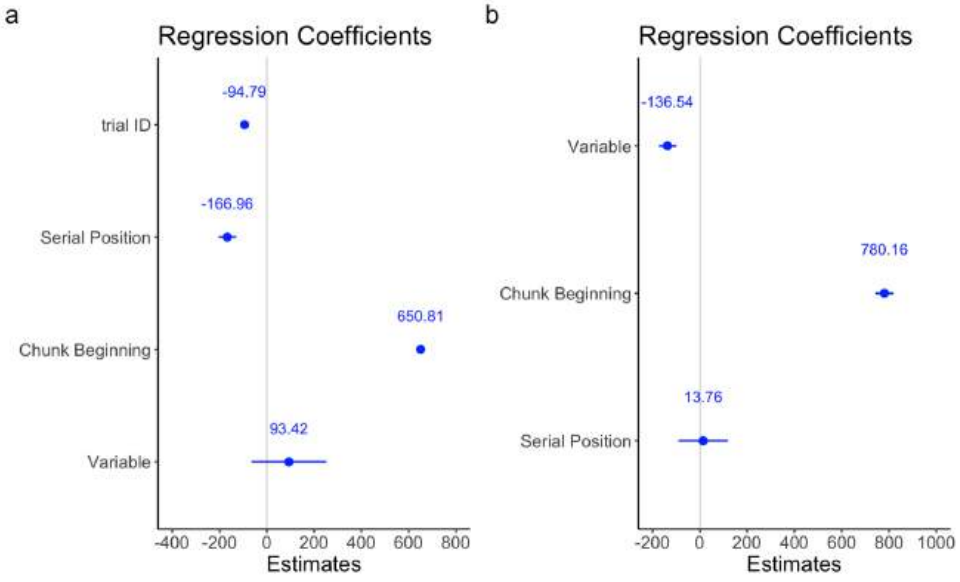


**Figure S3.** Reaction time analysis. a. regression coefficient of experiment 2 during training. b. transfer

# Building, Reusing, and Generalizing Abstract Representations from Concrete Sequences

# BUILDING, REUSING, AND GENERALIZING ABSTRACT REPRESENTATIONS FROM CONCRETE SEQUENCES

**Shuchen Wu**
Helmholtz Munich
Max Planck Institute for Biological Cybernetics
shuchen.wu@tuebingen.mpg.de

**Mirko Thalmann**
Institute for Human-Centered AI
Helmholtz Munich
mirko.thalmann@helmholtz-munich.de

**Peter Dayan**
Department of Computational Neuroscience
Max Planck Institute for Biological Cybernetics
dayan@tuebingen.mpg.de

**Zeynep Akata**
Helmholtz Munich
Technical University of Munich
zeynep.akata@helmholtz-munich.de

**Eric Schulz**
Institute for Human-Centered AI
Helmholtz Munich
eric.schulz@helmholtz-munich.de

## ABSTRACT

Humans excel at learning abstract patterns across different sequences, filtering out irrelevant details, and transferring these generalized concepts to new sequences. In contrast, many sequence learning models lack the ability to abstract, which leads to memory inefficiency and poor transfer. We introduce a non-parametric hierarchical variable learning model (HVM) that learns chunks from sequences and abstracts contextually similar chunks as variables. HVM efficiently organizes memory while uncovering abstractions, leading to compact sequence representations. When learning on language datasets such as babyLM, HVM learns a more efficient dictionary than standard compression algorithms such as Lempel-Ziv. In a sequence recall task requiring the acquisition and transfer of variables embedded in sequences, we demonstrate HVM's sequence likelihood correlates with human recall times. In contrast, large language models (LLMs) struggle to transfer abstract variables as effectively as humans. From HVM's adjustable layer of abstraction, we demonstrate that the model realizes a precise trade-off between compression and generalization. Our work offers a cognitive model that captures the learning and transfer of abstract representations in human cognition and differentiates itself from the behavior of large language models.

## 1 INTRODUCTION

Abstraction plays a key role in intelligence (Konidaris, 2019). Philosophers traditionally view abstract ideas as formed by identifying commonalities across experiences, distilled from concrete impressions grounded in perception (Kant, 1998; Fichte, 2005; STERN, 1977). Psychologists suggest that abstraction arises from personal experiences, such as forming the concept of "whiteness" by observing various white objects (Yee, 2019; Barsalou, 1999). Abstract concepts are thought to build on concrete concepts and on top of the previously learned abstractions, thereby varying in complexity (Cuccio & Gallese, 2018; Van Oers, 2001; Collins & Quillian, 1969; Piaget, 1964). The ability to abstract, which is often seen as a human-specific trait, enables reasoning, generalization, and problem-solving in novel contexts (Ohlsson & Lehtinen, 1997; Dehaene et al., 2022; Duncker, 1945).

We hypothesize that the world contains patterns across scales of time and abstraction. Intelligent agents-facing sequences with nested hierarchical structures- need to model these structures to store, process, and interact in such environments. As a rational strategy, intelligent agents shall characterize

the temporal structure via chunking and characterize the abstract structure via identifying common features and grouping collections of different items that play similar roles. To explore these operations, we design a generative model that produces sequences nested with hierarchical structure and propose an approximate recognition model that conjunctively learns to chunk and abstract.

We go beyond previous proposals that introduce chunking as a mechanism for learning and composing complex structures from elementary perceptual units (Gobet et al., 2001; Miller, 1956; Wu et al., 2022) and propose a cognitive model that combines chunking and abstraction in one single system. The model uses abstraction in two key ways: first, by identifying shared features to facilitate efficient pattern retrieval, such as recognizing patterns from learned animals when fur is observed; and second, by categorizing different sequential items that appear in the same context, much like a variable in a computer program— for example, using the category "animal" to represent a cat, dog, or squirrel. These paired mechanisms enable the model to parse sequences into chunks and form abstract patterns based on both concrete and previously learned abstractions. Abstraction allows for memory-efficient, compact representations and reveals recurring patterns at the level of abstract categories. This allows the model to discover increasingly complex or abstract patterns, layer by layer, with controllably varying degrees of detail.

We first demonstrate the benefits of abstraction in memory efficiency and sequence parsing by comparing our algorithm with previous chunking models and other dictionary-based compression methods. Then, we show that the model exhibits human-like signatures of abstraction in a memory experiment requiring the transfer of abstract concepts. In the same experiment, we contrast the model's generalization behavior with large language models (LLMs), which have demonstrated rudimentary reasoning abilities (Wei et al., 2022) in recent years. We show that LLMs do not transfer abstract variables and rely more on associative learning than abstraction. We demonstrate the connection between abstraction level and abstract concept transfer by varying the level of abstraction as a parameter in the model. Our work offers a cognitive model that captures the learning and transfer of abstract representations in human cognition and differentiates itself from the behavior of artificial agents.

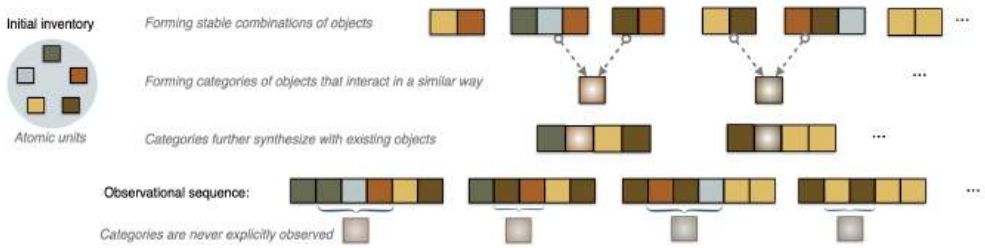


Figure 1: Generative Model. The inventory is initialized with a set of atomic units. These units randomly combine into objects. Some objects are randomly selected to become categories. Categories of existing objects have similar interaction properties. The dashed arrow points from those objects to their corresponding categories. These categories further combine with existing objects. An observational sequence is composed of objects randomly sampled from the existing inventory. Categories are latent and never explicitly observed but are manifested in one of the denoting objects.

## 2 Generating sequences with objects that contain hierarchical abstract structures

We design a probabilistic generative model that generates sequences by sampling objects from an inventory, with these objects organized hierarchically, resembling the structure found in natural sequences—such as molecules composed of chemical elements or the hierarchical nature of language (Abler, 1989; Sportiche et al., 2013; von Humboldt & Losonsky, 1999).

The generative model creates an inventory of recurring objects over $d$ iterations of expansion. As illustrated in Figure 1, the inventory starts with a set of atomic units $\mathbb{A}$. On each iteration, a novel object or category is created equiprobably. A new category (graded node) is created by pointing to a random selection of objects from the inventory up to the moment (these objects are treated

disjunctively). A new object is created by concatenating a random selection of pre-existing objects or categories from the existing inventory. After the inventory has been expanded up to the $d$-th iteration, the independent occurrence probability in the sequence is sampled from a flat Dirichlet distribution $f(p(c_1), ..., p(c_{|\mathbb{A}|}); \alpha_1, ..., \alpha_{|\mathbb{A}|}) = \frac{1}{\mathbf{B(a)}} \prod_{i=1}^{|\mathbb{A}|} P(c_i)^{\alpha_i - 1}$, $\alpha_i = 1 \forall i$ and assigned to each object in the inventory. Similarly, a probability is sampled from a flat Dirichlet to assign the occurrence probability of the objects within each category. More details can be found in the Appendix A.2.

To create an observational sequence, objects are randomly drawn from the inventory one after another with the assigned probability until they reach the desired sequence length. If the sampled object contains an embedded category, one of the objects corresponding to the category is sampled. This process is repeated recursively until all categories are replaced by specific objects, resulting in a sequence of discrete atomic units.

Intuitively, objects represent recurring observations of specific entities, such as molecules composed of chemical elements. Categories, on the other hand, represent broader categories of entities that share common properties, such as chemical elements belonging to the same class (e.g., noble gases or alkali metals). The observation sequence forms a nested hierarchy, where entities and their categories frequently interact and combine with others.

## 3 Learning abstractions from sequences

We ask what computational principles could help an agent discover objects and categories from such observational sequences without supervision. We propose that two mechanisms suggested by the cognitive literature are vital: chunk proposal and variable discovery. Chunking concatenates learned objects and forms new ones; variable discovery groups chunks with similar interaction properties into a category. We build on top of the hierarchical chunking model (HCM) (Wu et al., 2022), which learns a belief set $\mathbb{B}$ of a chunk dictionary $\mathbb{C}$ from discrete sequences. We expand the model so that it can also learn variables while improving the memory efficiency of the model. By identifying stably recurring entities as chunks and grouping similar entities into categories as variables, the agent can learn a structured inventory of identifiable patterns and use these patterns as entities to parse the sequence, leading to a more compressed factorization of perceptual sequences.

HVM learns a belief set $\mathbb{B}$ that contains both a dictionary of chunks $\mathbb{C}$ and variables $\mathbb{V}$. The variables are proposed as abstract entities based on the transition and marginal counts. As shown in Figure 2a, each variable $v \in \mathbb{V}$ denotes a set of chunks $E(v) = \{c_j\}$, $c_j \in \mathbb{C}$. The model also learns the probability of each chunk that a variable denotes $\forall v \in \mathbb{V}$, $\sum_j P(v \to c_j) = 1$, $c_j \in \mathbb{C}$.

**Parsing the sequence one chunk at a time** A fundamental feature of the model is to parse sequences in chunks as the basic cognitive units (Miller, 1956). Along with each parse $t$, the transition counts between the previous $c_L$ and next chunks $c_R$ are recursively updated: $T_{ij}(t+1) = T_{ij}(t) + [i = c_L][j = c_R]$ ($[\cdot] = 1$ if the argument is true and 0 otherwise) and along with the identification frequency of each parsed chunk $M_i(t+1) = M_i(t) + [i = c]$. When modeling human behavior, each entry of $M$ and $T$ multiplies with a memory decay parameter $\theta$ per parsing step. The probability of observing a sequence of parsed chunks $c_1, c_2, ..., c_N$ the becomes $P(c_1, c_2, ..., c_N) = \prod_{c_i \in \mathbb{C}} P_{\mathbb{C}}(c_i)$.

To parse a sequence, HCM iteratively chooses the biggest chunk amongst its learned dictionary $\mathbb{C}$ consistent with the upcoming sequence. The end of a previous parse initiates the next parse. As the dictionary size $|\mathbb{C}|$ increases, searching for the biggest consistent chunk becomes computationally expensive (Schreiber et al., 2023). In HVM, we introduce one notion of abstraction as finding *commonalities amongst memory items* to organize memory structure and speed up chunk retrieval during parsing. Memory items in the learned dictionary are organized into a hierarchical parsing graph that connects chunks with their common prefixes. Hence, all children chunks are different except for sharing a common prefix from the parent. The parsing graph arranges the chunks in $\mathbb{C}$ into a prefix Trie structure (Figure 2b), reflecting the cue-based, content-addressable nature of memory retrieval (Cunnings & Sturt, 2018; Dotlačil, 2021; Anderson, 1974). This design reduces search time to retrieve a chunk as arranged in the parsing graph, commonly used in predictive text or auto-complete dictionaries to speed up search steps (Fredkin, 1960). At every parsing step, HVM identifies the deepest chunk in the parsing graph that is consistent with the upcoming sequence. The end of the previous parse initiates the next parse. The search process would take the time complexity

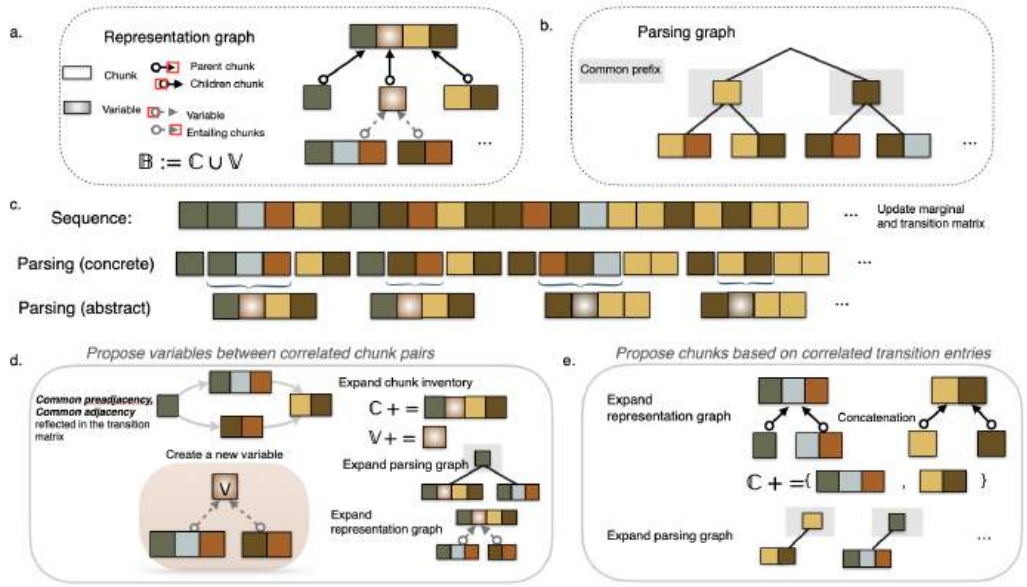


Figure 2: HVM builds up a representation graph and a parsing graph. a. A representation graph contains the learned chunks (nodes with solid colors) $\mathbb{C}$ and the contained variables (nodes with gradient colors) $\mathbb{V}$. Black arrows denote concatenation. Gray and dashed arrows point from chunks of the same category to the variable node that denotes this category. b. Abstraction organizes memory in a parsing graph. Nodes shaded by gray are abstract chunks being the common prefix intersection of its children. During parsing, to allocate to the matching chunk in $\mathbb{C}$, the model starts searching from the root of the parsing tree and traverses to the deepest node consistent with the upcoming sequence content. c. Upon completing each sequence parse, HVM updates counts on chunk frequencies and transitions and proposes inventory expansions, enabling a layer-wise discovery of recurring patterns per iteration, from specific to the more abstract. d. HVM proposes variables amongst correlated consecutive chunk pairs. e. Chunk proposal.

of $O(D)$, scaling with the depth $D$ of the tree compared to HCM time complexity of $O(|\mathbb{C}|)$ scaling with the size of $\mathbb{C}$. The appendix shows a guarantee to reduce the number of parsing steps for chunk retrieval A.4.

**Learning chunks** From parsed sequences, the model estimates the occurrence probability of chunk via the entries of $M$ as $P_{\mathbb{C}}(c_i) = \frac{M_i}{\sum_{c_j \in \mathbb{C}} M_j}$. The model starts with the null hypothesis $\mathcal{H}_0$ that all consecutively parsed chunks $c_L$ and $c_R$ are statistically independent $P(c_L, c_R) = P(c_L)P(c_R)$. If a significant correlation is found between consecutively parsed chunk pairs (with $p = 0.05$), the null hypothesis is rejected, and the pair is concatenated into a new chunk, $c_L \oplus c_R$, which is then added to the dictionary $\mathbb{C}$ as a new entry. Upon creating each new chunk, an edge from the left parent chunk (sharing the same prefix) connects to the newly created chunk in the parsing graph. When no parent chunk can be found, an ancestor node is created in the parsing graph, usually the sequence's atomic units. We prove in the SI section A.3 that under restricted conditions in which the sub-objects constituting the object in the ground truth are parsed faithfully by the model as sub-chunks, then these sub-chunks will be eventually proposed to concatenate into one chunk.

**Learning variables** A variable denotes distinct observations appearing in the same context (here defined as distinct chunks sharing preceding and succeeding chunks). Given consistent parsing of subchunks as object combination components in the generative model, the true chunks that belong to the same category will necessarily appear in the preadjacency and postadjacency entries (more explanation in SI A.3). HVM proposes variables amongst the significantly correlated chunks ($p \leq 0.05$) in the transition matrix consistent with this structure. For example, consider the case that the sequence includes A-ABC, A-DC, and ABC-ED, DC-ED. This suggests that ABC and DC play similar roles relative to A (coming after) and ED (before). Therefore, ABC and ED can be proposed as a new variable V to represent the abstract property that captures the distinct entities. V denotes ABC and DC, and the model identifies V if either ABC or DC is identified during parsing, as illustrated in Figure 2d and Figure 7 in SI.

Amongst the correlated consecutive chunk pairs, the model identifies all eligible transitions by intersecting the post-adjacency columns with the pre-adjacency rows, proposing a set of chunks $E(v) = c_1, c_2, ..., c_j$ represented by a new variable $v$. This variable proposal is accepted if the set of chunks that satisfy the condition $T_{min} \leq |E(v)| \leq T_{max}$ and $\sum_{c_i} M(c_i) \geq freq_T$. A variable denotes a set of chunks $E(v)$ and is identified if any of the chunks that it denotes is parsed (Figure 2c). Together, the common preceding chunk, in conjunction with the newly proposed variable, followed by the common succeeding chunk, is concatenated and proposed as a novel chunk $c_L \oplus v \oplus c_R$ to add to the inventory $\mathbb{C}$. An edge that connects $c_L$ to the new chunk $c_L \oplus v \oplus c_R$ to be added to the parsing graph, as illustrated in Figure 2d. This process introduces chunks with embedded variables and takes the computational complexity of $O(PQ)$, where $P$ and $Q$ are the numbers of unique entries of the adjacency and postadjaency chunks amongst the correlated consecutive chunk pairs. Two variables are merged into one if they share the same preceding and succeeding chunks. During parsing, a chunk is consistent with the sequence if any of its included variables contain a denoting chunk that is consistent with the sequence. Upon identifying a chunk during parsing, the marginal and transition count for both the chunk and the immediate variables that the chunk belongs to is incremented.

**Representation cleaning** Upon completion of each sequence parsing, the model uses correlations to propose new variables and chunks. It then uses the expanded inventory for the next parse. Unused variables and chunks are deleted upon the completion of each parsing iteration.

## 4 RESULTS

### 4.1 MODEL EVALUATION

HVM is an approximate inverse of the generative model, as in practice, learning the ground truth hierarchical patterns that generate observed data is a nonidentifiable problem (Post, 1946; Greibach, 1968). Therefore, we use a set of measures for model evaluation, focusing on parsing search steps, sequence length, sequence negative log-likelihood, and encoding efficiency. We trained models on sequences generated by the hierarchical generative model until convergence. For each iteration, the models parse the entire sequence using the dictionary updated from the previous iteration and propose

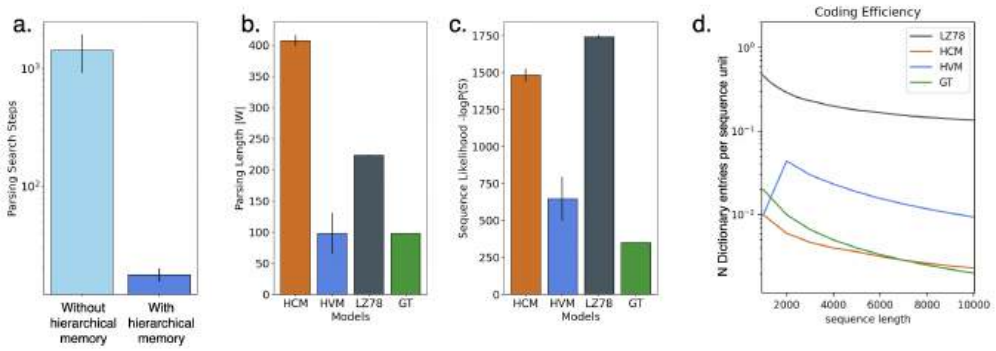


Figure 3: a. The effect of hierarchical memory structure on parsing search steps. b. Comparison between models in terms of sequence length after parsing. GT denotes the ground truth sequence length by the generative model. c. Model comparison based on sequence likelihood. d. Model comparison based on coding efficiency. Example model comparison with sequence length $|S| = 1000$, nested hierarchy depth $d = 30$, atomic set of size $|A| = 10$.

novel chunks/abstractions upon completing the parse. While HVM learns both chunks and variables per iteration, HCM learns only chunks and no variables. Previous research has related human memory in specific tasks to lossy compression (Nassar et al., 2018), with abstraction playing a crucial role in successful learning and transfer in sequence memory recall tasks. For reference, we also compared the models with LZ78, an off-the-shelf compression algorithm underlying compression schemes such as GIF and PNG, which also parses sequences into chunks and builds a dictionary to compress sequence data (Ziv & Lempel, 1978). Comparisons are shown in Figure 2.

**Organizing chunks in a parsing graph based on common prefix reduces parsing search steps** We compared HVM with HCM, which does not organize memory into a hierarchical parsing graph. Figure 3 a. shows that organizing chunks in a parsing graph dramatically reduces the average number of parsing search steps, i.e., the number of search steps needed for the model to locate the biggest chunk per parse. This search step now scales with the depth of the prefix tree compared to the size of the inventory in HCM. More discussions can be found in A.4.

**Discovering bigger recurring patterns** We compared HVM with its counterpart, HCM, which does not learn variables at the end of the learning iteration. Figure 3 b compares sequence length $|W|$ after parsing. HVM transforms the sequence $|S|$ into a code with a much smaller size $|W|$ compared to HCM and LZ78 and is on par with the ground truth (GT). Learning variables that denote distinct chunks occurring in similar contexts helps HVM to learn bigger patterns underlying the sequence on the description level of these variables.

**Parsing sequences with higher likelihood** Figure 3 c compares the negative log likelihood $(-\log P(S) = \prod_{c_i \in \mathbb{C}} P(c_i))$ of the parsed sequence upon convergence. HVM parses sequences more efficiently, as variables recur more frequently in sequences than their representing chunks, and therefore, the model that learns variables parses the sequence with higher likelihood.

**Encoding more efficiently** In many applications, the size of the dictionary affects compression efficiency (Navarro & Mäkinen, 2007; Ferragina & Manzini, 2005). We compare the encoding efficiency, i.e., the ratio between the dictionary size (number of entries) and the original sequence length $S$. Figure 3d shows the relation between the ratio and the sequence length. To encode a sequence of the same length, LZ78 creates a bigger dictionary than HVM and HCM. Models that harness the embedded hierarchical structure in sequences encode sequences more efficiently and learn a smaller dictionary to encode the same sequence length.

### 4.2 Learning from real-world sequential data

Going beyond artificially generated sequences, we evaluate HVM, HCM, and LZ78 in text domains from the BabyLM language dataset, which contain text snippets from a collection of data domains Warstadt et al. (2023). For each data domain, we took random snippets of 1000 characters and

| Data Domain | Model | Compression Ratio | NLL | Coding Efficiency |
|---|---|---|---|---|
| CHILDES (MacWhinney, 2000) | LZ78 | 0.38 | 2837.50 | 0.34 |
| | HCM | 0.51 | 2783.71 | 0.05 |
| | HVM | 0.36 | 1953.01 | 0.06 |
| BNC (BNC Consortium, 2007) | LZ78 | 0.39 | 3136.67 | 0.37 |
| | HCM | 0.65 | 3591.60 | 0.06 |
| | HVM | 0.50 | 3108.33 | 0.08 |
| Gutenberg (Gerlach & Font-Clos, 2020) | LZ78 | 0.39 | 3156.61 | 0.37 |
| | HCM | 0.69 | 3770.36 | 0.06 |
| | HVM | 0.54 | 3252.84 | 0.12 |
| Open Subtitles (Lison & Tiedemann, 2016) | LZ78 | 0.41 | 3395.09 | 0.39 |
| | HCM | 0.74 | 4151.89 | 0.07 |
| | HVM | 0.63 | 3764.48 | 0.07 |

Table 1: Model comparison on alternative sequences. NLL stands for negative log-likelihood.

calculated evaluation metrics, including compression ratio (the number of tokens before compression divided by the number of tokens after compression), sequence complexity (negative log-likelihood), and compression efficiency (the length of the compressed sequence divided by the number of dictionary entries).

As shown in table 1, LZ78 performs well in terms of compression efficiency, which is expected given its design purpose. However, the HVM model exhibits notable advantages when considering alternative evaluation metrics. Specifically, for text across the four data domains analyzed, cognitive models that incorporate the hierarchical structure of data outperform traditional compression methods in terms of encoding efficiency.This is because LZ78, which does not make strong assumptions about sequence structure, tends to create redundant entries in its dictionary. While this redundancy reduces the sequence length after compression, it introduces many infrequently used entries. In some data domains, HVM outperforms LZ78 in compression ratio and negative log-likelihood as well. Amongst all domains, HVM compresses the sequence further and parses the sequence with lower complexity than HCM.

### 4.3 HVM RESEMBLES HUMAN SEQUENCE MEMORIZATION AND TRANSFER

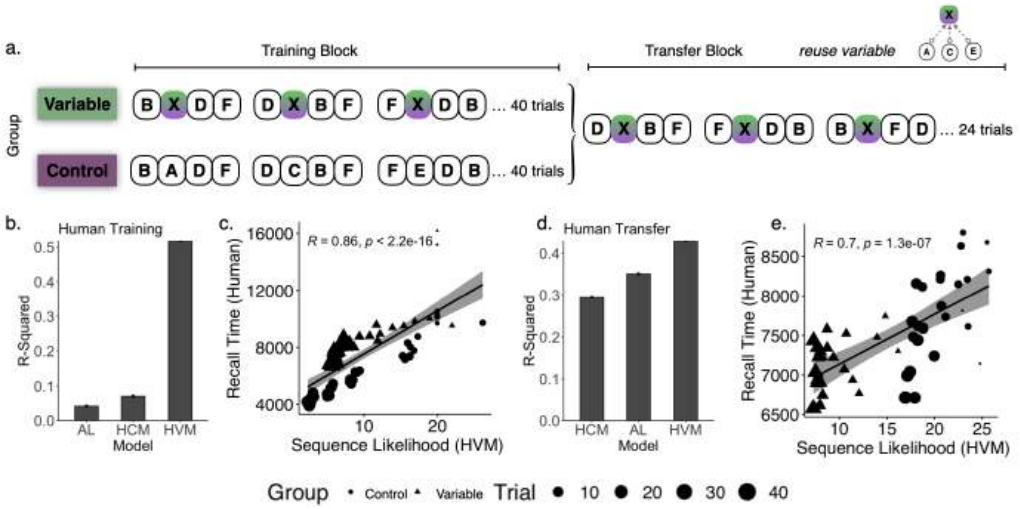


Figure 4: a. A sequence memory task that demands transferring a variable x from the training to the test block. b., c., d., e., Complexity (negative log-likelihood) progression of HVM correlates with human memory recall time in the training (b., c.) and the transfer block (d., e.). Memory decay parameter $\theta = 0.996$. Marker size increases with trial number.

We take the relation between transfer to learning abstractions to the setting of a memory experiment that demands learning and transferring variables (Wu et al., 2023). We ask whether HVM resembles aspects of human learning while enjoying the advantage of learning an interpretable dictionary.

In the experiment, 112 participants were instructed to recall a sequence of presented colors from memory (Figure 4 a). Participants were randomly assigned to a variable group and a control group. During the training block (40 trials), the control group was instructed to remember a fixed sequence BADF DCBF FEDB for each trial (each letter denotes a distinct color). In contrast, the variable group was instructed to remember sequences overlapping in 9 of the 12 colors: BXDF DXBF FXDB. A variable X, however, appeared at serial positions 2, 6, and 10, which could be occupied with the letters A, C, or E with equal probability for each trial. After the training block, both groups were then tested on the transfer block. The transfer sequence overlapped with the training sequence only at the variable positions: DXBF FXDB BXFD. The time participants took to recall the entire sequence for each trial was recorded. Previous studies suggest human recall time relates logarithmically to the perceptual predictability of the sequence (Carpenter & Williams, 1995; Anderson & Milson, 1989; Smith & Levy, 2013; Elman, 1990). We compare human recall time to the model's negative log-probability (likelihood) evaluated on the instruction sequence given.

To simulate the sequence likelihood for trial $n$, we used the instruction sequences up to trial $n-1$. This allowed the HVM to learn in an online manner, i.e. abstractions and chunks were proposed upon each parsing completion. For comparison purposes, we also simulated HCM with the same parameter setting, and an associative learning model (AL) that updates first order transitions between atomic sequential units. The sequence likelihood was evaluated as $-\log P(S) = -\log P(c_1)\prod_{i=2,...,n} P(c_i|c_{i-1})$. $c_1, c_2, ..., c_n$ are the units used to parse the sequence (chunks for HCM and HVM and atomic units for the associative learning (AL) model).

Figure 4 shows the relation between model sequence likelihood and human recall time during the training (b, c) and the transfer block (d, e). The size of the dot represents the trial number, and the shape of the dot represents the group (the triangle being the variable group, the circle is the control group). Figure 4b and d show the R-squared goodness-of-fit regressing the various models' sequence likelihoods onto human subjects' sequence recall times. The sequence negative log-likelihood from the HVM correlates with human recall time during the training ($R = 0.86, p \leq 0.001$) and transfer blocks ($R = 0.7, p \leq 0.001$). The latter suggests that HVM can generate behavior that resembles that of human knowledge transfer in the memorization of novel sequences with embedded variables.

### 4.4 Comparing abstraction learning and transfer in LLMs

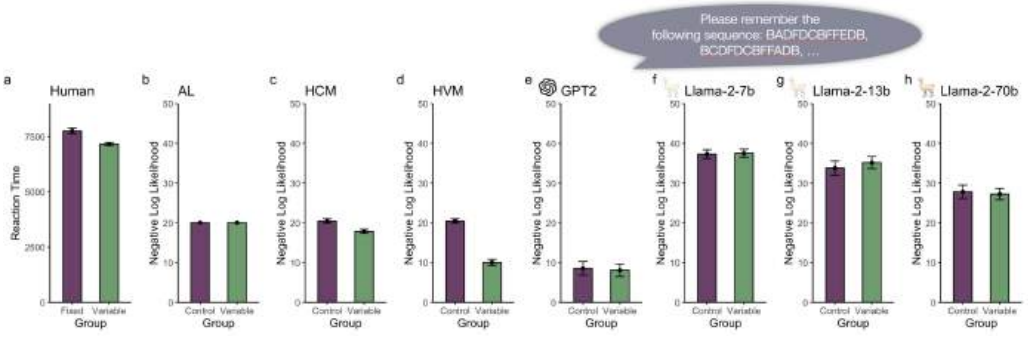


Figure 5: Comparison between human, variations of cognitive models, and AI models on the transfer block of the memory recall experiment. Bar plot shows the average human sequence recall time, and the sequence negative log-likelihood evaluated by the various cognitive and large language models.

Given that large language models (LLMs) have been praised for their emergent abilities (Wei et al., 2022), yet also compared to "lossy compression" (Chiang, 2023) and struggle with tasks that demand abstract thought (Fleuret et al., 2011; Odouard & Mitchell, 2022), we want to investigate whether models with or without abstraction better describe them. Specifically, we ask whether LLMs abstract similarly to humans and how their behavior compares to cognitive models incorporating abstraction. To explore this question, we conduct the same sequence recall experiment on LLMs.

To do that, we adapt the instructions from the human experiment into prompts for in-context learning in LLMs. The LLMs are tasked with predicting the next subsequent tokens based on the context of previously seen instruction sequences. The specific prompt is taken from the instruction given to the human participants: "Please remember the following sequence: BADFDCBFFEDB, ..., BCDFDCBFFADB." We then calculate the conditional probability of the next token, $z_i$, given the instruction sequence history up until the current token $P(z_i|prompt)$. After obtaining this probability, the subsequent token in the instruction is added to the prompt, and the LLM's conditional probability for the following token is updated accordingly. Using this prompt-chaining method, we can calculate LLM's negative log-likelihood for the next tokenized instruction sequence: $-\log P(S) = -\log P(z_1|prompt) \prod_{i=2,\dots,n} P(z_i|prompt_{i-1})$, analogous to the cognitive models. We apply this approach to evaluate the token prediction likelihood in four large language models: GPT-2 (Radford et al., 2019) and three variations of the Llama 2 model (Touvron et al., 2023)—Llama with 7 billion parameters (Llama 7B), 13 billion parameters (Llama 13B), and 70 billion parameters (Llama 70B).

Shown in Figure 5 are the transfer behavior across humans, LLMs, and various cognitive models for comparison. Supplementary section A.11 also shows the R-square value regressing model negative log-likelihoods on human recall time. For humans, the group trained on sequences with embedded variables recalls transfer sequences faster than the group trained on control sequences. Relating sequence recall time with sequence negative log-likelihood, the cognitive models HCM and HVM, on average, exhibit lower sequence negative log-likelihood after learning from a training block that shares variables with the transfer block. In contrast, the associative learning model and all variants of large language models do not differentiate between training sequences that share variables with the transfer block and those from a training block without transferrable variables.

### 4.5 Abstraction, distortion, generalization

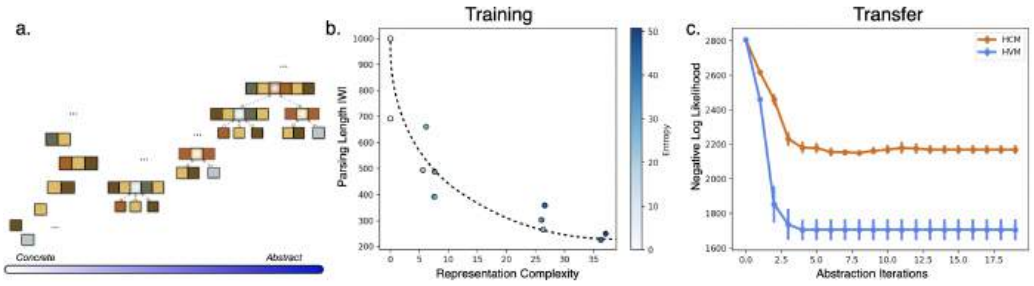


Figure 6: a. New chunks or variables are created from previously learned ones, adding increasingly nested structures to the representation graph. b. As the allowable distortion increases (moving right on the x-axis), the required rate to encode the data decreases (moving down on the y-axis). Introducing variables enables a more compressed sequence, which trades off with higher uncertainty introduced by the variables. c. Abstraction and transfer: The higher the abstraction layer, the higher the likelihood (lower negative log-likelihood) that HVM parses novel sequences.

With a model that can learn interpretable abstraction, we can delve into the relationship between learning layers of abstraction, compression, and uncertainty. HVM updates its dictionary each iteration by proposing chunks and variables, which are used to parse the sequence for the next iteration. New chunks or variables are created from previously learned ones, adding increasingly nested structures to the representation graph. This feature affords a controlled accessment of the relation between the level of abstraction, the amount of distortion that abstract representation introduces, and its relation to transfer. We measure nestedness using the representation complexity $RC(\mathbb{V}) = \sum_{v\in\mathbb{V}} \sum_{u\in\mathbb{E}(v)} -\log P(u|v)$, which is closely related to the encoding cost of the representation graph A.5. We also measured the uncertainty carried by the embedded variables in the chunks via entropy formulated as $\sum_{c\in\mathbb{C}} P(c) \sum_{v\in V(c)} \sum_{u\in E(v)} -P(u|v) \log P(u|v)$.

Figure 6a shows the relation between the three factors as a consequence of increasing abstraction learning iterations. As the representation graph becomes increasingly nested, the newly learned chunks explain a longer part of the sequence, and the chunks to parse the sequence contain more uncertain variables. This trade-off between compression and uncertainty reflects rate-distortion theory

specifying that the best possible compression of a signal $X$ contains a lower bound on quality loss specified by the Rate-Distortion Function (R(D)): $R(D) = \inf_Q R_Q \text{s.t.} \mathbb{E}[d(X, \hat{X})] \leq D$ (Shannon, 1959; Cover & Thomas, 2012). Taking representation complexity as a distortion function, as the level of abstraction increases, HVM learns patterns that span and predict longer parts of the sequence while more distortion and uncertainty are introduced. This observation on distortion with the level of abstraction resonates with previous findings on an increased level of mental errors with learning more abstract representations in humans (Lynn et al., 2020).

**Higher levels of abstraction implies more flexible transfer** If learning higher abstraction layers implies more distortion, what would be the benefit apart from compression? We suggest that the other side of the coin lies in generalization and demonstrates the effect of abstraction on model transfer. We trained HVMs with increasing abstraction layers and evaluated the models' transfer likelihood upon parsing a novel sequence. As shown in Figure 6b, the higher the abstraction layer, the higher the likelihood (lower negative log-likelihood) that HVM parses the transfer sequence. To compare, we also evaluated the transfer performance of the chunking model (HCM) with an increasing level of iteration. While the transfer performance for HCM improves with more layers of chunk learning, it converges at a higher negative log-likelihood. Learning an increasing abstract representation by the HVM enables it to compress novel transfer sequences in a more succinct way. Furthermore, the more abstraction layers are acquired, the higher the relative advantage for HVM that learns abstraction holds over the HCM that only learns chunks.

## 5 Related Work

Previous modeling work on abstraction can be divided into two categories. The first category implements abstraction as searching for commonalities in explicitly symbolic systems. These are predominately cognitive models meant to explain human abstraction and transfer behavior, such as humans can quickly solve a new problem by finding its conceptual analogies with old problems. Lu et al. (2021) implemented abstraction and concept analogy as finding a common graph structure between a ground problem and an analogical problem in a semantic relational network that captures this property. Another example is the copycat project (Hofstadter & Mitchell, 1994) that models how people generalize rules shared amongst a couple of examples by searching for rules from a network of concepts. These works model abstraction on symbolic, explicit representations of knowledge graphs but do not address how the explicit knowledge graph itself arises from perceptual data. Our work proposes a mechanism for learning explicit abstraction from sequences of perceptual experience.

The second category implements abstraction in implicit connectionist systems. Works on meta-learning models and LLMs suggest their ability to generalize across contexts and solve problems in a way similar to humans (Wei et al., 2022; Binz & Schulz, 2023) in some tasks while failing short on other abstract reasoning tasks (Fleuret et al., 2011). Meanwhile, abstraction has been argued to be implicitly present in artificial neural networks (Yee, 2019; Kozma et al., 2018; Johnston & Fusi, 2023; Ito et al., 2022) and biological neural activities (Bernardi et al., 2020; Goudar et al., 2023). However, the abstractions neural networks learn are challenging to interpret (Fan et al., 2021). Relative to this approach, our work provides a reductionist cognitive model that explicitly specifies the minimal components needed for a model to learn interpretable abstract structure from sequences. Besides cognitive models, our work also differs from other approaches to sequence learning, such as probabilistic context-free grammar (PCFG) (Jelinek et al., 1992) or adaptor grammar (Johnson et al., 2006). Our generative mechanism offers a probabilistic, hierarchical sequence generation model relying on chunk-based recursive generation and inventory growth rather than formal grammar rules. HVM's learning is also inventory-based instead of manipulating many substructures and rules.

## 6 Discussion

Our work has limitations. One is that variables are only proposed to be embedded between chunks and cannot come at the beginning or end of sequences, restricting the location of discoverable variables. Secondly, representation learned later in iteration depends on the earlier acquired representations. Furthermore, sometimes the variable structure can be arbitrarily nested depending on the learned representations up until that moment. Future work may look at optimizing the structure of the representation via refactorization features during learning. Finally, our work focused on the cognitive

resemblance of HVM and has not explicitly optimized the model specifically for computational efficiency. Future work may look into optimizing run-time efficiencies of models that exploit hierarchical structures in data adapting to specific application proposes.

Our work opens up interesting future directions both in cognitive science and machine learning. Previously, grammar learning, chunk learning, and statistical/associative learning were studied in isolation as distinct aspects of sequence learning. We propose via HVM that the three can be treated as finding invariant recurring chunks from sequences, the means of it (statistical/associative learning), and the consequence of it (learning grammar-like structures). Our work suggests a normative origin of concrete and abstract chunk learning as a learning agent uncovering the underlying entities that constitute perceptual sequences. Future work can relate this model to the development of concept understanding during development. Meanwhile, the close tie between abstraction and generalization also urges future hypothesis-driven research regarding preconditions for learning abstract concepts to acquire more abstract representations. This can help to illuminate the emergence of abstract representation during excessive data exposure and training and its relation to the generalizability of machine learning models (Power et al., 2022; Miller et al., 2024).

## 7 Conclusion

We propose a hierarchical variable learning model (HVM) that builds up entities of recurring abstract and concrete patterns from perceptual sequences, utilizing chunking, finding commonality, and variable proposal chunking and abstraction to unearth independently recurring entities in sequences with nested hierarchical structures. We show the model's resemblance to human sequence learning and transfer and highlight the relation between abstraction and generalization. Our work is relevant for cognitive science and the AI community in understanding memory in humans and machines.

## 8 Reproducibility Statement

Detailed information about the HVM algorithm, proof, generative model, test and experimental details and results can be found in the supplementary information section. The code used for the algorithm and experiments will be available as a comment to the reviewers and area chairs as a link to an anonymous repository as soon as the discussion forum for all submitted papers is open.

## References

W Abler. On the particulate principle of self-diversifying systems. *Journal of Social and Biological Systems*, 12(1):1–13, January 1989. ISSN 01401750. doi: 10.1016/0140-1750(89)90015-8. URL https://linkinghub.elsevier.com/retrieve/pii/0140175089900158.

John Anderson and Robert Milson. Human memory: An adaptive perspective. *Psychological Review*, 96:703–719, 10 1989. doi: 10.1037/0033-295X.96.4.703.

John R. Anderson. Retrieval of propositional information from long-term memory. *Cognitive Psychology*, 6:451–474, 1974. doi: 10.1016/0010-0285(74)90021-8.

Lawrence W. Barsalou. Perceptual symbol systems. *Behavioral and Brain Sciences*, 22(4):577–660, 1999. doi: 10.1017/S0140525X99002149.

Silvia Bernardi, Marcus K. Benna, Mattia Rigotti, Jerome Munuera, Stefano Fusi, and C. Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183: 954–967, 2020. doi: 10.1016/j.cell.2020.09.031.

Marcel Binz and Eric Schulz. Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120, 2023. doi: 10.1073/pnas.2218523120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2218523120.

BNC Consortium. The british national corpus, xml edition, 2007. URL http://hdl.handle.net/20.500.12024/2554.

R. H. S. Carpenter and M. L. Williams. Neural computation of log likelihood in control of saccadic eye movements. *Nature*, 377(6544):59–62, 1995. doi: 10.1038/377059a0. URL https://www.ncbi.nlm.nih.gov/pubmed/7659161.

Ted Chiang. Chatgpt is a blurry jpeg of the web. *The New Yorker*, Feb 2023. "OpenAI's chatbot offers paraphrases, whereas Google offers quotes. Which do we prefer?".

Allan M. Collins and M. Ross Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning & Verbal Behavior*, 8(2):240–247, 1969. doi: 10.1016/S0022-5371(69)80069-1. URL https://doi.org/10.1016/S0022-5371(69)80069-1.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

Valentina Cuccio and Vittorio Gallese. A Peircean account of concepts: grounding abstraction in phylogeny through a comparative neuroscientific perspective. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1752):20170128, June 2018. doi: 10.1098/rstb.2017.0128. URL https://royalsocietypublishing.org/doi/full/10.1098/rstb.2017.0128. Publisher: Royal Society.

Ian Cunnings and Patrick Sturt. Retrieval interference and semantic interpretation. *Journal of Memory and Language*, 102:16–27, 2018. doi: 10.1016/j.jml.2018.05.001. URL https://doi.org/10.1016/j.jml.2018.05.001.

Stanislas Dehaene, Fosca Al Roumi, Yair Lakretz, Samuel Planton, and Mathias Sablé-Meyer. Symbols and mental programs: a hypothesis about human singularity. *Trends in Cognitive Sciences*, 26 (9):751–766, 2022. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2022.06.010. URL https://www.sciencedirect.com/science/article/pii/S1364661322001413.

Jakub Dotlačil. Parsing as a Cue-Based Retrieval Model. *Cognitive Science*, 45(8):e13020, 2021. doi: https://doi.org/10.1111/cogs.13020. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.13020. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.13020.

Karl Duncker. On problem-solving. *Psychological Monographs*, 58(5):i–113, 1945. ISSN 0096-9753. doi: 10.1037/h0093599. URL http://doi.apa.org/getdoi.cfm?doi=10.1037/h0093599.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 0364-0213. doi: https://doi.org/10.1016/0364-0213(90)90002-E. URL https://www.sciencedirect.com/science/article/pii/036402139090002E.

Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, PP:1–1, 03 2021. doi: 10.1109/TRPMS.2021.3066428.

Paolo Ferragina and Giovanni Manzini. Indexing compressed text. *J. ACM*, 52(4):552–581, July 2005. ISSN 0004-5411. doi: 10.1145/1082036.1082039. URL https://doi.org/10.1145/1082036.1082039.

Johann Gottlieb Fichte. *The Science of Knowing: Fichte's 1804 Lectures on the Wissenschaftslehre*. second series. State University of New York Press, Albany, 2005. Original work published 1804.

François Fleuret, Ting Li, Charles Dubout, Emma K. Wampler, Steven Yantis, and Donald Geman. Comparing machines and humans on a visual categorization test. *Proceedings of the National Academy of Sciences*, 108(43):17621–17625, 2011. doi: 10.1073/pnas.1109168108. URL https://www.pnas.org/doi/abs/10.1073/pnas.1109168108.

Edward Fredkin. Trie memory. *Commun. ACM*, 3(9):490–499, sep 1960. ISSN 0001-0782. doi: 10.1145/367390.367400. URL https://doi.org/10.1145/367390.367400.

M. Gerlach and F. Font-Clos. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126, 2020.

Fernand Gobet, Peter Lane, Steve Croker, Peter Cheng, Gary Jones, Iain Oliver, and Julian Pine. Chunking mechanisms in human learning. *Trends in cognitive sciences*, 5:236–243, 07 2001. doi: 10.1016/S1364-6613(00)01662-4.

Vishwa Goudar, Barbara Peysakhovich, David J. Freedman, Elizabeth A. Buffalo, and Xiao-Jing Wang. Schema formation in a neural population subspace underlies learning-to-learn in flexible sensorimotor problem-solving. *Nature Neuroscience*, 2023. doi: 10.1038/s41593-023-01293-9.

S. A. Greibach. A note on undecidable properties of formal languages. *Mathematical Systems Theory*, 2:1–6, 1968. doi: 10.1007/BF01691341.

D. R. Hofstadter and M. Mitchell. The copycat project: A model of mental fluidity and analogy-making. In K. J. Holyoak and J. A. Barnden (eds.), *Advances in Connectionist and Neural Computation Theory*, volume 2, pp. 31–112. Ablex Publishing Corporation, 1994.

Takuya Ito, Tim Klinger, Doug Schultz, John Murray, Michael Cole, and Mattia Rigotti. Compositional generalization through abstract representations in human and artificial neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32225–32239. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/d0241a0fb1fc9be477bdfde5e0da276a-Paper-Conference.pdf.

Fred Jelinek, John Lafferty, and Robert Mercer. Basic methods of probabilistic context free grammars. *Speech Recognition and Understanding, NATO ASI Series*, 75, 01 1992. doi: 10.1007/978-3-642-76626-8_35.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Adaptor grammars: a framework for specifying compositional nonparametric bayesian models. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pp. 641–648, Cambridge, MA, USA, 2006. MIT Press.

W. J. Johnston and S. Fusi. Abstract representations emerge naturally in neural networks trained to perform multiple tasks. *Nat Commun*, 14:1040, 2023. doi: 10.1038/s41467-023-36583-0. URL https://doi.org/10.1038/s41467-023-36583-0.

Immanuel Kant. *Critique of Pure Reason*. The Cambridge Edition of the Works of Immanuel Kant. Cambridge University Press, New York, NY, 1998. Translated by Paul Guyer and Allen W. Wood.

George Konidaris. On the necessity of abstraction. *Current Opinion in Behavioral Sciences*, 29:1–7, October 2019. ISSN 2352-1546. doi: 10.1016/j.cobeha.2018.11.005. URL https://www.sciencedirect.com/science/article/pii/S2352154618302080.

Robert Kozma, Roman Ilin, and Hava Siegelmann. Evolution of abstraction across layers in deep learning neural networks. *Procedia Computer Science*, 144:203–213, 01 2018. doi: 10.1016/j.procs.2018.10.520.

P. Lison and J. Tiedemann. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, 2016.

Hongjing Lu, Nicholas Ichien, and Keith J. Holyoak. Probabilistic Analogical Mapping with Semantic Relation Networks. Technical Report arXiv:2103.16704, arXiv, October 2021. URL http://arxiv.org/abs/2103.16704. arXiv:2103.16704 [cs] type: article.

Christopher W. Lynn, Ari E. Kahn, Nathaniel Nyema, and Danielle S. Bassett. Abstract representations of events arise from mental errors in learning and memory. *Nature Communications*, 11(1):2313, May 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-15146-7. URL https://www.nature.com/articles/s41467-020-15146-7.

B. MacWhinney. *The CHILDES Project: Tools for analyzing talk*. Lawrence Erlbaum Associates, Mahwah, NJ, third edition edition, 2000.

George A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 1956. ISSN 0033295X. doi: 10.1037/h0043158.

Jack Miller, Charles O'Neill, and Thang Bui. Grokking beyond neural networks: An empirical exploration with model complexity, 2024.

Matthew R. Nassar, Julie C. Helmers, and Michael J. Frank. Chunking as a rational strategy for lossy data compression in visual working memory. *Psychological Review*, 125(4):486–511, Jul 2018. doi: 10.1037/rev0000101.

Gonzalo Navarro and Veli Mäkinen. Compressed full-text indexes. *ACM Comput. Surv.*, 39(1):2–es, apr 2007. ISSN 0360-0300. doi: 10.1145/1216370.1216372. URL https://doi.org/10.1145/1216370.1216372.

Victor Odouard and Melanie Mitchell. Evaluating understanding on conceptual abstraction benchmarks, 06 2022.

Stellan Ohlsson and Erno Lehtinen. Abstraction and the acquisition of complex ideas. *International Journal of Educational Research*, 27(1):37–48, January 1997. ISSN 0883-0355. doi: 10.1016/S0883-0355(97)88442-X. URL https://www.sciencedirect.com/science/article/pii/S088303559788442X.

Jean Piaget. Cognitive development in children: Development and learning. *Journal of Research in Science Teaching*, 2:176–186, 1964. doi: 10.1002/tea.3660020306. URL http://dx.doi.org/10.1002/tea.3660020306.

E. L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52:264–268, 1946. doi: 10.1090/S0002-9904-1946-08555-9.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe.

Atilla Schreiber, Shuchen Wu, Chenxi Wu, Giacomo Indiveri, and Eric Schulz. Biologically-plausible hierarchical chunking on mixed-signal neuromorphic hardware. In *Machine Learning with New Compute Paradigms*, 2023. URL https://openreview.net/forum?id=IuN2WXtFSY.

Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record*, 4(1):142–163, 1959.

Nathaniel J. Smith and Roger Levy. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319, 2013. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2013.02.013. URL https://www.sciencedirect.com/science/article/pii/S0010027713000413.

D. Sportiche, H. Koopman, and E. Stabler. *An Introduction to Syntactic Analysis and Theory*. Wiley, 2013. ISBN 9781118470473. URL https://books.google.de/books?id=MXkjAQAAQBAJ.

CARL STERN. Kant's theory of empirical concept formation. *The Southwestern Journal of Philosophy*, 8(2):17–23, 1977. ISSN 0038481X, 21541043. URL http://www.jstor.org/stable/43155148.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra,

Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Bert Van Oers. Contextualisation for abstraction. Cognitive Science Quarterly, 2000, 1, 279 - 305. *Cognitive Science Quarterly*, 1:279–305, January 2001.

W. von Humboldt and M. Losonsky. *Humboldt: 'On Language': On the Diversity of Human Language Construction and Its Influence on the Mental Development of the Human Species*. Cambridge Texts in the History of Philosophy. Cambridge University Press, 1999. ISBN 9780521667722. URL https://books.google.de/books?id=_UODbGlD4WUC.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus, 2023. URL https://arxiv.org/abs/2301.11796.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.

Shuchen Wu, Noemi Elteto, Ishita Dasgupta, and Eric Schulz. Learning Structure from the Ground up—Hierarchical Representation Learning by Chunking. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36706–36721. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/ee5bb72130c332c3d4bf8d231e617506-Paper-Conference.pdf.

Shuchen Wu, Mirko Thalmann, and Eric schulz. Motif learning facilitates sequence memorization and generalization, Dec 2023. URL osf.io/preprints/psyarxiv/2a49z.

Eiling Yee. Abstraction and concepts: when, how, where, what and why? *Language, Cognition and Neuroscience*, 34(10):1257–1265, November 2019. ISSN 2327-3798. doi: 10.1080/23273798.2019.1660797. URL https://doi.org/10.1080/23273798.2019.1660797. Publisher: Routledge _eprint: https://doi.org/10.1080/23273798.2019.1660797.

J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978. doi: 10.1109/TIT.1978.1055934.

# A Appendix / supplemental material

## A.1 Set up

An observational sequence $S$ is made up of discrete, integer valued, size-one elementary observational unit coming from an atomic alphabet set $\mathbb{A}$. One example of such an observational sequence $S$ is:

$$010021002112000...$$

An example belief set that contains only concrete chunks can be $\mathbb{B} = \{0, 1, 21, 211, 12, 2112\}$.

Using the belief set to parse the sequence $S$ results in the following partition. $\underline{0}\ \underline{1}\ \underline{0}\ \underline{0}\ \underline{21}\ \underline{0}\ \underline{0}\ \underline{2112}\ \underline{0}\ \underline{0}\ \underline{0}$.

Another example belief set $\mathbb{B}$ that contains chunks $\mathbb{C} = \{21v, 0100, 000\}$ with embedded variables $\mathbb{V} = \{v\}$. The denoting set of $E(v) = \{00, 12\}$. Then the sequence $S$ is parsed as $\underline{0100}\ \underline{21v}\ \underline{21v}\ \underline{000}$.

**Definition 1 (*Completeness*)**

*We say that a belief set is* complete *if at any point during the sequence parsing process, the upcoming observations can be explained by at least one chunk in the belief set.*

In this work, the learning mechanism guarantees that the belief set is complete.

**Definition 2 (*Parsing Length* $|W|$)**

*A parsing length $|W|$ of a sequence is the length of the sequence measured in chunks.*

### A.2 GENERATIVE MODEL

**Algorithm 1** Pseudocode to generate sequences with nested abstract hierarchies.

**Input:** A set of atomic elements $\mathbb{A}$; the number of combinations $d$; sequence length $l$
**Output:** $seq$, a sequence made of concrete observational units
$cg \leftarrow$ initialize representation graph
**for** $i \leftarrow 1$ **to** $d$ **do**
  $RAND \leftarrow$ random number between 0 and 1
  **if** $RAND > 0.5$ **then**
    // Object Creation
    $B \leftarrow cg$.objectsAndCategories
    $n_{combo} \leftarrow$ random.choice($[2, 3, 4, 5]$)
    $samples \leftarrow$ random.sample($B$, k=$n_{combo}$)
    **while** *any of the first and last element in* $samples$ *are categories* **do**
      $samples \leftarrow$ random.sample($B$, k=$n_{combo}$)
    **end**
    $newobject \leftarrow$ concatenate($samples$)
    $cg$.addChunk($newobject$)
  **end**
  **else**
    // Category Creation
    $B \leftarrow cg$.objects
    $n_{combo} \leftarrow$ random.choice($[2, 3, 4, 5]$)
    $samples \leftarrow n_{combo}$ random.sample($B$, k=$n_{combo}$)
    $newcategory \leftarrow$ create a new category denoting $samples$
    $cg$.addVariable($newcategory$)
  **end**
**end**
$cg$.assignProbabilitiesToObjects()
$sampledseq \leftarrow cg$.sampleObjectAndSpecifyVariables($l$)
$seq \leftarrow$ convert $sampledseq$ to sequence

A new object is created by concatenating a random selection of pre-existing objects or categories from the existing inventory. After the inventory has been expanded up to the $d$-th iteration, objects are assigned with an independent occurrence probability sampled from a flat Dirichlet distribution $f(p(c_1), ..., p(c_{|\mathbb{A}|}); \alpha_1, ..., \alpha_{|\mathbb{A}|}) = \frac{1}{\mathbf{B}(\mathbf{a})} \prod_{i=1}^{|\mathbb{A}|} P(c_i)^{\alpha_i - 1}$, $\alpha_i = 1 \forall i$. Where the beta function when expressed using gamma function is: $\mathbf{B}(\mathbf{a}) = \frac{\prod_{i=1}^{|\mathbb{A}|} \Gamma(\alpha_i)}{\Gamma(\sum_i^{|\mathbb{A}|} \alpha_i)}$, and $\mathbf{a} = (\alpha_1, .., \alpha_{|\mathbb{A}|})$. The parameters $(\alpha_1, .., \alpha_{|\mathbb{A}|})$ are identically set to one.

Similarly, a probability is sampled from a flat Dirichlet to assign the independent occurrence probability of the set of object $E(v)$ that each category $v$ denotes. $\forall v \in \mathbb{V}$, $f(p(c_1), ..., p(c_{|E(v)|}); \alpha_1, ..., \alpha_{|E(v)|}) = \frac{1}{\mathbf{B}(\mathbf{a})} \prod_{i=1}^{|E(v)|} P(c_i)^{\alpha_i - 1}$, $\alpha_i = 1 \forall i$. This procedure is done for each created category.

### A.3 APPROXIMATE RECOGNITION INVERSE

**Theorem 1.** *If the left entity $c_L$ and the right $c_R$ (which can be either a chunk or a variable) of a ground truth variable $v$ in addition to its denoting chunks $c_1, c_2, ..., c_m$ has been learned, and every parsing of $c_L$, $c_R$, and $c_1, c_2, ..., c_m$ is consistent with the constitution of the sequence by the its way of generation, then $c_1, c_2, ..., c_m$ will be necessarily proposed by the common adjacency and common preadjacency criterion into a novel variable $v'$ and the true denoting entities will be a subset of the denoting set $E(v')$, $\{c_1, c_2, ..., c_m\} \in E(v')$.*

*Proof.* By definition, $\{c_1, c_2, ..., c_m\}$ is in the adjacency $Adj(c_L)$ entries of $c_L$ and the preadjacency entries of $c_R$, $Preadj(c_R)$. And therefore, $\{c_1, c_2, ..., c_m\} \in Adj(c_L) \cap Preadj(c_R)$ and hence will be included in the set of denoting entities for a new variable. □

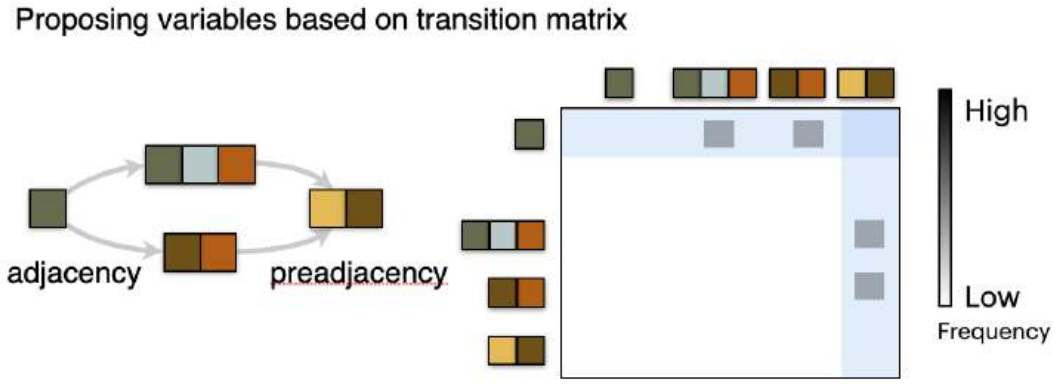


Figure 7: Common adjacency and preadjacency structure to identify a variable.

**Theorem 2.** *In an infinitely long sequence, a chunk $c$ in the generative model is made up by concatinating the entities $\{c_1, c_2, ..., c_n\}$, and these entities $\{c_1, c_2, ..., c_n\}$ are parsed in an identical way as how the generative model samples $c$, then HVM will eventually learn to chunk $c'$ including all the entities in $\{c_1, c_2, ..., c_n\}$.*

*Proof.* By contradiction. If any of the entities are not included in the learned chunks, a correlation will still exist in the transition entries between subparts that compose $c$, which will be resolved by the next chunk proposal iteration. □

In practice, identifying the ground truth representation graph is much more complicated, as the representations learned by HVM can be mapped to a context-sensitive grammar A.6, learning the correct chunks that exactly match the generative model relates to the non identifiability problem that multiple grammars can generate the same set of observed data (Post, 1946; Greibach, 1968). The sequence length comparable to the ground truth is possible to obtain. However, the dataset creation method does not guarantee a single optimal way, this is usually referred to as the undecidability problem in grammar induction literature: inferring the exact grammar from positive examples is undecidable. Also is the case for deciding if two grammar classes are equivalent. Our formulation of the generative model can be shown to be equivalent to context-sensitive grammar A.6, which is more complex and undecidable than context-free grammar. No general algorithm can take any finite set of examples and produce a grammar that will generate all and only those examples. Because of undecidability, heuristic and approximate methods are often used in practice in the linguistics grammar induction literature and formal language theory. These methods aim to find solutions that are good enough rather than exact ones. As the optimality of the ground truth cannot be guaranteed due to the common source of this problem with grammar induction, our approach restricts and simplifies the structures of patterns that are feasible for the HVM. Since there is a theoretical limit to what can be achieved when inferring grammatical structures from data. Our approach to HVM uses approximate, heuristic, and probabilistic methods to tackle this problem by imposing parsimonious computational characteristics that have been pronouncedly observed in the cognitive science learning literature. Instead of comparing the model with the specific chunks of the ground truth, we use a set of evaluation measures to evaluate the model's performance.

### A.4 Parsing Search Steps

HCM retrieves learned chunks to match them with the upcoming sequence, parsing the largest chunk that aligns with the sequence. However, as the dictionary size $|\mathbb{C}|$ grows, the number of search steps

**Algorithm 2** HVM (online version, for learning sequences from human experiments)

**Input:** Learning sequences $seq$; representation graph $cg$; boolean flag for chunk learning $threshold_chunk$; boolean flag for variable learning $abstraction$
**Output:** $cg$, $chunk_record$
$cg \leftarrow$ initialize representation graph `// Initialize chunk record`
$chunk_record \leftarrow \{\}$
$t \leftarrow 0$
**while** *not* $seq_over$ **do**
  $current_chunks, cg, seq, chunk_record \leftarrow identify_latest_chunks(cg, seq)$
  $cg \leftarrow learning_and_update(current_chunk, chunk_record, cg, threshold_chunk = True)$
  **if** $abstraction$ **then**
    $cg \leftarrow abstraction_update(current_chunks, cg)$
  **end**
  $cg.forget()$ `// multiply all frequency record by` $\theta$
**end**

increases, leading to inefficiency. To address this, HVM organizes the chunks in $\mathbb{C}$ within a parsing graph, harnessing shared nodes across multiple chunks to speed up the chunk identification process.

**Definition 3 (*Parsing Graph (PG)*)**

*Parsing graph is a graph for chunk identification. The nodes inside such a graph are the chunks $\mathbb{C}$. Each parent node is the overlap of all of its children.*

To parse a chunk, the model traverses the memory tree structure to find the path up to the deepest node inside the tree consistent with the upcoming sequence. Pseudocode A.4 describes this process.

**Algorithm 3** Pseudocode for traversing a tree to find a path consistent with an upcoming sequence

**Input:** $rootNode$, the root node of the tree; $sequence$, the sequence to be matched
**Output:** $path$, the path to the leaf consistent with the sequence, if found
$path \leftarrow$ an empty list **if** *not* $rootNode$ **then**
  **return** $null$
**end**
**else**
  **return** TRAVERSETREE(()$rootNode$, $sequence$, $path$)
**end**
**Function** `TraverseTree(`*node, sequence, path*`)` **:**
  $path$.append($node$) **if** ISCONSISTENT*(node, sequence)* **then**
    **if** *node.isLeaf()* **then**
      **return** $path$
    **end**
    **forall** *child in node.children* **do**
      $result \leftarrow$ TRAVERSETREE($child$, $sequence$, $path$.copy()) **if** $result$ **then**
        **return** $result$
      **end**
    **end**
  **end**
  **return** $null$

**Definition 4 (*Parsing Search Steps*)**

*Parsing search step refer to the number of search comparison made in the parsing graph to find the deepest chunk in the tree consistent with the upcoming sequence.*

**Example** Shown in Figure 8, when the sequence upcoming is 123, one starts from the ancestors of the graph and check whether 1 or 2 is at the beginning of the sequence, identifies 1, and proceeds into

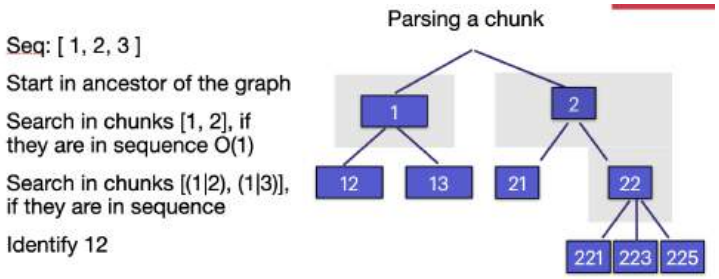


Figure 8: Example process when parsing a chunk using parsing graph. Gray region denotes chunk overlaps, which are the non-leaf nodes inside the parsing graph being the common prefix of their children.

the subtree which stems from 1. The leaves are contains 12 and 13. Then HVM checks whether 12 is in the sequence or 13, thereby identifying 12.

The mechanism to identify the biggest concrete chunk that can be used to parse the sequence involves the following steps:

- Search from the top of the tree (nodes with no parents)
- Find the node in the next layer consistent with sequence
- Go to the children of the node until the node contains no more children (leaf node) or none of the node's children is consistent with the sequence

As illustrated in Figure 8, the parsing search steps of chunk 12 is: $PSS(12) = 2+2 = |A| + |R(A)|$ For chunk 221: $PSS(221) = 2+2+3$ For 21: $PSS(21) = 2+2$

**The advantage of memory abstraction**

**Theorem 3** (Guarantee on fewer search steps)**.** *Identifying a chunk in a tree-structured memory takes fewer search steps than without.*

*Proof.* The hierarchically structured memory system reduces the searching step to the number of branches on the path that leads to the identified chunk, which is a subset of all nodes in the tree. □

**Expected parsing search steps** For a chunk $c$ that occur with probability $P(c)$, the parsing search steps is denoted as $PSS(c)$.

As an example, three chunks $c_1$, $c_2$, and $c_3$ composes a sequence and they are parsed with probability $P(c_1)$, $P(c_2)$, and $P(c_3)$ in a sequence.

Without the hierarchical memory structure, the average parsing search steps to identify chunks in the sequence is:

$$PSS(c_1)P(c_1) + PSS(c_2)P(c_2) + PSS(c_3)P(c_3)$$

With the prefix Trie, the number of times needed to check whether a chunk matches a certain abstraction branch requires a number of parsing steps:

$$PSS(c) = N_{children}(Pa(c)) \tag{1}$$

$Pa(c)$ denotes the parent of chunk $c$ in $PG$, $N_{children}(Pa(c))$ is the number of children that chunk $c$'s parent contain. An abstract chunk $a$ in such a parsing graph can be the prefix intersection of three chunks $c_1$, $c_2$, and $c_3$, $a = c_1 \cap c_2 \cap c_3$. Parsing abstraction summarizes all the subordinate chunks that underlies such abstraction. Therefore the abstraction node is more likely to be parsed compared to individual chunks.

$$P(a) = (P(c_1) + P(c_2) + P(c_3))$$

Additionally, abstraction can be used as an anchor to find the subsequent denoting chunks. $P(c_1|a)$, $P(c_2|a)$ and $P(c_3|a)$. The expected parsing search steps to identify chunks for such sequence $S$ becomes:

$$\mathbb{E}_S(PSS) = PL(c_1 \cap c_2 \cap c_3)(P(c_1)+P(c_2)+P(c_3))+PL(c_1-a)P(c_1|a)+PL(c_2-a)P(c_2|a)+PL(c_3-a)P(c_3|a)$$

Generalizing this formula to any parsing graph, the expected parsing search steps given the parsing graph's internal abstraction and concrete chunks would be

$$EPSS(PG) = \sum_{c \in \mathbb{C}} P(c) \sum_{path(c)} len(N_{children}(PA(c))) \tag{2}$$

$path(c)$ is the path starting with the root node of the graph that leads to the identification of the chunk node c. Since $P(c)_{c_i \in path(c)} = P(a) \prod_{c_i \in path(c)} P(c_i|pa(c_i))$, one can also give a lower and upper bound on the number of parsing search steps needed to arrive at a random chunk in the parsing graph:

$$EPSS(PG) = \sum_{v \in \mathbb{C}} P(v|PA(v))PSS(v - PA(v)) \tag{3}$$

$PSS(a-b)$ denotes the subgraph that starts from $b$ which reaches $a$. If $v$ is inside the ancestor nodes, then $ancestor(v) = \varnothing$, and therefore $P(v|ancestor(v)) = P(v)$.

The expected parsing search steps for a representation graph can be calculated by averaging out the expected parsing search steps for each sub graph inside the parsing graph. The lower bound is the number of steps that leads to the shallowest node, and the upper bound is the number of steps that leads to the deepest leaf node.

$$\arg\min_{c \in \mathbb{C}} \sum_{u \in path(c)} len(n_{children}(PA(c))) \leq \mathbb{E}_{c \in \mathbb{C}}(c) \leq \arg\max_{c \in \mathbb{C}} \sum_{u \in path(c)} len(n_{children}(PA(c))) \tag{4}$$

### A.5 Encoding Cost

A representation graph $RG$ specifies a distribution of observation instances encoded in a compositional manner. One can the minimal encoding cost to distinguish all the variables.

$$RC(G) = \sum_{v \in G} \sum_{u \in E(v)} -\log P(u|v) + \sum_{v \in G} -\log P(v) + \sum_{c \in G} -\log P(c) \tag{5}$$

The more ambiguous a variable is, the less resources needed to encode the variable; the smaller the variable graph is, the less edges it has, the bigger the probabilities of parsing and variable identification, the smaller the encoding cost. Every time when a new edge points from a pre-existing variable to a new variable, the encoding resource expands by an amount determined by the conditional probability.

**Example** Representation graphs with different nested structure translates to different encoding cost. In Figure 9, three graphs contains an increasing abstraction specificity. In graph 1, the variable 'world' is split into two variables: animals and plants. Each of these variables do not denote a specific observation, but serves as a placeholder for any instances of specific observation that fits into that particular category. The encoding cost for such a graph, given the specified observational probability on the edges would be:

$$EC(g1) = -\log(P(World))-\log(P(Plants|World))-\log(P(Animal|World)) = -2\log(0.5) \tag{6}$$

Since once the variable $World$ has been identified, one only need to distinguish the subcategories inside the variable $World$, and the minimal code length to distinguish one subcategory (animals) from another (plants) would be the conditional probability: $P(Plants|World)$.

Whereas if $g1$ does not contain a nested structure, one would need to encode the variable world separately from the variable animal and variable plants, in this way, the alternative encoding length without an edge connecting the sub categories with the main category would be:

$$EC(\hat{g1}) = -\log(P(World)) - \log(P(Plants)) - \log(P(Animals)) = -2\log(0.5) \tag{7}$$

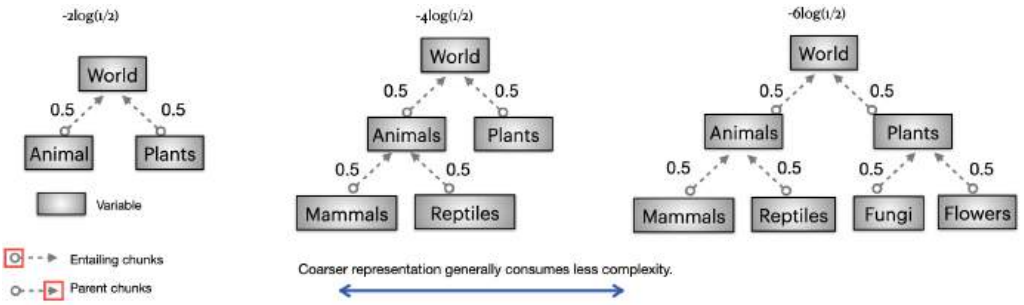


Figure 9: An increasing level of abstractions

For $g2$, the case is slightly different: the variable category that denotes animals is further split into the category of mammals and reptiles. Thereby, extra encoding costs are needed.

$$\begin{aligned} EC(g2) &= -\log(P(World)) - \log(P(Plants|World)) - \log(P(Animals|World)) \\ &\quad - \log(P(Mammals|Animals)) - \log(P(Reptiles|Animals)) \\ &= -4\log(0.5) \end{aligned} \tag{8}$$

If each of these categories are encoded separately and each variable is not pointed to other variables, the information content to encode a variable graph without edges in the case of $\hat{g2}$ would be:

$$\begin{aligned} EC(\hat{g2}) &= -\log(P(World)) - \log(P(Plants)) \\ &= -\log(P(Animals)) - \log(P(Mammals)) - \log(P(Reptiles)) \\ &= -6\log(0.5) \end{aligned} \tag{9}$$

Since $P(Mammals) = P(mammals|animals)P(animals)$.

This difference is more pronounced going from $g2$ to $g3$:

$$\begin{aligned} EC(g3) &= -\log(P(World)) - \log(P(Plants|World)) - \log(P(Animals|World)) \\ &\quad - \log(P(Mammals|Animals)) - \log(P(Reptiles|Animals)) \\ &\quad - \log(P(Fungi|Plants)) - \log(P(Flowers|Plants)) \\ &= -6\log(0.5) \end{aligned} \tag{10}$$

Whereas for $\hat{g3}$, not encoding variables in a nested structure would result in an encoding cost of

$$\begin{aligned} EC(\hat{g3}) &= EC(\hat{g2}) + -\log(P(Fungi)) - \log(P(Flowers)) \\ &= -10\log(0.5) \end{aligned} \tag{11}$$

These examples illustrate organizing variables into a nested recursive structure saves encoding cost.

### A.6 Alternative formation as learning a context sensitive grammar

HVM learns a 5-tuple from the sequences $G = \{\mathbb{A}, \mathbb{C}, \mathbb{V}, \mathbb{R}, \mathbb{P}\}$. A set of atomic units $\mathbb{A} = \{a_l\}$
A set of chunks $\mathbb{C} = \{c_k\}$, $k = 1, 2, ...,$, each chunk $c_k$ is a sequence of atomic units and variables. The model uses chunks from $\mathbb{C}$ to parse the observation sequence. Assume a random variable $C$ as the chunk being parsed, taking a value $c$ from $\mathbb{C}$, $c \in \mathbb{C}$, $c \sim P$. $P$ is the parsing probability. $\sum_{c \in \mathbb{C}} P(C = c) = 1$
A set of variables $\mathbb{V} = \{v_i\}$, $i = 1, 2, ...,$
A set of rules $\mathbb{R} = \{E(v_1), E(v_2), ..., E(v_n)\}$ that each specifies the set of chunks $E(v) = \{c_j\}$ denoting each variable $v \in \mathbb{V}$. Probabilities on variable denoting $\forall i$, $\sum_j P(v_i \to c_j) = 1$

### A.7 RATE-DISTORTION THEORY

Rate-distortion theory specifies that the best possible compression of a signal $X$ contains a lower bound on quality loss specified by the Rate-Distortion Function (R(D)): $R(D) = \inf_Q R_Q \text{s.t.} \mathbb{E}[d(X, \hat{X})] \leq D$ (Shannon, 1959; Cover & Thomas, 2012). According to the $RD$ theory, the minimum $R$ (the amount of compression) at which information can be transmitted over a communication channel for a given level of information loss D is specified by the **Rate-Distortion Function** $(R(D))$: $R(D) = \min_{p(\hat{x}|x):E[d(X,\hat{X})]\leq D} I(X; \hat{X})$. The mutual information quantifies how much information needs to be preserved during compression to achieve this fidelity, and the function minimizes this quantity while satisfying the distortion constraint. Written in another way,$R(D) = \inf_Q R_Q$ s.t.$E[d(X, \hat{X})] \leq D$. $Q$ represents the encoding function that maps from $x$ to $\hat{x}$. RD is agnostic to the choice of the distortion function, and the representation complexity, which measures the nestedness of the variables learned by HVM.

### A.8 PARSING PROBABILITY

A sequence $S$ is parsed by the chunks in $\mathbb{C}$ to obtain a distribution of chunk parsing probability $P_{\mathbb{C}}$. This distribution on the support set of chunks has shapes dependent on the parsing mechanism. HVM employs a greedy strategy and chooses the deepest consistent chunk in the memory tree to parse a segment of the upcoming sequence. Other parsing strategies will result in alternative distributions for $P_{\mathbb{C}}$, in addition to the transition probability $P_{\mathbb{C}}(c_i|c_j)$.

Define the set $H_{\mathbb{C}}(S)$ as the set of all distributions results in any parsing method using the chunks in $\mathbb{C}$ to parse a sequence $S$, and any parsing probability $P \in H_{\mathbb{C}}(S)$. Hence each evaluation measure on the representation is bounded by the optimal and the most unfortunate parsing occasions. Taking the predictive power measure as an example:

$$\arg \min_{P \in H_{\mathbb{C}}(S)} \mathbb{E}_P|c| \leq \mathbb{E}_{P_{HVM}}|c| \leq \arg \max_{P \in H_{\mathbb{C}}(S)} \mathbb{E}_P|c| \tag{12}$$

The measured value is bounded by the expected value from the best parsing distribution and the worst parsing distribution.

### A.9 UNCERTAINTY

Variables introduce uncertainty. Without variables, the chunks inside $\mathbb{E}_v$ needs to be encoded as distinct chunks, consuming an encoding cost of $\sum_{c \in \mathbb{E}_v} -\log P(c)$.

We have the ground truth set of sequential observational units $gt$. Let's say in a recall task, the agent learns a variable chunk to describe the ground truth, the variable denotes to three concrete chunks $c_1$, $c_2$ and $c_3$. The accuracy when $c_1$ is being sampled will be $D(gt, c_1)$, when $D$ is a distance function evaluating the agreement between the ground truth chunk $gt$ and $c_1$. Then one can evaluate the expected accuracy within a variable:

$$\mathbb{E}(gt, v) = \sum_{c \in \mathbb{E}_v} P(c|v) D(gt, c) \tag{13}$$

$P(c|v)$ is the probability of sampling chunk $c$ given that the variable $v$ is identified.

Compared to chunks without variables, recalling chunks with embedded variables introduces variability, and brings down recall accuracy. But this strategy is especially suited when encoding resources is limited. Then, it is better when the resources are assigned to the more probable chunks that are predictive of a bigger sequence, while the varying entities that occur with lower probability can be denoted as a variable that serves as a placeholder.

$$Uncertainty(G) = \sum_{c \in \mathbb{C}} P(c) \sum_{v \in C} H(v) \tag{14}$$

And $H(v) = -\sum_{c \in E(v)} P(c) \log P(c)$.

### A.10 OTHER EVALUATION MEASURES

**Explanatory Volume:** $\frac{|S|}{|W|}$ The length of the original sequence (in atomic units) divided by the length of the parsed sequence (in units of chunks), i.e., The average size of the sequence that the current representation graph can explain at each parsing step.
**Sequence Negative Log Likelihood:** The lower bound of information content needed to encode the sequence $S$, as $S$ is parsed into chunks $(c_1, c_2, ..., c_n)$, $-\log P(S) = -\log(\prod_{c_i \in \mathbb{C}} P(c_i))$.
**Representation Entropy:** The uncertainty contained within each chunk parse $\sum_{c \in \mathbb{C}} P(c) \sum_{v \in V(c)} \sum_{u \in E(v)} -P(u|v) \log P(u|v)$

### A.11 REGRESSING LLM ON HUMAN DATA

We also regressed the negative log likelihood (NLL) of all LLMs against human sequence recall time and presented the resulting R-squared values in Figure 10. During the training block, the NLL of LLMs shows a stronger correlation with human recall times compared to alternative models. However, when it comes to human transfer, the cognitive models demonstrate a much stronger correlation than the LLMs, with HVM aligning most closely to human transfer performance.

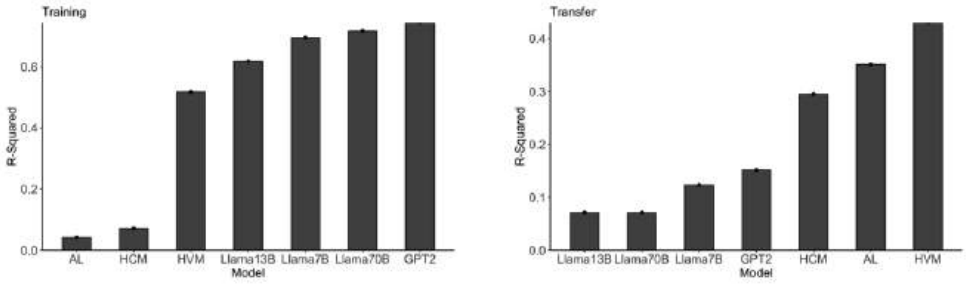


Figure 10: Regressing all models' sequence likelihood on human sequence recall time.

# 9

# Afterward: From Dionysius emerges Apollo

> *Wherever the Dionysian prevailed, the Apollonian was checked and destroyed.... wherever the first Dionysian onslaught was successfully withstood, the authority and majesty of the Delphic god Apollo exhibited itself as more rigid and menacing than ever.*
>
> — **Friedrich Nietzsche**
> The Birth of Tragedy

At the beginning, there was only the wild, untamed world of Dionysius. The god of wine's spirit ran wild like feral vines, thick and twisting, spreading across the land. Sensations, emotions, and desires flowed freely; chaos reigned supreme. Every moment was saturated with stimuli — flashes of ecstatic joy, rumbles of anger, and swells of passion. Dionysius' followers, ecstatic and fervent, sang in drunken unison, their feet pounding the earth in a frenzied ritual to honor their god. Everything is connected but without clear distinction —raw, intense, and boundless. The world was too vast, too complex to comprehend all at once, threatening to overwhelm the gods and mortals alike.

As Dionysius continued his wild revelry, a seed force began to stir deep within the heart of this madness and sensory overload. It was a faint glimmer of form within the formless, a rhythm within the dissonance. This seed was Apollo. Apollo noticed the repetition in the madness, the patterns beneath the confusion. The dancers, though frenzied, reiterated certain steps. The vines, though wild, grew in predictable directions. Even the songs of the revelers, returned to familiar melodies. According to the noticeable patterns, Apollo *chunked* the sequences of chaotic sensations into manageable, meaningful units: the once incomprehensible swirl of sensation was sorted into clear entities, each with its own beginning, middle, and end. The frenzied dancers slowed as their movements took on the grace of choreographed steps. The wild music began to follow a rhythm, its notes falling into place. The dancing bacchanals became structured rituals, their movements still passionate but now

guided by purpose and harmony. The wild screams turned to song, each note precise, each rhythm intentional.

Through the process of chunking, Apollo — the god of sunlight, music, and prophecy — emerged as a radiant presence, taming the chaos of perception. He transformed Dionysius' boundless primal power and wild energy into something finite and comprehensible, distilling raw, unfiltered experience into meaningful chunks and symbols. In doing so, laying the foundation for reason, knowledge, art, and beyond. Together, the two gods existed within the same perceptual world while governing a dichotomous characterization of reality: Dionysius was the intoxication of raw experience — perception as disordered and undifferentiated forms; Apollo was the clarity that followed — the ordered understanding that emerged after the chaos, differentiated by forms.