One model for the learning of language

Yuan Yang[a] and Steven T. Piantadosi[b,1]

[a]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332; and [b]Department of Psychology, Helen Wills Neuroscience Institute, University of California, Berkeley, CA 94720

Edited by Adele Goldberg, Linguistics, Princeton University, Princeton, NJ; received October 20, 2020; accepted November 18, 2021 by Editorial Board Member Susan T. Fiske

A major goal of linguistics and cognitive science is to understand what class of learning systems can acquire natural language. Until recently, the computational requirements of language have been used to argue that learning is impossible without a highly constrained hypothesis space. Here, we describe a learning system that is maximally unconstrained, operating over the space of all computations, and is able to acquire many of the key structures present in natural language from positive evidence alone. We demonstrate this by providing the same learning model with data from 74 distinct formal languages which have been argued to capture key features of language, have been studied in experimental work, or come from an interesting complexity class. The model is able to successfully induce the latent system generating the observed strings from small amounts of evidence in almost all cases, including for regular (e.g., a^n, $(ab)^n$, and $\{a, b\}^+$), context-free (e.g., a^nb^n, a^nb^{n+m}, and xx^R), and context-sensitive (e.g., $a^nb^nc^n$, $a^nb^mc^nd^m$, and xx) languages, as well as for many languages studied in learning experiments. These results show that relatively small amounts of positive evidence can support learning of rich classes of generative computations over structures. The model provides an idealized learning setup upon which additional cognitive constraints and biases can be formalized.

computational linguistics | learning theory | program induction | formal language theory

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

One of the central debates in language acquisition is whether the structures of natural language are genetically specified or learned through experience. A key tool in this debate has been the use of formal mathematical analysis to determine what learners could or could not logically induce from the type of data they observe. Perhaps the most influential formal result is Gold's Theorem (1), which implies that there are classes of even regular languages (2) which contain members that cannot be identified by learners who rely on positive examples alone. Gold's result has been interpreted to mean that human children could not succeed in learning natural language without substantially informative innate constraints, following arguments from Chomsky (3). Gold's result gave rise to detailed formal theories of learning under similar assumptions (4–8). Learnability proofs for certain classes of grammars have also been formulated (9–16), as well as closely related theories of induction in computer science (17–22).

However, Gold's negative learnability result is not widely accepted as relevant to human language (13, 23, 24). Gold relied on a worst-case analysis, which assumes that a parent would intentionally mislead a child through an unbounded number of incorrect hypotheses if they can. This made analysis tractable using his formal tools, but, critically, was disconnected from naturalistic parent–child interactions. In situations where some of Gold's assumptions are altered, learnability can be established under less antagonistic assumptions (22, 25–27). A striking, recent result shows that languages can be learned using positive evidence alone, out of the maximally unconstrained space of all possible computations (28). This more optimistic analysis involves several critically different assumptions. For one, it assumes that sentences are sampled from a distribution, meaning that it uses an average-case analysis rather than worst case; it also quantifies learning through how well a learner could predict future strings. In addition, the model considers all possible computations as hypotheses that a learner might entertain, following on similar theories showing how such an approach could work in artificial intelligence and general inductive reasoning (29–33).

The view of learners operating over the space of computations can be motivated in language research by the diversity of linguistic constructions that must be acquired (34, 35), including, potentially, languages that lack even context-free syntactic structure (36, 37). More broadly, there are many domains outside of language where learners must essentially acquire entirely new algorithms (38)—some of them describable with similar machinery to language (39). It is ordinary for children to come to know new computational processes in learning tasks like driving, cooking, programming, or playing games. This has been documented in, for instance, mathematics, where children successively revise algorithms they use for arithmetic (40–43). Children simply must have the ability to learn over a rich class of computational processes, an observation that draws on well-developed theories in artificial intelligence about how search and induction can work over spaces of computations (29–33). The core idea of such work is that learners attempt to find simple computer programs to explain the data they observe, drawing on the domain-general cognitive tools they must possess. Learners, in this view, are much like scientists (44) who look at data and construct computational theories in order to explain the patterns that they observe; indeed, the approach builds on similar efforts to automate scientific discovery (45–47).

While the above work addresses learnability as a mathematical question, the central ideas remain relatively unconnected to contemporary ideas about mental representations. First, most theoretical analyses do not provide working implementations. Second, the representations prior analyses have used tended to focus on Turing machines or treated languages as abstract sets,

Significance

It has long been hypothesized that language acquisition may be impossible without innate knowledge of the structures that occur in natural language. Here, we show that a domain general learning setup, originally developed in cognitive psychology to model rule learning, is able to acquire key pieces of natural language from relatively few examples of sentences. This develops a new approach to formalizing linguistic learning and highlights some features of language and language acquisition that may arise from general cognitive processes.

Author contributions: Y.Y. and S.T.P. designed research, performed research, analyzed data, and wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission. A.G. is a guest editor invited by the Editorial Board.

This open access article is distributed under Creative Commons Attribution License 4.0 (CC BY).

[1]To whom correspondence may be addressed. Email: stp@berkeley.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2021865119/-/DCSupplemental.

Published January 24, 2022.

but philosophers, cognitive scientists, and linguists have made a convincing case that compositional representations are natural for capturing human-like thinking (48–52) and computation itself (53–56). Indeed, theories of inductive inference centered on compositionality provide good fits to human learning curves across domains (57–61). Third, prior work has not targeted many of the specific computations that are thought to be important for natural language. Here, we address these problems by providing a learning model that uses components which have been independently argued for in concept learning experiments. We show how a compositional model can learn key generative pieces of natural language, including many that motivated classic theories of linguistic representation. We begin by considering simple linguistic patterns that inaugurated both modern linguistics and modern computer science.

Our starting point—and the inspiration for our title—is Chomsky's "Three models for the description of language" (62) which noted that many dependencies in natural language could be captured abstractly with distinct kinds of computational devices (63–65). Some devices require a finite amount of memory (e.g., finite-state machines), some require an unbounded amount of memory in a stack, and some require even more powerful systems. Chomsky's work, in collaboration with Marcel-Paul Schützenberger, charted out different kinds of string sets and classified what classes of devices could compute each. Following this literature, we use the term "formal language" to refer to a set of strings, typically one that is generated according to a simple computational pattern.

For example, the tail recursion allowed by English adjectives ("The adorable, friendly, young monkey") mirrors the structure in the simple formal language $a^* = \{\epsilon, a, aa, aaa, \ldots\}$, where zero ($\epsilon$) or more successive adjectives (a) could be concatenated into a valid substring of English. In turn, this can be captured with a computational device with one internal state, which may optionally emit an a and return to the same state. Dependencies between determiner–noun pairs ("Bring me two boats, three accordions, and six babies.") mirrors the formal language $(ab)^+$ where each determiner (a) requires a corresponding noun (b). The formal language $\{a^n b^n : n = 1, 2, \ldots\} = \{ab, aabb, aaabbb, \ldots\}$ might characterize the key dependencies in English "if–then" relationships. Specifically, every "if" (an "a") must be followed by a "then" (a "b"), as in "If Mary cried then John was sad then John cares about Mary" (63).[*] This formal language cannot be generated or recognized by a computational device with a finite number of states but can be captured by a context-free grammar. Although the applicability of such examples to natural language is often contested (66), the study of such formal languages has provided a fertile ground for linguistics (67) to characterize what computations underlie human language (2, 39, 62, 63). A considerable amount of effort has gone toward understanding whether natural language can be captured entirely by context-free grammars or other systems (66, 68–70), or how non–context-free aspects may be handled (71), although many arguments that have been made are inadequate technically (72).

Here, we study a variety of formal languages which pose many of the key challenges that have attracted attention in learning theory, including the underdetermination of grammars by evidence, the "subset problem" of how learners appropriately constrain their generalizations, and the puzzle of how learners come to know productive generative processes. As we show, these can all be resolved by formalizing proper statistical inference over a space of computations. We apply our model to a variety of test cases spanning simple formal languages, versions of stimuli from experimental work, and a simple English grammar.

Formal Model

Following a growing body of work in compositional Bayesian models (38, 58, 59, 61, 73–88), we assume that the representations learners must discover are built by combining primitives in a language of thought (LOT) (49) to form the mental analog of programs. In this setup, learners observe data (here, strings) and compare hypotheses that are built out of primitives, as a way to explain the data, much as scientists might consider possible physical laws which are compositions of mathematical operations (e.g., $F = G \cdot m_1 \cdot m_2 / r^2$). The specific primitives that we assume are motivated by minimalist functional programming languages like Scheme (54) which try to build in as little as is practical while remaining able to express all computations.

Table 1 lists the types of operations we consider, many of which are meant to be domain-general primitives that can be—and have been—deployed in other areas of concept learning. The first kind are list/string operations, `pair`, `first`, and `rest`, which build and manipulate sequences of characters from the alphabet. The functions `pair` and `append` are similar in spirit to "merge" in minimalist linguistics (89, 90), except they come with none of the associated machinery that is required in those theories; here, they only concatenate. The function `insert` puts one string in the middle of another (e.g., `insert`($'abcd'$, $'efg'$) yields $'abefgcd'$), which allows concise construction of long-range dependencies

Table 1. Assumed primitive functions

Primitive	Description
Functions on lists (strings)	
`pair`(*L*, *C*)	Concatenates character *C* onto list *L*
`first`(*L*)	Return the first character of *L*
`rest`(*L*)	Return everything except the first character of *L*
`insert`(*X*, *Y*)	Insert list *X* into the middle of *Y*
`append`(*X*, *Y*)	Append lists *X* and *Y*
Logical functions	
`flip`(*p*)	Returns true with probability *p*
`equals`(*X*, *Y*)	True if string *X* is the same string as *Y*
`empty`(*X*)	True if string *X* is empty; otherwise, false
`if`(*B*, *X*, *Y*)	Return *X* if *B* else return *Y* (*X* and *Y* may be lists, sets, or probabilities)
`and`, `or`, `not`	Standard Boolean connectives (with short circuit evaluation)
Set functions	
Σ	The set of alphabet symbols
{*s*}	A set consisting of a single string
`union`(*set*, *set*)	Union of twos sets
`setminus`(*set*, *s*)	Remove a string from a set
`sample`(*set*)	Sample from *s* of strings
Strings and characters	
ϵ	Empty string symbol
x	The argument to the function
'a', 'b', 'c', . . .	Alphabet characters (language specific)
Function calls	
`Fi`(*z*), `Fmi`(*z*)	Calls factor *Fi* with argument *z*; the *Fmi* version memoizes probabilistic choices (see text)

The space of hypotheses consists of all compositions of these functions that respect the input and output types.

[*] Note here, and elsewhere, that the connection is analogous rather than a direct formalization of English grammar, since "If if then then" is not a valid English sentence. We focus on examples like $a^n b^n$ instead of more complex formal languages that directly capture intervening material in order to simplify presentation. Note that closure properties of formal languages under, for example, intersection and homomorphism likely mean that simple formal languages are relevant to richer sets of strings.

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

because it permits dependent elements to be generated consecutively (e.g., *abcd*) and then displaced (e.g., $ab \ldots cd$). The assumed Boolean operators include common logical connectives and conditionals. Both `flip` and `sample` are notable in that they are stochastic, allowing nondeterminism in the program (91, 92), an ability that is possibly itself useful for organisms with finite memory (93). We note that the classical Chomsky hierarchy no longer directly applies to formal languages in this probabilistic setting (94, 95). Finally, we allow a computation to call another function `F1`, `F2`, etc. (potentially itself). This call may be memoized, which means that it remembers any stochastic choices that were made on a previous call with the same arguments (91). This use of recursion is meant to be domain general, as humans deal with recursion outside of syntax, like pragmatics (96), and even outside language (97–99). We note the similarity between these primitives and others used in modeling human conceptual systems (59, 61, 73, 74, 76, 78, 91, 100).

Valid compositions of primitives—those that respect the input and output types of each function—define an infinite set of hypotheses for learners to consider. For instance, the hypothesis

$$\texttt{F0}(x) := \texttt{pair}(\texttt{if}(\texttt{flip}(1/3), \epsilon, \texttt{F0}(\epsilon)), a)$$

concatenates (pairs) an "a" with either an empty string (ϵ) or the outcome of calling itself with an empty string as an argument, $\texttt{F0}(\epsilon)$. When called with the default argument of an empty string (i.e., $x = \epsilon$), `F0` generates the set of strings $\{a, aa, aaa, aaaa, \ldots\} = \{a^n : n = 1, 2, 3, \ldots\}$. Importantly, this function also gives a probability distribution over strings through its use of flip. Here, the distribution is geometric, meaning that the length of each output is determined by the number of times a 1/3-weighted coin can be flipped before getting heads, giving $P(a^n) = (1 - \frac{1}{3})^{n-1} \cdot 1/3$.

If we define H to be the space of all functions that can be constructed by composing the operations in Table 1 and define a variable D to be a multiset of observed strings, an idealized learner will compute a posterior distribution on hypotheses $P(H \mid D)$ via Bayes rule $P(H \mid D) \propto P(H) \cdot P(D \mid H)$. Here, $P(H)$ is given by a probabilistic context-free grammar (PFG) on the operations in Table 1, which effectively penalizes complexity. Thus, $P(H)$ implements a simplicity preference just as in induction of Turing machines (28–31), although the question of precisely which simplicity measure is psychologically accurate is an empirical one (63), and one which has been examined in closely related domains (59). $P(D \mid H)$ is a likelihood specifying how likely the observed strings are to be generated by H, as discussed in the next section.

Technical Innovations

Our implementation includes several important technical innovations that allow it to be scaled to interesting classes of formal languages. First, the choice of likelihood function $P(D \mid H)$ is chosen to allow incremental improvements to hypotheses. In its simplest form, we might just run the program H and use its output set of strings. The problem with this is that it does not assign any partial credit to hypotheses which get most pieces of a string correct. For instance, if a hypothesis generates the strings $\{a, aa, aaa, \ldots\}$, this likelihood would assign zero probability to observed data $\{ab, aab, aaab, \ldots\}$ even though most characters in most strings were produced. To address this, we apply a "prefix likelihood" which assumes a noise process that may delete and then append on the end of a string in a stochastic manner where each deletion or generation happens with a fixed probability. Thus, if a hypothesis generates *aaa*, it will assign *aaab* a nonzero likelihood equal to the probability of appending one *b*.

Second, we allow the possibility that hypotheses involve multiple LOT expressions (101), here called "factors" `F0`, `F1`, `F2`, etc. For example, a hypothesis might be

$$\begin{aligned}
&\texttt{F0}(x) := \texttt{pair}(\texttt{if}(\texttt{flip}(1/3), \epsilon, \texttt{F0}(\epsilon)), a).\\
&\texttt{F1}(x) := \texttt{if}(\texttt{empty}(x),\\
&\qquad\qquad \epsilon,\\
&\qquad\qquad \texttt{append}(\texttt{pair}(\epsilon, \texttt{first}(x)), \texttt{pair}(\texttt{F1}(\texttt{rest}(x)), b))).\\
&\texttt{F2}(x) := \texttt{F1}(\texttt{F0}(\epsilon))
\end{aligned}$$

The first function, `F0`, is the a^n formal language shown above. The second function, `F1`, takes an argument *x*, and recursively concatenates the first element of *x* with a recursive call to itself, followed by a *b*. Because this is recursive on a shortening string ($\texttt{rest}(x)$), one *b* will be added for each element of *x*. For instance, if we called `F2` on the string *xyz*, it would return the string *xyzbbb*. Finally, `F2` puts `F0` and `F1` together. Thus, the strings a^n generated by `F0` will be passed to `F1`, which will attach a single *b* for each *a*. This therefore generates strings of the form $a^n b^n$ (although it is not the simplest way). Allowing for multiple factors, in conjunction with the likelihood, allows for complex computations to be learned via individual pieces which are more manageable.

As described above, the expressions we evaluate may be nondeterministic. This provides a challenge for evaluating the set of strings that a given expression generates. Our implementation handles randomness by following multiple possible execution paths when a random primitive is encountered, enumerating the possible paths greedily in order of their probability, an idea drawing on techniques for evaluation of probabilistic programs more generally (102). We then compute the overall distribution of output strings, marginalizing approximately over execution paths. We use stochastic sampling methods to find high posterior probability hypotheses (see *Methods*). To do this, we developed a C++ library called Fleet (distributed under the GNU Public License v3 at https://github.com/piantado/Fleet). Notably, the Fleet implementation includes examples in other domains outside language, such as number learning (73) and logical rule induction (58), demonstrating the generality of the approach.

Results

We first study the model's ability to learn probabilistic versions of several formal languages which have been motivated by the structures present in natural language. We choose the targets of learning primarily following examples from reviews and theoretical pieces (67, 72, 103), while adding a number of other interesting examples. Our intention is not to engage the debate about which examples correspond to natural language; rather, we use these to motivate the range of formal systems that a learner is likely able to acquire. In all cases, we provide the learning model with positive examples only of the formal language. Moreover, for all formal languages and sections below, the inferential setup is identical between these languages—the only things that change are the data provided, the alphabet of symbols the model is expected to work over, and the amount of run time.

We visualize results by approximating the posterior model-average precision and recall for the most likely strings generated by each hypothesis (see *Methods*). When precision is high but recall is low, the model undergeneralizes (the strings it generates are all in the target formal language, but it does not capture all of them). Conversely, when recall is high and precision is low, the model overgeneralizes. When precision and recall are both 1.0, that means that the model has successfully learned the target formal language up to the approximation used to compute precision and recall. We also include a gray "Memorized (F)" line corresponding to what F score would be achieved by simply memorizing the observed data.

Learning Simple Formal Languages. Fig. 1 first shows learning curves for 56 different simple formal languages. Each plot shows the estimated precision and recall (*y* axis) as a function of the

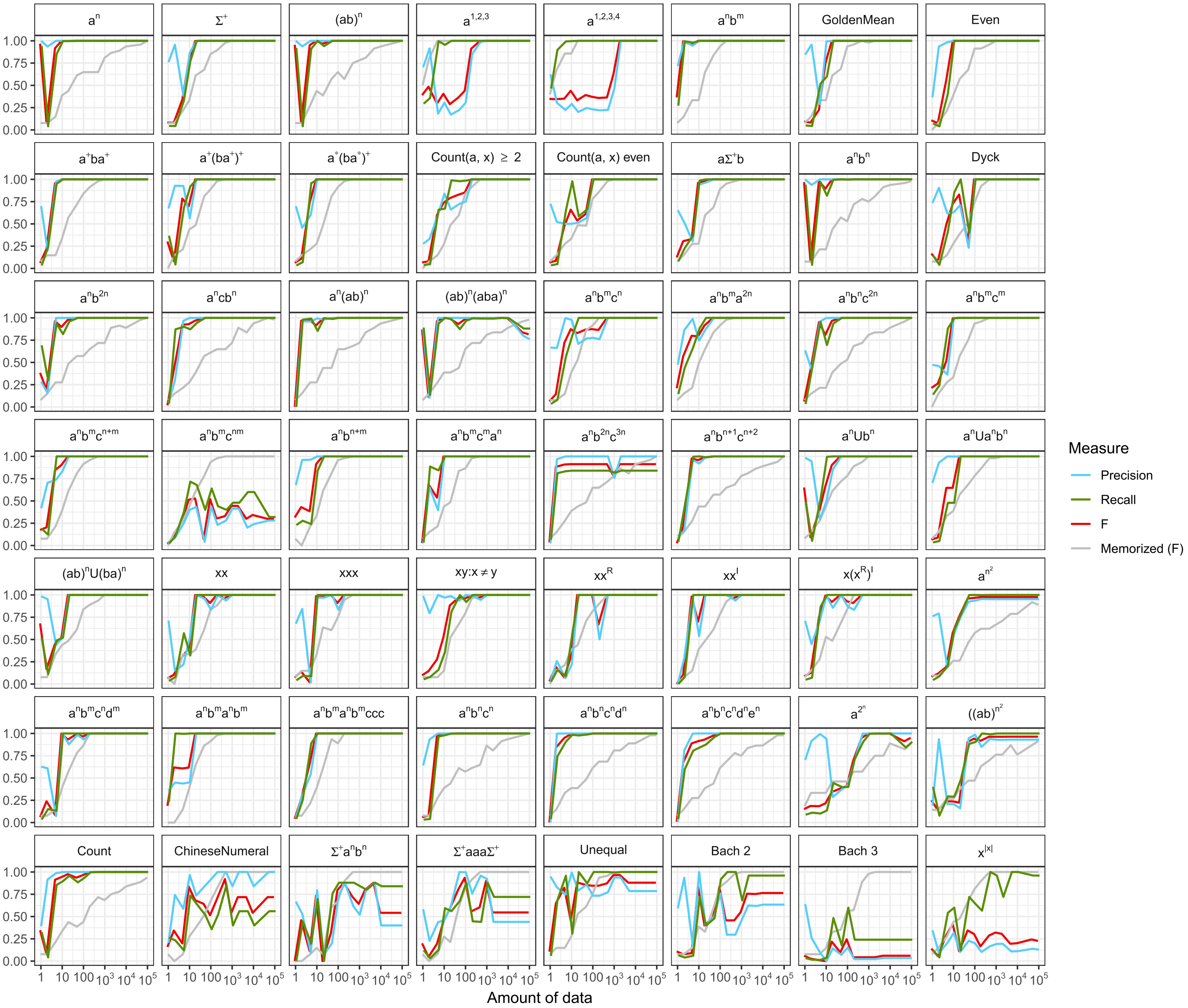


Fig. 1. Learning curves for the model, provided with data from each target formal language. Precision (blue), Recall (green), and F score (red) are for the learning model, while Memorized (F) (gray) gives the F score for a model that only memorized the training data. Here, we have adopted the convention that $\Sigma = \{a, b\}$, $n, m \in \mathbb{N}$, $Count(v, x)$ counts the number of times the character v occurs in string x, and $|w|$ is the number of characters in a string w. The x axis is the number of sampled tokens from the target formal language (note the logarithmic scale).

amount of data provided to the model (x axis) (see *Methods*). In general, determining equivalence of hypotheses is uncomputable, so our precision and recall measures provide a quantification of how well the learner has acquired the target language, but we caution that, in all our figures, the results must be interpreted with care because these values are approximate.

The model is able to learn many types of formal languages. Fig. 1 shows that the implemented model is able to learn programs for formal languages that vary in computational complexity, including regular (e.g., a^n, $(ab)^n$, and Σ^+), context-free (e.g., a^nb^n, a^nb^{n+m}, and a^nb^{2n}), and context-sensitive (e.g., $a^nb^nc^n$, $a^nb^nc^{2n}$, and $a^nb^{n+1}c^{n+2}$). It succeeds on the Dyck language, consisting of balanced sequences of parentheses—valid parse trees—and the "mirror" language xx^R. Both are context-free, and versions of them have been examined in recent experimental work (104, 105). The model's success across such diverse formal languages shows that techniques for program induction can work across levels of computational complexity.

Researchers have also explored formal languages that can be captured with other formalisms, including linear indexed grammars (106), tree-adjoining grammars (107) or combinatory categorial grammars (50). Formal languages relevant to these additional classes are learnable, such as $a^nb^nc^nd^n$. Several other interesting examples are also shown (GoldenMean, $Count(a, x) \geq 2$, $Count(a, x)$ is even, a^+ba^+, $(a^*)(ba^*)^+$, $(a^+)(ba^+)^+$), including those which distinguish other classes (103).[†] The formal language $a^nb^ma^nb^mccc$ is an example from ref. 64; the ChineseNumeral language $\{ab^{n_1}ab^{n_2}\ldots ab^{n_k} : n_i > n_{n+1}\}$ has been discussed previously as a model of numerals in Chinese (109, 110) and a motivating example for range concatenation grammars. This formal language is only learned

[†] The GoldenMean language—strings over $\Sigma = \{a, b\}$ with no two adjacent *a*s—is a classic example in symbolic dynamics. It contains a Fibonacci number of strings of each length and has a symbolic dynamic entropy equaling the log of the Golden Mean (108).

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

approximately. The model learns $a^n b^m c^n d^m$, which was motivated by crossed-serial dependencies in natural language (69, 111–113). The formal language $xx = \{xx : x \in \Sigma^+\}$ is motivated by "respectively" examples (114, 115) where lists must be paired up 1–1, as in "Bob, Pietro, and Johnny are a guitarist, accordionist, and troubadour, respectively" as well as examples from Mohawk's use of noun stems in verbs (116). Another interesting example is the Count language $\{ab, ababb, ababbabbb, \ldots\}$, which shows that other computations—perhaps unlike those needed in natural language—can be inferred in the same setup.

The hypotheses the model constructs reflect importantly different underlying processes. Examination of the specific hypotheses that the model constructs shows that it develops a generative model of the data, rather than simply memorizing patterns it sees. To illustrate a few examples, the model learns to construct a system equivalent to a context-free grammar when given strings from $a^n b^n$,

$$\texttt{F0}(x) := \texttt{append}(\texttt{pair}(\epsilon, a), \texttt{pair}(\texttt{if}(\texttt{flip}(1/3), x, \texttt{F0}(\epsilon)), b)).$$

This puts an a at the start of a string, puts a b at the end, and flips a coin for whether to add a recursive call in between the two. Not only does this generate the right set of strings $\{ab, aabb, aaabbb, \ldots\}$, it does so with the correct probabilities. Moreover, following the trace of recursive calls to F0 in generating a string reveals that the execution path of this program is tantamount to a parse tree,

$$[a\ b], \quad [a\ [a\ b]\ b], \quad [a\ [a\ [a\ b]\ b]\ b], \quad \ldots. \qquad \textbf{[1]}$$

In cases like this, the learner acquires a structure matching what a grammar would generate, although note that, in general, our evaluation metric evaluates only the generated string set, not its latent structure. The model discovers a clever and nonobvious way to express $a^n b^n c^n$:

$$\begin{aligned}\texttt{F0}(x) := \texttt{append}(&\texttt{pair}(\epsilon, a), \texttt{pair}(\texttt{if}(\texttt{flip}(1/3), \texttt{pair}(x, b),\\ &\texttt{Fm0}(\texttt{pair}(x, b))), c)).\end{aligned}$$

SI Appendix contains listings of the top hypotheses found at the end of learning for each of the languages tested, as well as the observed number of data tokens. In most cases, the best hypotheses correspond to ones that can be seen to correctly compute each language, or come very close, showing that the model is genuinely discovering appropriate generative processes.

Learning takes little data, and the model does more than memorize. Note that, in most cases, the amount of data required for learning to succeed is surprisingly small—often within less than 10 sentence tokens sampled from the target language. This accords with the small estimates of the total information required for syntax (117) and highlights that, even though the hypothesis space is large, that doesn't mean that huge amounts of data will be required. Intuitively, since each observed string is unlikely under most hypotheses, only a few strings are enough to reduce the set of likely hypotheses to a manageable number. Moreover, in most cases where the model is able to learn, it learns much faster than a model which simply memorized the data, as shown by the model F score (red) generally being above the memorized F score (gray). For example, the model acquires a nearly perfect F score on $a^n b^n c^n$ after about 10 tokens, but it takes around 10^5 tokens for memorization to achieve a similar level. We note that the slow speed of memorization results from how long it takes to sample the top 25 strings under the assumed distribution on string lengths. This comparison highlights that the model generalizes far beyond the data it has seen.

The model eagerly generalizes finite data to infinite string sets. An interesting comparison can be made between the formal language a^n and $a^{1,2,3} = \{a, aa, aaa\}$. The latter is a subset of the former, but it has a finite cardinality. One of the most striking properties of natural language is that English grammar seems to permit arbitrarily long sentences (cf. ref. 118). This fact might be considered to be a core aspect of our innate linguistic endowment (119, 120) or a consequence of evolving a communication system with many signals (121). Alternatively, learners might even consider as statistical hypotheses that language was finite or infinite, as suggested by the purportedly finite languages in existence like Pirahã (36, 37). It might seem that there could be no data to show a learner that their language was infinite because that hypothesis necessarily goes beyond what has been observed. Indeed, infinite generalization is contrary to subset-principle accounts (6, 122–124) that posit learners make only the narrowest generalization possible from data (cf. refs. 125 and 126). Here, however, the issue comes down to whether finite or infinite languages are easier to express. The model shows that a^n is easier to learn than $a^{1,2,3}$ or $a^{1,2,3,4}$, and this is because the infinite language has a simpler description, matching prior theoretical analysis (127). The model's success in learning infinitely productive generative systems of rules can be contrasted with claims in generative linguistic textbooks. For instance, one states that "a productive system like the rules of Language probably could not be learned or acquired. Infinite systems are in principle, given certain assumptions, both unlearnable and unacquirable" (128).

We note that all of the learning results use data which are sampled from the target formal language, typically using geometrically distributed string lengths. In principle, however, this kind of model allows us to examine how many—or which—individual data points at different embedding depths license learners to infer that the language is infinite. Detailed investigation of these kinds of learning patterns should help to refine debates about the possible role of rare data in natural language acquisition, as even a few sentences can lead to unbounded productivity.

Not all formal languages are easily learned, even though the model is Turing complete. Finally, it is often argued that models which have a capacity to learn any formal language are inappropriate for human language because humans appear unwilling to learn some patterns. But, even for unconstrained models like this one, the limitations of inference, the strength of priors, and the informativity of data make some languages effectively unlearnable. For instance, this model has difficulty with the Bach 3 formal language, which consists of all strings in $\{a, b, c\}^+$ with an equal number of *a*s, *b*s, and *c*s (129). This language might be more easily learnable with other operations like "scrambling" (130) but is difficult to express with our assumed primitives. The formal language $x^{|x|}$ isn't learned, and others like a^{2^n} can be learned only approximately even with large amounts of data. Which languages are difficult to learn is determined by a subtle combination of the available primitives, the assumed inferential biases, and the way in which the data distinguish close alternatives.

A related consideration in linguistics is whether learning models can acquire patterns which are not typologically attested. Of course, any model like this that operates over an infinite hypothesis space must acquire unattested languages, since there are only finitely many attested languages. Importantly, however, we do not take the model as making typological predictions, mainly because there are many other pressures that factor into the form of languages beyond structural biases, including considerations of communicative usefulness (131, 132). To illustrate, the formal language computed by $\texttt{if}(\texttt{flip}(), a, b)$ is high probability in the prior (since it is short) and would be easy for the model to learn. But this language only contains two symbols and therefore cannot be used to communicate much information. The model does not predict that such small languages with just two sentences should exist in the natural sample of languages. Instead, it claims

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

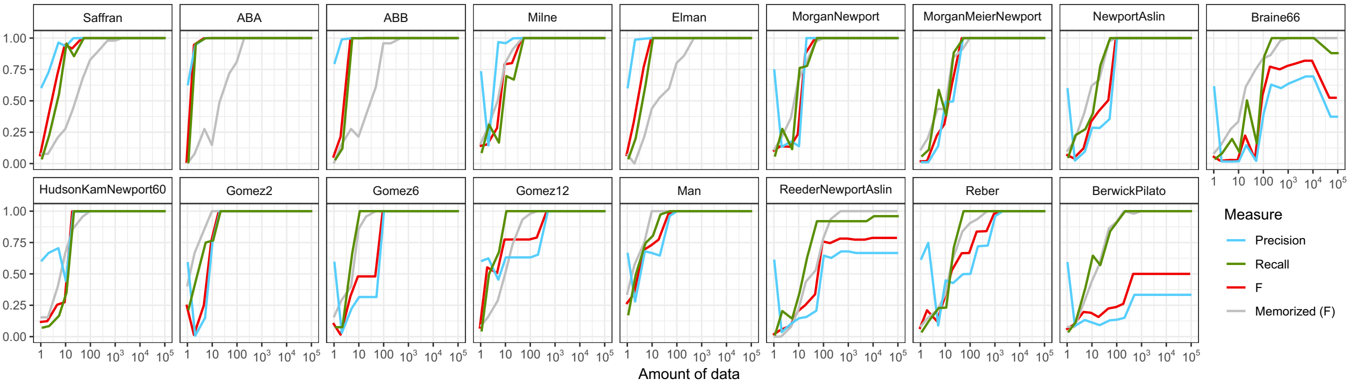


Fig. 2. Learning curves for the model when run on examples from artificial language learning, including the Finite State grammar from Reber (149), a version of word segmentation motivated by Saffran et al. (134), "ABA" and "ABB" grammars from Marcus et al. (139), context-free grammars from Newport and coworkers (141, 146), a grammar from Morgan et al. (142), a finite set of strings consisting of all valid English words with "m," "a," and" n" (e.g., "man, am, an, a, mam"), a finite state machine for English auxiliaries from Berwick and Pilato (150), the QAXB grammar from Reeder et al. (154), a grammar modified from Hudson Kam and Newport (219), a finite language from Braine (147), a version of lexical learning from ref. 190, and string sets analogous to Gómez (144).

that people could learn a two-sentence formal language if given enough input consistent with it.

Connections to Artificial Language Learning. We next study simplified versions of languages that have been examined in artificial language learning studies. Experimental work has characterized specific biases and limitations in real biological learners, and it is not our intention to show how these can be built into the model. Instead, our goal is to articulate a coherent computational-level (133) framework for learning onto which one can later add specific limitations and abilities documented in empirical studies.

One of the most well-known artificial language paradigms, by Saffran et al. (134, 135), examines the ability of adults and children to segment a continuous speech stream into words. We provided the learning model with data modeled after ref. 134, with the primary change that the distribution of string lengths was geometric. This was necessary in order to avoid having to deal with hypotheses that generate infinite sequences (which are possible, although more challenging technically), and also may better match the observation that utterance boundaries provide an important cue in segmentation (136). Results are shown in Fig. 2 and illustrate that the model is able to create representations of the regularities in Saffran et al.'s stimuli. Notably, even though the model, like people, is provided only with an unsegmented stream of syllables, it is able to construct the appropriate recursive calls to generate these syllables as a stream of words. For instance, it learns to generate a word "tapiro" by constructing an internal structure $\mathtt{pair}(\mathtt{pair}(\mathtt{pair}(\epsilon, ta), pi), ro)$. The model succeeds even though it is not told to search for words, nor is it given special cues like transitional probabilities. Words are learned simply as an efficient way of statistically "compressing" the observed data, in line with, for example, ref. 137. Such success shows how processes like segmentation, lexical learning, and syntactic category learning may all fall under the same umbrella (138).

Marcus et al. (139) studied infants' ability to learn abstract variables in the form of languages that followed an "ABA" pattern (e.g., do–re–do) vs. an "ABB" pattern (e.g., do–re–re). The relevant feature of these languages is that "A" and "B" are variables which get realized as specific syllables. Infants' success in learning these patterns suggests that the role of variables may be more broadly central to cognition (78, 140). Fig. 2 shows that the learning model is capable of learning these patterns by using the appropriate combination of function calls and variables. The representation learned in this case is

$$\mathtt{F0}(x) := \mathtt{append}(\mathtt{append}(\mathtt{sample}(\Sigma), x), x).$$
$$\mathtt{F1}(x) := \mathtt{Fm0}(\mathtt{sample}(\Sigma)).$$

This hypothesis explicitly captures the ABB pattern in its structure.[‡]

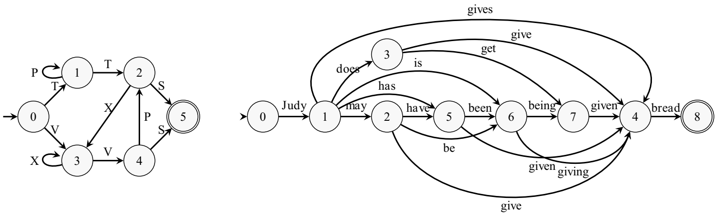
"Morgan & Newport" (141) follows simple phrase structure grammar in Fig. 3, and has been examined in several empirical studies (142, 143). In this grammar, elements in parentheses are optional. "Morgan, Meier, & Newport" is a version from ref. 142 with "function words" (o, a, u, i) that mark the syntactic grouping and speed learning. Fig. 2 shows that both versions of the language are learnable with very little data (~ 100 tokens), although note that, despite being presented as grammars, these are finite languages and are not learned faster than memorization. Work by Gómez (144) and Gómez and Maye (145) has examined languages of the form aXb where the number of possible X elements varied. In Fig. 2, several "Gómez" curves are shown, corresponding to languages with varying cardinality of 2, 6, and 12. As this makes clear, these languages are learnable; similarly, the model is also able to learn the nonadjacent dependencies studied by Newport and Aslin (146), consisting of sentences of the form $\{bXt, gXd, pXr, kXu, lXi\}$ where X ranges over $\{1, 2, 3, 4\}$. The model has some difficulty learning the simple finite grammar from Braine (147, 148) that mimicked some properties of serial order and phrase structure rules.

Fig. 2 also shows learning curves for finite state languages examined by Reber (149) and Berwick and Pilato (150), which are described in Fig. 3. These languages are interesting, in part, because they are fairly complex in terms of description length, yet are simple—finite state—in computational complexity. Reber's was constructed to provide a learnable but nontrivial set of strings, and both it and similar languages have been modeled with neural networks (151, 152), as well as models based on chunking (153). Berwick and Pilato's captures the English auxiliary system, and the model can only approximate it, likely due to the complexity of this language and the model's limited inference scheme.

Reeder et al. (154, table 2) is an interesting case where some strings are held out from the language (and the human training). Learners and the model are given a "QAXB" pattern and then tested on unseen strings. This means that, for the model to perform well, it should exhibit a high recall (correctly reproducing the test strings) and a lower precision (generating strings outside of the training set). It does this, and generates, for instance, many strings like "scb" and "axB" which are not in the training set but do fit the intended pattern of the training data.

[‡] We did not require $A \neq B$ in this implementation.

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.



Morgan & Newport

S → AP BP (CP)
AP → a (d)
BP → CP f | e
CP → c (g)

Morgan, Meier, & Newport

S → AP BP (CP)
AP → o a (d)
BP → a CP f | u e
CP → i c (g).

Fig. 3. Finite state machines from Reber (149) (*Left*) and Berwick and Pilato (150) (*Center*). Words in this latter machine were converted to single characters to simplify the learning model. Grammars are from Morgan and Newport (141) and Morgan et al. (142) (*Right*).

Overall, these results demonstrate that the model is capable of learning many of the patterns examined in experimental work. We expect that the model will not provide a tight fit to the details of human behavior in such tasks without incorporating additional cognitive considerations, including memory limits and perceptual salience.

Toward Natural Language Grammars. The problem faced in natural syntax is not just a series of unrelated, simple languages but that of learning a language that combines many patterns simultaneously. We next examine an English-like grammar that one might find in an introductory linguistic textbook,

$$\begin{aligned} S &\to NP\ VP \\ NP &\to n \mid d\ n \mid d\ AP\ n \mid NP\ PP \\ AP &\to a \mid a\ AP \\ VP &\to v \mid v\ NP \mid v\ t\ S \mid VP\ PP \\ PP &\to p\ NP \end{aligned} \quad . \qquad [2]$$

This target grammar includes several syntactic structures of English, including multiple expansions of a nonterminal type (e.g., NP), tail recursion (in AP), transitive and intransitive verbs (in VP), sentential embedding (in VP), and prepositional phrases with embedded noun phrases. For instance, one string the model might observe is “d a n p d a n v t n v,” which corresponds to the part-of-speech sequence for a sentence like “The delightful accordionist by the old cafe knew that music heals.” As with all datasets, the model is not told there is a (e.g., context-free) grammar to learn, but is only told to construct something out of its primitives to explain the observed strings.

The model’s learning curve for this grammar is shown in Fig. 4. The model is able to learn a program that closely approximates the target grammar after seeing a few hundred sampled string tokens, showing that the learning model is capable of rapidly learning the required patterns.

Fig. 5 visualizes the space of generalizations after observing 200 tokens. Each row shows a different hypothesis, and each column is a string, with cells colored according to the log probability that hypothesis assigns to each string. This figure shows only strings which are not seen in the data and therefore provides a view of how the model generalizes to rare data, a key consideration in many linguistic theories. First, the space of generalizations is complicated: The space of high probability strings does not vary smoothly with hypotheses. Sometimes, hypotheses which are close in posterior probability exhibit very subtly different patterns of generalization; sometimes, hypotheses which agree for many strings differ in other less frequent strings. This shows that the task of selecting exactly a parent’s grammar must be difficult, because top hypotheses on this dataset are only distinguished on rare strings. Additionally, because the space of computations is highly structured, learners will exhibit nontrivial beliefs about constructions they have never seen—even in the absence of language-specific constraints.

Discussion

Although “poverty of the stimulus” arguments are often considered classic and foundational in linguistic theorizing (3, 155–159), prior reviews (35, 160–162) and theoretical results (28) have been devastating to the hypothesis that learners must be highly constrained in order to ensure that language can be learned. The model developed here provides a concrete alternative to pessimism about the power of learning. It acquires generative systems from relatively small amounts of positive data, working out of a broad class of computations. Ongoing developments in program induction models—such as the ability to reuse components (101), evaluate parital hypotheses (163), work in continuous spaces (164), and apply rich toolkits of programming methods (38)—are likely to help such learning scale.

Learners who operate over relatively unrestricted spaces show many hallmarks which have traditionally been attributed to universal grammar (UG), a domain-specific genetic endowment hypothesized to support language acquisition. Pullum and Scholz (161) identify and reject a number of common claims about language acquisition that have been used to support UG, including the speed with which children learn, the fact that children acquire a system that can deal with essentially unbounded sentences, the fact that their data underdetermine their grammar, and that different individuals in the same community learn similar grammars. The model provides an implementation which also shows such properties—quickly and reliably learning computations from underdetermined evidence—while operating in the most unrestricted space possible. Other authors have emphasized, in support of UG, that children’s knowledge of language

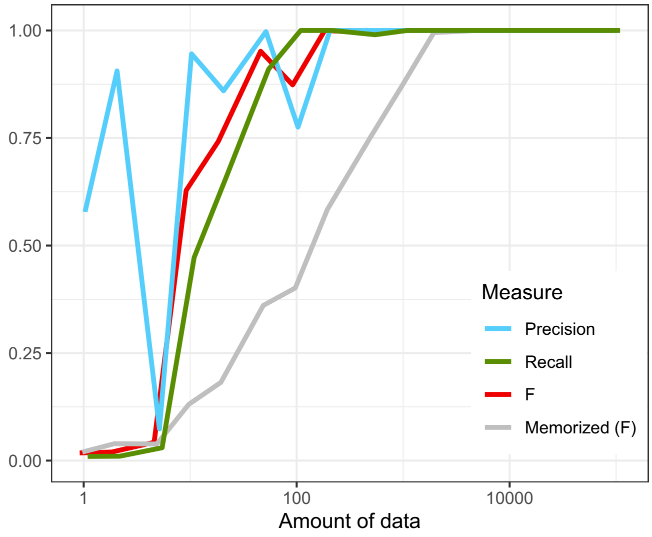


Fig. 4. Learning curves for the simplified English grammar (2). Note that here, precision and recall was computed on the top 100 strings. This language is learned to high accuracy considerably faster than memorization.

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

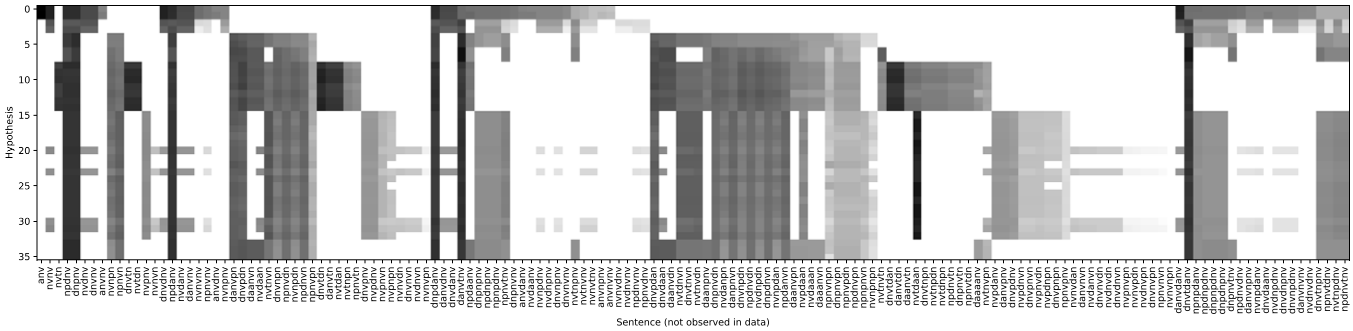


Fig. 5. A visualization of the generalizations by the learning model for the simple English grammar, after observing 200 data points. This shows the posterior probability (greyscale) assigned to strings not observed in the data (columns) by each hypothesis (row). The rows are sorted with the highest posterior probability at the top, and duplicate rows with identical generalizations patterns have been removed. The pattern of generalizations is complex, structured (e.g., with correlations apparent between strings), and highly dependent on which hypothesis is selected, even among those with close posterior probabilities.

goes beyond what they directly observe. The present work highlights one sense in which that fact might be unremarkable: Any learning system that works over a sufficiently rich space of computations will appear to know about strings that have not been observed. This can be seen in even the simplest case of a^n, but the point is most clear for grammars which yield complex patterns of generalizations.

A huge body of literature has been dedicated to articulating and critiquing the subset problem of how it is that learners who use only positive evidence can avoid arriving at an overly general grammar (122, 125, 159, 165–170). If a learner hypothesizes that an ungrammatical string is actually grammatical, they will never receive directly contradictory evidence from positive examples. One solution to this can be found in the present model's use of distributions, ideas which date back to ref. 22 (see ref. 171) and have a simple basis in probability theory (172): Since probabilities must sum to one, hypotheses that use some probability for unseen strings will necessarily have less probability for the data that are observed, lowering the hypothesis' likelihood. This fact also interacts with the structural priors, which sometimes favor hypothesizing that unobserved strings are actually grammatical because this leads to a simpler computation.

It is worth emphasizing that the assumptions of no negative evidence (173) have been questioned by many language acquisition researchers (161, 174–178). Our general model could easily be made compatible with positive evidence, or even other forms, including pedagogical evidence, as has been examined in similar statistical setups (179). However, even if children do receive substantially informative negative evidence, they still face a deep inductive problem. To choose a simple example, the formal language a^n is infinite, and yet any data a learner sees—positive or negative—will be finite. Apart from explicit metalinguistic instruction, there does not seem to be any form of negative evidence or rephrasing that could lead learners to deduce this fact directly. Instead, learners must have an inductive mechanism that can consider the infinite language as a possibility, and weigh it against alternatives. We suspect that many structural aspects in language acquisition will have this flavor, where the precise form of evidence is not as crucial as the inductive biases learners possess and the hypotheses they consider.

Our results demonstrate that positive evidence not only can work in theory for discovering the right computation, but it works in practice, using simple, domain-general search techniques that have been applied broadly across cognitive science. At the same time, these methods are still in their infancy. We have only applied them to some of the basic descriptions of linguistic phenomena—ideas like recursion, nesting, simple context-free phrase structure, and context-sensitive dependencies. This work leaves open the question of how more-detailed kinds of phenomena may be acquired—for instance, long-distance dependencies in syntactic islands (180, 181). We note, however, that many such phenomena have been hypothesized to result from UG, but may, in fact, be better explained as properties of constructions and pragmatics (182–184). Either way, the learning of such aspects of grammar have simply not yet been explored using powerful domain-general learning tools.

Our approach was motivated, in part, by Perfors et al. (27), who compared different grammars—for example, finite, finite state, and context-free—and showed that a small amount of child-directed speech from the CHILDES project (185) was sufficient for learners to infer that language had a hierarchical structure. One response to Perfors et al. has been to emphasize linguistic facts that go beyond the mere presence of hierarchical structure, often depending on subtle distinctions and connections between constructions or interpretations (157), although Bayesian models have shown success in learning some such phenomena (186). A more interesting—although we think ultimately misguided—critique of Perfors et al. is that their model "builds in" more than UG because learners consider multiple grammars. Surely it is a more parsimonious theory of human nature that learners come equipped with one single grammar than the seven Perfors et al. used? The argument also applies to our model: How could "building in" an infinite number of possible computations be a more parsimonious solution to the problem of language learning?

It turns out that this thinking is wrong, for a very interesting reason. The issue can best be understood by considering Jorge Luis Borges' short story, "The Library of Babel" (187). Borges imagines an infinite library full of every possible book—every possible sequence of characters. The curious fact about the library which contains every possible book is that it actually contains essentially no information at all. Its contents can be completely specified with just a few words ("all possible books"), or compressed into an extremely short generating computer program, yielding a negligible minimum description length.[§] Certainly, it is much easier to describe the entire library than to describe just one of its typical books. The story illustrates that it is often more parsimonious to build in a larger, unconstrained space of hypotheses because a large space can easily have a more concise description. Theories which hypothesize that minimal amounts of structure and content are innate should embrace this perspective.

Our results also address a critique of domain-general approaches to language learning: It is often argued that children must possess constrained sets of hypotheses because there are

[§] For example, $\texttt{F0}(x) := \texttt{pair}(\texttt{F0}(x), \texttt{sample}(\Sigma))$.

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

many constructions that children simply don't learn. However, our results also showed that there are languages that are difficult to learn, even for a universal model. This fact is not remarkable—it is inevitable. Only a few languages can have short description lengths or equivalently high priors, and so almost all logically possible languages will be difficult to learn. Consequently, there simply must be logically possible generalizations children never make, regardless of whether there are constraints specific to language or not. Learnability analyses have too often ignored these issues and conflated the class which can practically be learned with the total hypothesis space (160).

This model tackles acquisition of language from a different perspective than connectionist models (188, 189), which often are taken as contrary to poverty of the stimulus claims due to their ability to acquire key aspects of language as well as model human performance on learning tasks (152, 190–196). Connectionist approaches have seen remarkable success in recent years in natural language tasks (197–199), including hierarchical languages (200). The datasets studied here may provide a compelling testbed for such neural network models, and are closely related to Turing-complete neural architectures (164).

Finally, our results suggest that the Chomsky–Schützenberger hierarchy—popular for characterizing human and animal communication (201, 202)—may not align with psychological notions of complexity. Because of the chosen form of the prior, the model's inferences are sensitive to description length rather than hierarchy level: Some languages that have short descriptions are higher on the hierarchy (like $a^n b^n c^n$), and some languages (like Reber) that are lower on the hierarchy are nonetheless difficult because they do not have concise descriptions as programs. Theories of language acquisition would do well to prioritize measures of description length (203, 204) in considering the complexity of hypotheses, bringing such theories in line with experimental work in human learning (59, 88, 131, 205–207).

These results also point to several important directions for future modeling, experimentation, and theoretical work. Many aspects of learning—from words to grammars—may primarily require learners to find concise descriptions of observed data (29, 31, 204, 205). While some languages like the simple English grammar presuppose that categories are known, some of the formal languages—even simple ones like $a^n b^n$—have the model implicitly discover that different symbols (a and b) have different distributional properties. The goal of finding concise computational descriptions can likely be applied to other levels of linguistic representation, including part of speech categories or phonemes, where the relevant abstractions are thought to provide simple descriptions of apparent patterns. Generally, however, this work leaves open the question of what domain-specific constraints and knowledge are present in each of these levels of linguistic analysis and how those interface with domain-general capacities. At the very least, claims of domain-specific knowledge will only be tenable when formulated relative to demonstrable shortcomings in implemented domain-general learning theories.

While this model provides a theoretical proposal for learning the patterns that govern sequences of discrete symbols, it is important to remember that real-life language learning is closely tied to language use (208–210), including semantics, pragmatics, and social inference. None of these are captured in the present framework, which was designed to study a simplified setting akin to Gold's. Although these other processes are central to how children learn language, the present work does suggest that there is unlikely to be an in principle learnability problem even without them. One additional limitation of our work is that it is not clear how to extend such models to a mental lexicon that contains thousands of lexical items—or, more specifically, how induction of structure interfaces with humans' distinctive memory architecture. Relatedly, this learning model does not learn any meanings associated with strings. However, similar program-learning models have been used to acquire compositional representations of semantics (211–214) while operating over larger vocabularies. These models illustrate one way in which program-like learning models may be extended to richer settings.

Conclusion

The model described in this paper shows how learners could begin with a simple set of domain-general computational operations and create grammars in order to explain observed data. The model shows that implemented program induction techniques can build representations of provably different computational power, including regular, context-free, and context-sensitive hypotheses. It also shows that such learning requires surprisingly little data, and that positive evidence is sufficient. This model provides an inferential foundation onto which psychological or linguistic constraints—perhaps including pressures in memory or computational limitations (215)—can be added in order to refine debates about what resources learners must necessarily bring to language acquisition. Notably, in this approach, grammars are just one kind of computation that learners might acquire, and the model's key assumptions and mechanisms have been independently argued to explain nonlinguistic learning in other domains. The model thus points to how language acquisition might be unified with learning more broadly in cognition, including the many domains where children also acquire structures and abstraction from statistical evidence (216).

Methods

In all formal languages, words and syllables in prior literature have been reduced to single characters so that the primary components that are generated are sequences of characters in a fixed alphabet. Languages are made probabilistic using generative (`flip`) parameters that prevent data generation from creating strings that are too long or recursions that are too deep. Note that, in evaluating a hypothesis, the initial value for x that is passed in is the empty string (ϵ); however, when one factor is called by another, it may pass in other arguments. In running, we enumerate execution paths down to a log probability of -15, 64 recursions, 1,024 program steps, or 256 different outputs. For the English example, we increased these bounds to output a larger distribution of strings. All of the code for specifying and sampling the input data and the input data themselves are available at the Fleet GitHub.

Inference was run using a custom implementation of the adaptive parallel tempering scheme from ref. 217 in Fleet, using proposals from ref. 58. This ran five temperatures from 1 to 1.2, spaced exponentially. The inference proposed swaps between chains every 0.25 s and adapting the temperature ladder every 5 s. This inference scheme was run on data consisting of 1, 2, 5, 10, 20, 50, 100, 200, 500, 1,000, 2,000, 5,000, 10,000, 50,000, and 100,000 strings sampled from the target grammar. This inference was run on times varying from 1 min to 7 d total, across all amounts of data (evenly divided). The specific times each language was run for are provided in *SI Appendix*. Note that, in general, it is difficult to search over factors, since changes to one factor may depend on what another factor computes. We therefore run multiple parallel searches while constraining the number of factors $n = 1, 2, 3, 4$ within each search. For instance, on the search with three factors, we reject any proposal that fails to call all three factors, although we note that this still often results in trivial factors. We ran this inference on a collection of Dell servers for varying amounts of time using GNU Parallel (218). We collected the 500 highest posterior hypotheses found in each chain at each amount of data and used this to plot the posterior-weighted F curves shown in figures. We note that such approximation and sampling is not intended as part of the high-level interpretation of our approach. We intend the model to be on Marr's computational level (133) in that people solve the same problem of inferring the algorithm that generates strings; we do not claim that they necessarily use the same methods as our implementation.

For all languages but the simplified English grammar, we computed precision and recall using the most frequent 25 strings in the data and generated by each hypothesis. Let $S(h)$ be the set of strings from a hypothesis h and let $S(D)$ be the set of strings for a dataset D. Let $S_{25}(h)$ and $S_{25}(D)$ denote the most frequent (or high probability) 25 strings enumerated by the model and occurring in the data, respectively. Let $\hat{P}(h \mid D)$ denote the probability of h given the data, renormalized to sum to one over the top 500 hypotheses found (note that, since the hypothesis space is discrete, the

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

top 500 hypotheses contain virtually all of the posterior probability mass). Then, posterior-weighted precision is defined as

$$\text{precision} = \sum_{h \in H} \hat{P}(h \mid D) \frac{\mid S_{25}(h) \cap S(D) \mid}{\mid S_{25}(h) \mid}, \qquad [3]$$

with recall defined analogously. Thus, the precision measures the proportion of the top 25 model strings that occur in the data, and the recall measures the proportion of the top 25 data strings that are output by the model.

Because the simplified English grammar was substantially more complex than other languages, it was run on factor sizes of 1, 2, 3, 4, 5, and 6, with 15 threads each, for 7 d. Precision and recall was computed on the top 100 strings instead of the top 25 in order to better assess learning. In addition, these runs permitted more steps and smaller log probability in order to enumerate more strings.

Several languages (e.g., Braine 66) introduced nonuniform sentence probabilities into the original grammar. This was done so that the strings yielded could be more easily evaluated in our metric of number of top strings. In this way, the tested languages were sometimes more skewed in distribution than the original references. While this makes the string set easier to identify by our metric, it also makes the distribution harder to match, since uniform distributions should be very easy for the model to learn.

Data Availability. The C++ library called Fleet, which we created, is distributed under the GNU Public License v3 at GitHub, https://github.com/piantado/Fleet.

ACKNOWLEDGMENTS. We are grateful to Yim Register, Frank Mollica, Sam Cheyette, Holly Palmeri, Josh Rule, András Kornai, and Ced Zhang for helpful comments and conversations about this work. This work was supported by Grant 1760874 from the NSF, Division of Research on Learning (to S.T.P.) and Award 1R01HD085996 from the Eunice Kennedy Shriver National Institute of Child Health & Human Development at the NIH (to S.T.P. and Jessica Cantlon).

1. E. Gold, Language identification in the limit. *Inf. Control* **10**, 447–474 (1967).
2. J. Hopcroft, R. Motwani, J. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Reading, MA, 1979), **vol. 3**.
3. N. Chomsky, *Aspects of the Theory of Syntax* (MIT Press, Cambridge, MA, 1965).
4. K. N. Wexler, H. Hamburger, "On the insufficiency of surface data for the learning of transformational languages" in *Approaches to Natural Language*, K.J.J. Hintikka, J.M.E. Moravcsik, P. Suppes, Eds. (Springer, 1973), pp. 167–179.
5. H. Hamburger, K. Wexler, A mathematical theory of learning transformational grammar. *J. Math. Psychol.* **12**, 137–177 (1975).
6. K. Wexler, P. Culicover, *Formal Principles of Language Acquisition* (MIT Press, Cambridge, MA, 1983).
7. P. Niyogi, R. C. Berwick, A language learning model for finite parameter spaces. *Cognition* **61**, 161–193 (1996).
8. D. N. Osherson, M. Stob, S. Weinstein, *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists* (MIT Press, 1986).
9. M. Kanazawa, Identification in the limit of categorial grammars. *J. Log. Lang. Inf.* **5**, 115–155 (1996).
10. F. Denis, Learning regular languages from simple positive examples. *Mach. Learn.* **44**, 37–66 (2001).
11. A. Clark, R. Eyraud, "Identification in the limit of substitutable context-free languages" in *International Conference on Algorithmic Learning Theory*, S. Jain, H.U. Simon, E. Tomita, Eds. (Springer, 2005), pp. 283–296.
12. A. Clark, Learning trees from strings: A strong learning algorithm for some context-free grammars. *J. Mach. Learn. Res.* **14**, 3537–3559 (2013).
13. A. Clark, S. Lappin, Computational learning theory and language acquisition. *Philos. Linguist.* **14**, 445–475 (2010).
14. A. Clark, "Towards general algorithms for grammatical inference" in *Proceedings of the 21st International Conference on Algorithmic Learning Theory*, M. Hutter, F. Stephan, V. Vovk, T. Zeugmann, Eds. (Springer, 2010), pp. 11–30.
15. A. Clark, "Three learnable models for the description of language" in *Language and Automata Theory and Applications: 4$^{\text{th}}$ International Conference*, A.-H. Dediu, H. Fernau, C. Martin-Vide, Eds. (Springer, 2010), pp. 16–31.
16. J. A. Feldman, J. Gips, J. J. Horning, S. Reder, "Grammatical complexity and inference" (Tech. Rep. CS 125, Department of Computer Science, Stanford University, Stanford, CA, 1969).
17. D. Angluin, "Finding patterns common to a set of strings" in *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, 1979), pp. 130–141.
18. D. Angluin, Inductive inference of formal languages from positive data. *Inf. Control* **45**, 117–135 (1980).
19. Y. Sakakibara, Recent advances of grammatical inference. *Theor. Comput. Sci.* **185**, 15–45 (1997).
20. C. De la Higuera, *Grammatical Inference: Learning Automata and Grammars* (Cambridge University Press, 2010).
21. L. G. Valiant, A theory of the learnable. *Commun. ACM* **27**, 1134–1142 (1984).
22. J. J. Horning, "A study of grammatical inference," PhD thesis, Stanford University, Stanford, CA (1969).
23. K. Johnson, Gold's theorem and cognitive science. *Philos. Sci.* **71**, 571–592 (2004).
24. A. Clark, S. Lappin, *Linguistic Nativism and the Poverty of the Stimulus* (John Wiley, 2010).
25. J. Feldman, Some decidability results on grammatical inference and complexity. *Inf. Control* **20**, 244–262 (1972).
26. D. Angluin, C. H. Smith, Inductive inference: Theory and methods. *ACM Comput. Surv.* **15**, 237–269 (1983).
27. A. Perfors, J. B. Tenenbaum, T. Regier, The learnability of abstract syntactic principles. *Cognition* **118**, 306–338 (2011).
28. N. Chater, P. Vitányi, Ideal learning of natural language: Positive results about learning from positive evidence. *J. Math. Psychol.* **51**, 135–163 (2007).
29. R. J. Solomonoff, A formal theory of inductive inference. Part I. *Inf. Control* **7**, 1–22 (1964).
30. R. J. Solomonoff, A formal theory of inductive inference. Part II. *Inf. Control* **7**, 224–254 (1964).
31. M. Hutter, *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability* (Springer Science & Business Media, 2005).
32. J. Schmidhuber, "Discovering solutions with low Kolmogorov complexity and high generalization capability" in *Machine Learning Proceedings 1995*, A. Prieditis, S. Russell, Eds. (Elsevier, 1995), pp. 488–496.
33. J. Schmidhuber, "Gödel machines: Fully self-referential optimal universal self-improvers" in *Artificial General Intelligence*, B. Goertzel, C. Pennachin, Eds. (Springer, 2007), pp. 199–226.
34. N. Evans, S. C. Levinson, The myth of language universals: Language diversity and its importance for cognitive science. *Behav. Brain Sci.* **32**, 429–448 (2009). Discussion in: *Behav. Brain Sci.* **32**, 448–494 (2009).
35. E. Dąbrowska, What exactly is Universal Grammar, and has anyone seen it? *Front. Psychol.* **6**, 852 (2015).
36. D. Everett, Cultural constraints on grammar and cognition in Pirahã: Another look at the design features of human language. *Curr. Anthropol.* **46**, 621–646 (2005).
37. R. Futrell, L. Stearns, D. L. Everett, S. T. Piantadosi, E. Gibson, A corpus investigation of syntactic embedding in Pirahã. *PLoS One* **11**, e0145289 (2016).
38. J. S. Rule, J. B. Tenenbaum, S. T. Piantadosi, The child as hacker. *Trends Cogn. Sci.* **24**, 900–915 (2020).
39. G. Miller, N. Chomsky, Finitary models of language users. *Handb. Math. Psychol.* **2**, 419–491 (1963).
40. R. Siegler, J. Shrager, "Strategy choices in addition and subtraction: How do children know what to do?" in *Origins of Cognitive Skills*, C. Sophian, Ed. (Lawrence Erlbaum Associates, 1984), pp. 229–293.
41. R. Siegler, E. Jenkins, *How Children Discover New Strategies* (Lawrence Erlbaum Associates, 1989).
42. J. Shrager, R. Siegler, Scads: A model of children's strategy choices and strategy discoveries. *Psychol. Sci.* **9**, 405–410 (1998).
43. R. M. Jones, K. Van Lehn, Acquisition of children's addition strategies: A model of impasse-free, knowledge-level learning. *Mach. Learn.* **16**, 11–36 (1994).
44. A. Gopnik, A. N. Meltzoff, P. K. Kuhl, *The Scientist in the Crib: What Early Learning Tells Us about the Mind* (Perennial, New York, NY, 2001).
45. P. Langley, G. L. Bradshaw, H. A. Simon, BACON.5: The discovery of conservation laws. *IJCAI* **81**, 121–126 (1981).
46. M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
47. J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (MIT Press, Cambridge, MA, 1992).
48. G. Boole, *An Investigation of the Laws of Thought: On Which Are Founded the Mathematical Theories of Logic and Probabilities* (Walton and Maberly, London, United Kingdom, 1854).
49. J. Fodor, *The Language of Thought* (Harvard University Press, Cambridge, MA, 1975).
50. M. Steedman, *The Syntactic Process* (MIT Press, Cambridge, MA, 2001).
51. I. Heim, A. Kratzer, *Semantics in Generative Grammar* (Wiley-Blackwell, Malden, MA, 1998).
52. S. T. Piantadosi, The computational origin of representation. *Minds Mach. (Dordr)* **31**, 1–58 (2021).
53. A. Church, An unsolvable problem of elementary number theory. *Am. J. Math.* **58**, 345–363 (1936).
54. H. Abelson, G. Sussman, *Structure and Interpretation of Computer Programs* (MIT Press, Cambridge, MA, 1996).
55. F. Cardone, J. R. Hindley, History of lambda-calculus and combinatory logic. *Handb. Hist. Log.* **5**, 723–817 (2006).
56. J. R. Hindley, J. P. Seldin, *Lambda-Calculus and Combinators, an Introduction* (Cambridge University Press, Cambridge, United Kingdom, 2008).
57. J. Feldman, Minimization of Boolean complexity in human concept learning. *Nature* **407**, 630–633 (2000).
58. N. D. Goodman, J. B. Tenenbaum, J. Feldman, T. L. Griffiths, A rational analysis of rule-based concept learning. *Cogn. Sci.* **32**, 108–154 (2008).
59. S. T. Piantadosi, J. B. Tenenbaum, N. D. Goodman, The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychol. Rev.* **123**, 392–424 (2016).
60. B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum, Human-level concept learning through probabilistic program induction. *Science* **350**, 1332–1338 (2015).
61. M. Amalric *et al.*, The language of geometry: Fast comprehension of geometrical primitives and rules in human adults and preschoolers. *PLOS Comput. Biol.* **13**, e1005273 (2017).
62. N. Chomsky, Three models for the description of language. *IRE Trans. Inf. Theory* **2**, 113–124 (1956).
63. N. Chomsky, *Syntactic Structures* (Mouton, The Hague, The Netherlands, 1957).

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

64. N. Chomsky, On certain formal properties of grammars. *Inf. Control* **2**, 137–167 (1959).
65. N. Chomsky, M. P. Schützenberger, "The algebraic theory of context-free languages" in *Provability, Computability and Reflection*, L. Beklemishev, Ed. (Studies in Logic and the Foundations of Mathematics, Elsevier), **vol. 26**, pp. 118–161 (1959).
66. P. A. Reich, The finiteness of natural language. *Language* **45**, 831–843 (1969).
67. G. Jäger, J. Rogers, Formal language theory: Refining the Chomsky hierarchy. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **367**, 1956–1970 (2012).
68. J. Higginbotham, "English is not a context-free language" in *The Formal Complexity of Natural Language*, W. Savitch, E. Bach, W. Marsh, G. Safran-Naveh, Eds. (Springer, 1987), pp. 335–348.
69. S. M. Shieber, "Evidence against the context-freeness of natural language" in *The Formal Complexity of Natural Language*, W. Savitch, E. Bach, W. Marsh, G. Safran-Naveh, Eds. (Springer, 1987), pp. 320–334.
70. W. J. Savitch, E. Bach, W. Marsh, G. Safran-Naveh, *The Formal Complexity of Natural Language* (Springer Science & Business Media, 1987).
71. L. Kallmeyer, *Parsing Beyond Context-Free Grammars* (Springer Science & Business Media, 2010).
72. G. K. Pullum, G. Gazdar, Natural languages and context-free languages. *Linguist. Philos.* **4**, 471–504 (1982).
73. S. T. Piantadosi, J. B. Tenenbaum, N. D. Goodman, Bootstrapping in a language of thought: A formal model of numerical concept learning. *Cognition* **123**, 199–217 (2012).
74. S. T. Piantadosi, "Learning and the language of thought," PhD thesis, Massachusetts Institute of Technology, Cambridge, MA (2011).
75. S. T. Piantadosi, R. Jacobs, Four problems solved by the probabilistic language of thought. *Curr. Dir. Psychol. Sci.* **25**, 54–59 (2016).
76. I. Yildirim, R. A. Jacobs, Transfer of object category knowledge across visual and haptic modalities: Experimental and computational studies. *Cognition* **126**, 135–148 (2013).
77. S. Depeweg, C. A. Rothkopf, F. Jäkel, Solving Bongard problems with a visual language and pragmatic reasoning. arXiv [Preprint] (2018). https://arxiv.org/abs/1804.04452 (Accessed 1 September 2021).
78. M. C. Overlan, R. A. Jacobs, S. T. Piantadosi, "A hierarchical probabilistic language-of-thought model of human visual concept learning" in *Proceedings of the 38th Annual Meeting of the Cognitive Science Society*, A. Papafragou, D. Grodner, D. Mirman, J.C. Trueswell, Eds. (Cognitive Science Society, 2016), pp. 1259–1264.
79. N. D. Goodman, J. B. Tenenbaum, T. Gerstenberg, *Concepts in a Probabilistic Language of Thought* (MIT Press, 2015).
80. B. M. Lake, T. D. Ullman, J. B. Tenenbaum, S. J. Gershman, Building machines that learn and think like people. *Behav. Brain Sci.* **40**, e253 (2017).
81. A. Rothe, B. M. Lake, T. Gureckis, "Question asking as program generation" in *Advances in Neural Information Processing Systems*, I. Guyon et al., Eds. (Curran Associates, 2017), pp. 1046–1055.
82. G. Erdogan, I. Yildirim, R. A. Jacobs, From sensory signals to modality-independent conceptual representations: A probabilistic language of thought approach. *PLOS Comput. Biol.* **11**, e1004610 (2015).
83. I. Yildirim, R. A. Jacobs, Learning multisensory representations for auditory-visual transfer of sequence category knowledge: A probabilistic language of thought approach. *Psychon. Bull. Rev.* **22**, 673–686 (2015).
84. S. Romano *et al.*, Bayesian validation of grammar productions for the language of thought. *PLoS One* **13**, e0200420 (2018).
85. J. Rule, E. Schulz, S. T. Piantadosi, J. B. Tenenbaum, "Learnin g list concepts through program induction" in *Proceedings of the Cognitive Science Society* (Cognitive Science Society, 2018).
86. K. Ellis, L. Morales, M. Sablé-Meyer, A. Solar-Lezama, J. Tenenbaum, "Learning libraries of subroutines for neurally–guided Bayesian program induction" in *Advances in Neural Information Processing Systems*, S. Bengio et al., Eds. (Curran Associates, 2018), pp. 7805–7815.
87. T. Ullman, N. Goodman, J. Tenenbaum, Theory learning as stochastic search in the language of thought. *Cogn. Dev.* **27**, 455–480 (2012).
88. S. Planton *et al.*, A theory of memory for binary sequences: Evidence for a mental compression algorithm in humans. *PLOS Comput. Biol.* **17**, e1008598 (2021).
89. N. Chomsky, *The Minimalist Program* (Cambridge University Press, 1995).
90. D. Adger, *Core Syntax: A Minimalist Approach* (Oxford University Press, Oxford, United Kingdom, 2003).
91. N. Goodman, V. Mansinghka, D. Roy, K. Bonawitz, D. Tarlow, Church: A language for generative models. arXiv [Preprint] (2012). https://arxiv.org/abs/1206.3255 (Accessed 1 September 2021).
92. D. Koller, D. McAllester, A. Pfeffer, "Effective Bayesian inference for stochastic programs" in *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (AAAI Press, 1997), pp. 740–747.
93. T. Icard, Why be random? *Mind* **130**, 111–139 (2021).
94. A. Kornai, Probabilistic grammars and languages. *J. Log. Lang. Inf.* **20**, 317–328 (2011).
95. T. F. Icard, Calibrating generative models: The probabilistic Chomsky-Schützenberger hierarchy. *J. Math. Psychol.* **95**, 102308 (2020).
96. S. C. Levinson, Recursion in pragmatics. *Language* **89**, 149–162 (2013).
97. M. C. Corballis, *The Recursive Mind: The Origins of Human Language, Thought, and Civilization* (Princeton University Press, 2011).
98. B. Lake, S. T. Piantadosi, People infer recursive visual concepts from just a few examples. *Comput. Brain Behav.* **3**, 54–65 (2020).
99. M. D. Martins, W. T. Fitch, "Investigating recursion within a domain-specific framework" in *Language and Recursion*, F. Lowenthal, L. Lefebvre, Eds. (Springer, 2014), pp. 15–26.
100. S. Muggleton, "Learning from positive data" in *International Conference on Inductive Logic Programming*, S. Muggleton, Ed. (Springer), pp. 358–376 (1996).
101. E. Dechter, J. Malmaud, R. P. Adams, J. B. Tenenbaum, "Bootstrap learning via modular concept discovery" in *Twenty-Third International Joint Conference on Artificial Intelligence*, F. Rossi, Ed. (AAAI Press, 2013), pp. 1302–1309.
102. N. D. Goodman, A. Stuhlmüller, *The design and implementation of probabilistic programming languages*. http://dippl.org. Accessed 1 September 2021.
103. J. Rogers, G. K. Pullum, Aural pattern recognition experiments and the subregular hierarchy. *J. Log. Lang. Inf.* **20**, 329–342 (2011).
104. S. Ferrigno, S. J. Cheyette, S. T. Piantadosi, J. F. Cantlon, Recursive sequence generation in monkeys, children, U.S. adults, and native Amazonians. *Sci. Adv.* **6**, eaaz1002 (2020).
105. X. Jiang *et al.*, Production of supra-regular spatial sequences by macaque monkeys. *Curr. Biol.* **28**, 1851–1859.e4 (2018).
106. K. Vijay-Shanker, D. J. Weir, The equivalence of four extensions of context-free grammars. *Math. Syst. Theory* **27**, 511–546 (1994).
107. A. K. Joshi, An introduction to tree adjoining grammars. *Math. Lang.* **1**, 87–115 (1987).
108. D. Lind, B. Marcus, *An Introduction to Symbolic Dynamics and Coding* (Cambridge University Press, 1995).
109. P. Boullier, "Chinese numbers, mix, scrambling, and range concatenation grammars" in *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, H.S. Thompson, A. Lascarides, Eds. (Association for Computational Linguistics, 1999), pp. 53–60.
110. D. Radzinski, Chinese number-names, tree adjoining languages, and mild context-sensitivity. *Comput. Linguist.* **17**, 277–299 (1991).
111. R. Huybregts, "The weak inadequacy of context-free phrase structure grammars" in *Van periferie naar kern*, G.J. de Haan, M. Trommelen, W. Zonneveld, Eds. (Foris, Dordrecht, The Netherlands, 1984), pp. 81–99.
112. C. Culy, The complexity of the vocabulary of Bambara. *Linguist. Philos.* **8**, 345–351 (1985).
113. E. P. Stabler, Varieties of crossing dependencies: Structure dependence and mild context sensitivity. *Cogn. Sci.* **28**, 699–720 (2004).
114. Y. Bar-Hillel, E. Shamir, Finite-state languages: Formal representations and adequacy problems. *Bull. Res. Counc. Isr.* **8F**, 155–166 (1960).
115. D. T. Langendoen, "On the inadequacy of type-3 and type-2 grammars for human languages" in *Studies in Descriptive and Historical Linguistics: Festschrift for Winfred P. Lehmann*, P.J. Hopper, Ed. (John Benjamins, 1977), pp. 159–171.
116. P. M. Postal, "Limitations of phrase-structure grammars" in *The Structure of Language: Readings in the Philosophy of Language*, J. Fodor, J. Katz, Eds. (Prentice-Hall, Englewood Cliffs, NJ, 1964), pp. 137–151.
117. F. Mollica, S. T. Piantadosi, Humans store about 1.5 megabytes of information during language acquisition. *R. Soc. Open Sci.* **6**, 181393 (2019).
118. G. K. Pullum, B. C. Scholz, "Recursion and the infinitude claim" in *Recursion in Human Language*, H. van der Hulst, Ed. (Mouton de Gruyter, Berlin, Germany, 2010), pp. 113–138.
119. C. Yang, *The Infinite Gift* (Simon and Schuster, New York, NY, 2006).
120. H. Lasnik, M. A. Depiante, A. Stepanov, *Syntactic Structures Revisited: Contemporary Lectures on Classic Transformational Theory* (MIT Press, 2000).
121. S. T. Piantadosi, E. Fedorenko, Infinitely productive language can arise from chance under communicative pressure. *J. Lang. Evol.* **2**, 141–147 (2016).
122. R. Berwick, *The Acquisition of Syntactic Knowledge* (MIT Press, Cambridge, MA, 1985).
123. M. R. Manzini, K. Wexler, Parameters, binding theory, and learnability. *Linguist. Inq.* **18**, 413–444 (1987).
124. R. Clark, The selection of syntactic knowledge. *Lang. Acquis.* **2**, 83–149 (1992).
125. J. Musolino, K. Laity d'Agostino, S. Piantadosi, Why we should abandon the semantic subset principle. *Lang. Learn. Dev.* **15**, 32–46 (2019).
126. J. D. Fodor, W. G. Sakas, The subset principle in syntax: Costs of compliance. *J. Linguist.* **41**, 513–569 (2005).
127. W. J. Savitch, Why it might pay to assume that languages are infinite. *Ann. Math. Artif. Intell.* **8**, 17–25 (1993).
128. A. Carnie, *Syntax: A Generative Introduction* (John Wiley, 2013).
129. E. W. Bach, Discontinuous constituents in generalized categorial grammar. *North East Linguistic Soc.* **11**, 1–12 (1981).
130. G. K. Pullum, "Context-freeness and the computer processing of human languages" in *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, M. Marcus, Ed. (Association for Computational Linguistics, 1983), pp. 1–6.
131. C. Kemp, T. Regier, Kinship categories across languages reflect general communicative principles. *Science* **336**, 1049–1054 (2012).
132. E. Gibson *et al.*, How efficiency shapes human language. *Trends Cogn. Sci.* **23**, 389–407 (2019).
133. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (W.H. Freeman, 1982).
134. J. R. Saffran, R. N. Aslin, E. L. Newport, Statistical learning by 8-month-old infants. *Science* **274**, 1926–1928 (1996).
135. J. R. Saffran, E. K. Johnson, R. N. Aslin, E. L. Newport, Statistical learning of tone sequences by human infants and adults. *Cognition* **70**, 27–52 (1999).
136. M. R. Brent, T. A. Cartwright, Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition* **61**, 93–125 (1996).
137. S. Goldwater, T. L. Griffiths, M. Johnson, A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition* **112**, 21–54 (2009).
138. R. N. Aslin, E. L. Newport, Distributional language learning: Mechanisms and models of category formation. *Lang. Learn.* **64**, 86–105 (2014).
139. G. F. Marcus, S. Vijayan, S. Bandi Rao, P. M. Vishton, Rule learning by seven-month-old infants. *Science* **283**, 77–80 (1999).

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.

140. G. F. Marcus, *The Algebraic Mind: Integrating Connectionism and Cognitive Science* (MIT Press, 2003).
141. J. L. Morgan, E. L. Newport, The role of constituent structure in the induction of an artificial language. *J. Verbal Learning Verbal Behav.* **20**, 67–85 (1981).
142. J. L. Morgan, R. P. Meier, E. L. Newport, Structural packaging in the input to language learning: Contributions of prosodic and morphological marking of phrases to the acquisition of language. *Cognit. Psychol.* **19**, 498–550 (1987).
143. J. Saffran, The use of predictive dependencies in language learning. *J. Mem. Lang.* **44**, 493–515 (2001).
144. R. L. Gómez, Variability and detection of invariant structure. *Psychol. Sci.* **13**, 431–436 (2002).
145. R. Gómez, J. Maye, The developmental trajectory of nonadjacent dependency learning. *Infancy* **7**, 183–206 (2005).
146. E. L. Newport, R. N. Aslin, Learning at a distance I. Statistical learning of non-adjacent dependencies. *Cognit. Psychol.* **48**, 127–162 (2004).
147. M. D. Braine, On learning the grammatical order of words. *Psychol. Rev.* **70**, 323–348 (1963).
148. M. D. Braine, Learning the positions of words relative to a marker element. *J. Exp. Psychol.* **72**, 532–540 (1966).
149. A. S. Reber, Implicit learning of artificial grammars. *J. Verbal Learning Verbal Behav.* **6**, 855–863 (1967).
150. R. C. Berwick, S. Pilato, Learning syntax by automata induction. *Mach. Learn.* **2**, 9–38 (1987).
151. A. Cleeremans, J. L. McClelland, Learning the structure of event sequences. *J. Exp. Psychol. Gen.* **120**, 235–253 (1991).
152. A. Cleeremans, D. Servan-Schreiber, J. L. McClelland, Finite state automata and simple recurrent networks. *Neural Comput.* **1**, 372–381 (1989).
153. E. Servan-Schreiber, J. R. Anderson, Learning artificial grammars with competitive chunking. *J. Exp. Psychol. Learn. Mem. Cogn.* **16**, 592 (1990).
154. P. A. Reeder, E. L. Newport, R. N. Aslin, From shared contexts to syntactic categories: The role of distributional information in learning linguistic form-classes. *Cognit. Psychol.* **66**, 30–54 (2013).
155. N. Chomsky *et al.*, *Language and Problems of Knowledge: The Managua Lectures* (MIT Press, 1988).
156. S. Laurence, E. Margolis, The poverty of the stimulus argument. *Br. J. Philos. Sci.* **52**, 217–276 (2001).
157. R. C. Berwick, P. Pietroski, B. Yankama, N. Chomsky, Poverty of the stimulus revisited. *Cogn. Sci.* **35**, 1207–1242 (2011).
158. L. Pearl, Poverty of the stimulus without tears. *Lang. Learning Dev.*, 10.1080/15475441.2021.1981908 (2020).
159. C. L. Baker, Syntactic theory and the projection problem. *Linguist. Inq.* **10**, 533–581 (1979).
160. A. Clark, S. Lappin, Complexity in language acquisition. *Top. Cogn. Sci.* **5**, 89–110 (2013).
161. G. Pullum, B. Scholz, Empirical assessment of stimulus poverty arguments. *Linguist. Rev.* **18**, 9–50 (2002).
162. B. MacWhinney, A multiple process solution to the logical problem of language acquisition. *J. Child Lang.* **31**, 883–914 (2004).
163. K. Ellis *et al.*, Write, execute, assess: Program synthesis with a REPL. arXiv [Preprint] (2019). https://arxiv.org/abs/1906.04604 (Accessed 1 September 2021).
164. A. Graves, G. Wayne, I. Danihelka, Neural turing machines. arXiv [Preprint] (2014). https://arxiv.org/abs/1410.5401 (Accessed 1 September 2021).
165. K. Wexler, M. Manzini, "Parameters and learnability in binding theory" in *Parameter Setting*, T. Roeper, Ed. (Springer, 1987), pp. 41–76.
166. P. Smolensky, "*The initial state and 'richness of the base' in Optimality Theory*" (Tech. Rep. JHU-CogSci-96-4, Department of Cognitive Science, Johns Hopkins University, 1996).
167. S. Crain, "The semantic subset principle in the acquisition of quantification" in *Workshop on the Acquisition of WH-Extraction and Related Work on Quantification* (University of Massachusetts, Amherst, MA, 1992).
168. J. Musolino, On the semantics of the subset principle. *Lang. Learn. Dev.* **2**, 195–218 (2006).
169. D. MacLaughlin, Language acquisition and the subset principle. *Linguist. Rev.* **12**, 143–191 (1995).
170. M. Bowerman, "The 'no negative evidence' problem: How do children avoid constructing an overly general grammar?" in *Explaining Language Universals*, J.A. Hawkins, Ed. (Basil Blackwell, 1988), pp. 73–101.
171. D. Angluin, "*Identifying languages from stochastic examples*" (Rep. YALEU/DCS/RR-614, Department of Computer Science, Yale University, 1988).
172. J. Tenenbaum, "A Bayesian framework for concept learning," PhD thesis, Massachusetts Institute of Technology, Cambridge, MA (1999).
173. G. F. Marcus, Negative evidence in language acquisition. *Cognition* **46**, 53–85 (1993).
174. M. M. Chouinard, E. V. Clark, Adult reformulations of child errors as negative evidence. *J. Child Lang.* **30**, 637–669 (2003).
175. E. V. Clark, *First Language Acquisition* (Cambridge University Press, 2009).
176. E. V. Clark, Conversational repair and the acquisition of language. *Discourse Process.* **57**, 441–459 (2020).
177. M. Saxton, The contrast theory of negative input. *J. Child Lang.* **24**, 139–161 (1997).
178. T. Schoneberger, Three myths from the language acquisition literature. *Anal. Verbal Behav.* **26**, 107–131 (2010).
179. P. Shafto, N. D. Goodman, T. L. Griffiths, A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognit. Psychol.* **71**, 55–89 (2014).
180. L. Pearl, J. Sprouse, Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem. *Lang. Acquis.* **20**, 23–68 (2013).
181. C. Boeckx, Islands. *Lang. Linguist. Compass* **2**, 151–167 (2008).
182. A. E. Goldberg, *Constructions at Work: The Nature of Generalization in Language* (Oxford University Press on Demand, 2006).
183. A. E. Goldberg, Subtle implicit language facts emerge from the functions of constructions. *Front. Psychol.* **6**, 2019 (2016).
184. A. Abeillé, B. Hemforth, E. Winckel, E. Gibson, Extraction from subjects: Differences in acceptability depend on the discourse function of the construction. *Cognition* **204**, 104293 (2020).
185. B. MacWhinney, *The CHILDES Project: Tools for Analyzing Talk* (Lawrence Erlbaum, Hillsdale, NJ, 2000).
186. W. G. Mitchener, M. Becker, Computational models of learning the raising-control distinction. *Res. Lang. Comput.* **8**, 169–207 (2010).
187. J. Borges, *The Library of Babel in Labyrinths* (Penguin, Harmondsworth, United Kingdom, 1970).
188. D. Rumelhart, J. McClelland, *Parallel Distributed Processing* (MIT Press, Cambridge, MA, 1986).
189. P. Smolensky, G. Legendre, *The Harmonic Mind* (MIT Press, Cambridge, MA, 2006).
190. J. Elman, Finding structure in time. *Cogn. Sci.* **14**, 179–211 (1990).
191. C. L. Giles *et al.*, Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Comput.* **4**, 393–405 (1992).
192. M. H. Christiansen, N. Chater, Toward a connectionist model of recursion in human linguistic performance. *Cogn. Sci.* **23**, 157–205 (1999).
193. W. Tabor, "Recursion and recursion-like structure in ensembles of neural elements" in *Unifying Themes in Complex Systems. Proceedings of the VIII International Conference on Complex Systems*, H. Sayama, A. Minai, D. Braha, Y. Bar-Yam, Eds. (New England Complex Systems Institute, Cambridge, MA), pp. 1494–1508 (2011).
194. W. Tabor, A dynamical systems perspective on the relationship between symbolic and non-symbolic computation. *Cogn Neurodyn* **3**, 415–427 (2009).
195. W. Tabor, C. Juliano, M. K. Tanenhaus, Parsing in a dynamical system: An attractor-based account of the interaction of lexical and structural constraints in sentence processing. *Lang. Cogn. Process.* **12**, 211–271 (1997).
196. A. Cleeremans, J. Elman, *Mechanisms of Implicit Learning: Connectionist Models of Sequence Processing* (MIT Press, 1993).
197. T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **13**, 55–75 (2018).
198. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
199. Y. Goldberg, A primer on neural network models for natural language processing. *J. Artif. Intell. Res.* **57**, 345–420 (2016).
200. J. Hewitt, M. Hahn, S. Ganguli, P. Liang, C. D. Manning, RNNs can generate bounded hierarchical languages with optimal memory. arXiv [Preprint] (2020). https://arxiv.org/abs/2010.07515 (Accessed 1 September 2021).
201. W. T. Fitch, M. D. Hauser, Computational constraints on syntactic processing in a nonhuman primate. *Science* **303**, 377–380 (2004).
202. T. Hunter, "The Chomsky Hierarchy" in *A Companion to Chomsky*, N. Allott, T. Lohndal, G. Rey, Eds. (Blackwell, 2020), pp. 74–95.
203. M. Li, P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications* (Springer-Verlag, New York, NY, 2008).
204. P. D. Grünwald, *The Minimum Description Length Principle* (MIT Press, 2007).
205. J. Feldman, The simplicity principle in human concept learning. *Curr. Dir. Psychol. Sci.* **12**, 227 (2003).
206. N. Chater, P. Vitányi, Simplicity: A unifying principle in cognitive science? *Trends Cogn. Sci.* **7**, 19–22 (2003).
207. A. S. Hsu, N. Chater, P. M. Vitányi, The probabilistic analysis of language acquisition: Theoretical, computational, and experimental analysis. *Cognition* **120**, 380–390 (2011).
208. M. Tomasello, *Constructing a Language: A Usage-Based Theory of Language Acquisition* (Harvard University Press, 2009).
209. M. Tomasello, "The usage-based theory of language acquisition" in *The Cambridge Handbook of Child Language*, E.L. Bavin, Ed. (Cambridge University Press, 2009), pp. 69–87.
210. A. Goldberg, L. Suttle, Construction grammar. *Wiley Interdiscip. Rev. Cogn. Sci.* **1**, 468–477 (2010).
211. L. S. Zettlemoyer, M. Collins, "Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars" in *UAI*'05: Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence, F. Bacchus, T. Jaakkola, Eds. (AUAI Press, 2005), pp. 658–666.
212. P. Liang, M. I. Jordan, D. Klein, Learning dependency-based compositional semantics. *Comput. Linguist.* **39**, 389–446 (2013).
213. T. Kwiatkowski, S. Goldwater, L. Zettlemoyer, M. Steedman, "A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings" in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, W. Daelemans, Ed. (Association for Computational Linguistics, 2012), pp. 234–244.
214. L. Dong, M. Lapata, "Language to logical form with neural attention" in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, K. Erk, N.A. Smith, Eds. (Association for Computational Linguistics, 2016), vol. 1, pp. 33–43.
215. I. Van Rooij, The tractable cognition thesis. *Cogn. Sci.* **32**, 939–984 (2008).
216. J. B. Tenenbaum, C. Kemp, T. L. Griffiths, N. D. Goodman, How to grow a mind: Statistics, structure, and abstraction. *Science* **331**, 1279–1285 (2011).
217. W. Vousden, W. M. Farr, I. Mandel, Dynamic temperature selection for parallel tempering in Markov chain Monte Carlo simulations. *Mon. Not. R. Astron. Soc.* **455**, 1919–1937 (2015).
218. O. Tange, Gnu parallel – The command-line power tool.; *login. The USENIX Mag.* **36**, 42–47 (2011).
219. C. L. Hudson Kam, E. L. Newport, Getting it right by getting it wrong: When learners change languages. *Cognit. Psychol.* **59**, 30–66 (2009).

Downloaded from https://www.pnas.org by 12.215.51.2 on May 6, 2022 from IP address 12.215.51.2.