Cognition 168 (2017) 320–334

Contents lists available at ScienceDirect

Cognition

journal homepage: www.elsevier.com/locate/COGNIT

Original Articles

Learning abstract visual concepts via probabilistic program induction in a Language of Thought ☆

Matthew C. Overlan, Robert A. Jacobs *, Steven T. Piantadosi

Department of Brain & Cognitive Sciences, University of Rochester, Rochester, NY 14627, United States

ARTICLE INFO

Article history:
Received 14 February 2017
Revised 6 July 2017
Accepted 9 July 2017

Keywords:
Concept learning
Visual learning
Language of Thought
Computational modeling
Behavioral experiment

ABSTRACT

The ability to learn abstract concepts is a powerful component of human cognition. It has been argued that variable binding is the key element enabling this ability, but the computational aspects of variable binding remain poorly understood. Here, we address this shortcoming by formalizing the Hierarchical Language of Thought (HLOT) model of rule learning. Given a set of data items, the model uses Bayesian inference to infer a probability distribution over stochastic programs that implement variable binding. Because the model makes use of symbolic variables as well as Bayesian inference and programs with stochastic primitives, it combines many of the advantages of both symbolic and statistical approaches to cognitive modeling. To evaluate the model, we conducted an experiment in which human subjects viewed training items and then judged which test items belong to the same concept as the training items. We found that the HLOT model provides a close match to human generalization patterns, significantly outperforming two variants of the Generalized Context Model, one variant based on string similarity and the other based on visual similarity using features from a deep convolutional neural network. Additional results suggest that variable binding happens automatically, implying that binding operations do not add complexity to peoples' hypothesized rules. Overall, this work demonstrates that a cognitive model combining symbolic variables with Bayesian inference and stochastic program primitives provides a new perspective for understanding people's patterns of generalization.

1. Introduction

Induction, the ability to discover latent patterns and structure from a set of data items, is a hallmark of human thinking. This ability underlies our remarkable language acquisition and conceptual development, and its roots have been found in infancy. Marcus, Vijayan, Bandi Rao, and Vishton (1999) studied the ability of seven-month-olds to infer abstract rules from acoustic sequences. They showed that infants presented with syllable sequences that follow an *ABA* pattern, like "ga ti ga" and "li na li", recognized novel sequences following that pattern even when those sequences contained new syllables that the infants had not heard, like "wo fe wo". Because the test items could not be distinguished based on concrete features like transitional statistics between syllables, sequence length, or prosody, they reasoned that the infants had learned an abstract rule that reflected latent structure.

Even though rules like *ABA* are simple, they illustrate a foundational computational element of human abstract rule learning: we can easily and fluidly handle *variable binding* (Jackendoff, 2003; Marcus, 2003). Variable binding refers to the ability to assign a name to some piece of information for storage and later retrieval. In the case of *ABA* rules, infants must remember the first syllable (that is, store it in a variable *A*) so it can be compared to subsequent syllables. The use of variables is what allows the *ABA* rule to be abstract: computations can refer to variable names rather than the values stored therein, so the rule can reflect the *relationship* between pieces of information rather than the concrete features of that information. It does not matter what values the *A*s and *B*s have, so long as the resulting sequence obeys the right pattern of repetition.

Variable binding has been at the center of a key debate in cognitive science (Jackendoff, 2003; Marcus, 2003), much of which has focused on the role of statistics in learning abstract rules.

☆ A preliminary version of a portion of this research appeared in the *Proceedings of the 38th Annual Meeting of the Cognitive Science Society*. We thank Goker Erdogan and Frank Mollica for helpful discussions, and Goker Erdogan for neural network code. This work was supported by research grants from the Air Force Office of Scientific Research (FA9550-12-1-0303) and the National Science Foundation (BCS-1400784).

* Corresponding author.
E-mail addresses: m.overlan@rochester.edu (M.C. Overlan), robbie@bcs.rochester.edu (R.A. Jacobs), spiantadosi@bcs.rochester.edu (S.T. Piantadosi).

http://dx.doi.org/10.1016/j.cognition.2017.07.005
0010-0277/

Proponents of a rule-based approach point out that statistics alone are insufficient for learning rules that require variable binding, since (as with *ABA* rules) variable binding allows learners to generalize to novel stimuli for which they have no statistical information.[1] In response, proponents of a statistical approach point out that a pure rule-based approach cannot explain why learners choose the rules they do from the infinitely many that are consistent with the input. In addition, infant studies have shown that the statistics of the input affect generalization of *ABA*-like rules. For instance, Gerken (2006) presented infants with syllable sequences that were logically consistent with two different rules and showed that they learned the one that was best supported by the statistics of the input they had received.

This tension between rules and statistics has been addressed in recent years by hybrid models, sometimes referred to as probabilistic language of thought (pLOT) models (Piantadosi & Jacobs, 2016). pLOT models operate with infinite hypothesis spaces by employing a compositional system for creating rules (Erdogan, Yildirim, & Jacobs, 2015; Goodman, Tenenbaum, Feldman, & Griffiths, 2008; Kemp, 2012; Siskind, 1996; Piantadosi, Tenenbaum, & Goodman, 2012; Piantadosi, Tenenbaum, & Goodman, 2016; Ullman, Goodman, & Tenenbaum, 2012; Yildirim & Jacobs, 2015). These models integrate rules and statistics by employing statistical (e.g., Bayesian) inference over such structured hypothesis spaces (Tenenbaum & Griffiths, 2001). By using structured, symbolic hypotheses, these models can represent "rule-based" concepts. And by maintaining uncertainty over rules, these models can operate in the presence of noisy data showing gradience or typicality effects.

While there have been several process-level models of variable binding in neural networks (Hummel & Holyoak, 1997; Smolensky, 1990), few models have approached the problem of variable binding from the ideal-observer (Geisler, 2003) perspective, considering a computational-level explanation for rule learning as the rational outcome of an optimal computation.

Frank and Tenenbaum (2011) implemented an ideal-observer model of *ABA*-style rule learning in which variables are *implicit*. They represented these rules with 3-tuples like $(is_{ga}, *, =_1)$, which meant that the first syllable is a specific one ('ga' in this case), the second syllable is free (it could be anything), and the third syllable is equal to the first. Their model used Bayesian techniques to approximately capture generalization patterns by performing inference over this space of hypotheses. While this representation is strictly sufficient to capture *ABA*-like patterns, it has important shortcomings. Since its variables are simply built in as a baseline in the representation, their model is unable to explain why learners may or may not come to hypothesize variables in the first place. Second, the space of concepts and rules lacks any prior biases over hypotheses. In particular, there is no notion of simplicity or complexity, a key inductive bias for human learners (Chater & Vitanyi, 2003; Feldman, 2000). A simplicity bias allows learners to avoid over-fitting and to come to a reasoned compromise between generality and fit-to-data. Finally, their representation is highly specific to identity-based, *ABA*-like patterns. This makes it unclear how their methods and ideas might generalize to the many other classes of rule-based concepts that have been studied in the literature.

Our model addresses all of these issues. We represent concepts as *probabilistic programs*, programs with stochastic primitives such that they produce different random outputs each time they are run. A program-based representation allows hypotheses in our model to contain explicit variable binding operations. To infer these programs from data, we build upon the pLOT framework's capability of Bayesian statistical inference over a structured space of symbolic hypotheses. Following Goodman et al. (2008), we assume a rich generative model for concepts that uses a probabilistic context-free grammar to represent an infinite space of hypotheses. This grammar-based approach provides a natural simplicity-favoring prior over programs. These qualities of explicit variable binding and robust statistical inference allow us to reason about abstraction in rule learning in a way that is not possible with a fixed assignment of items to slots and a uniform prior.

At the highest level of generality, the goal of our research program is to characterize human learning and reasoning as forms of program induction. We regard the pLOT as a promising framework for developing such a characterization. Our model combines a symbolic approach, which provides a means for achieving abstraction (through variable binding) and for defining an infinite structured hypothesis space (through compositionality), with a statistical approach, which provides a means for learning representations from noisy data in a way that quantifies uncertainty (through Bayesian inference and the use of programs with stochastic primitives). A novel innovation of our model is that it combines statistical program induction (i.e., Bayesian inference of a probability distribution over programs) with the use of probabilistic programs (i.e., those with stochastic primitives). We see the work presented in this paper as an early step toward extending symbolic-statistical hybrid models so that they can be used to develop theoretical accounts in many domains of human cognition.

In addition to the theoretical contributions of our computational framework, our secondary goal is to provide empirical results that further our understanding of human rule learning. Currently, our knowledge of human learning of *ABA*-like rules is limited to data available from infant studies. The necessary sparseness of these data makes it difficult to distinguish between competing models at a fine grain. Therefore, we carried out a behavioral experiment with adults that is inspired by infants' learning of *ABA*-like patterns. This allows us to assess subjects' generalization patterns at a detailed level.

The plan of this paper is as follows. First, we describe the details of our behavioral experiment, then the details of our model. We next compare the generalization performance of our model with those of our experimental subjects. We find that our model provides an excellent account of our experimental data, outperforming alternative models that lack key elements such as variable binding. Finally, we test a variant of our model in order to determine which way of handling variable abstractions provides the most accurate fit to human generalizations.

2. Behavioral experiment

In our behavioral experiment, we evaluated human subjects' abilities to infer an abstract visual concept or category from a small number of exemplars. This was accomplished by showing subjects exemplars consistent with a concept, and then asking them whether they believed each of several test items was also an exemplar from the same concept. All subjects were US residents over the age of 18. They participated in the experiment over the world wide web using the Amazon Mechanical Turk crowdsourcing platform. Raw data from the experiment can be found in the online supplemental materials.

Visual stimuli were images depicting 3D, part-based objects rendered with realistic lighting and texture (see Fig. 1a for the set of possible object parts). Based on these images, it was easy to segment an object into its component parts. Our stimuli have the advantage of being both novel—meaning that subjects did

[1] While much of this debate has focused on the deficiencies of connectionist models (Fodor & Pylyshyn, 1988) and possible connectionist solutions (Gayler, 2004; Smolensky, 1990; Smolensky & Legendre, 2006; van der Velde & de Kamps, 2006), these arguments apply to any sub-symbolic theory that does not have an explicit representation of variables.

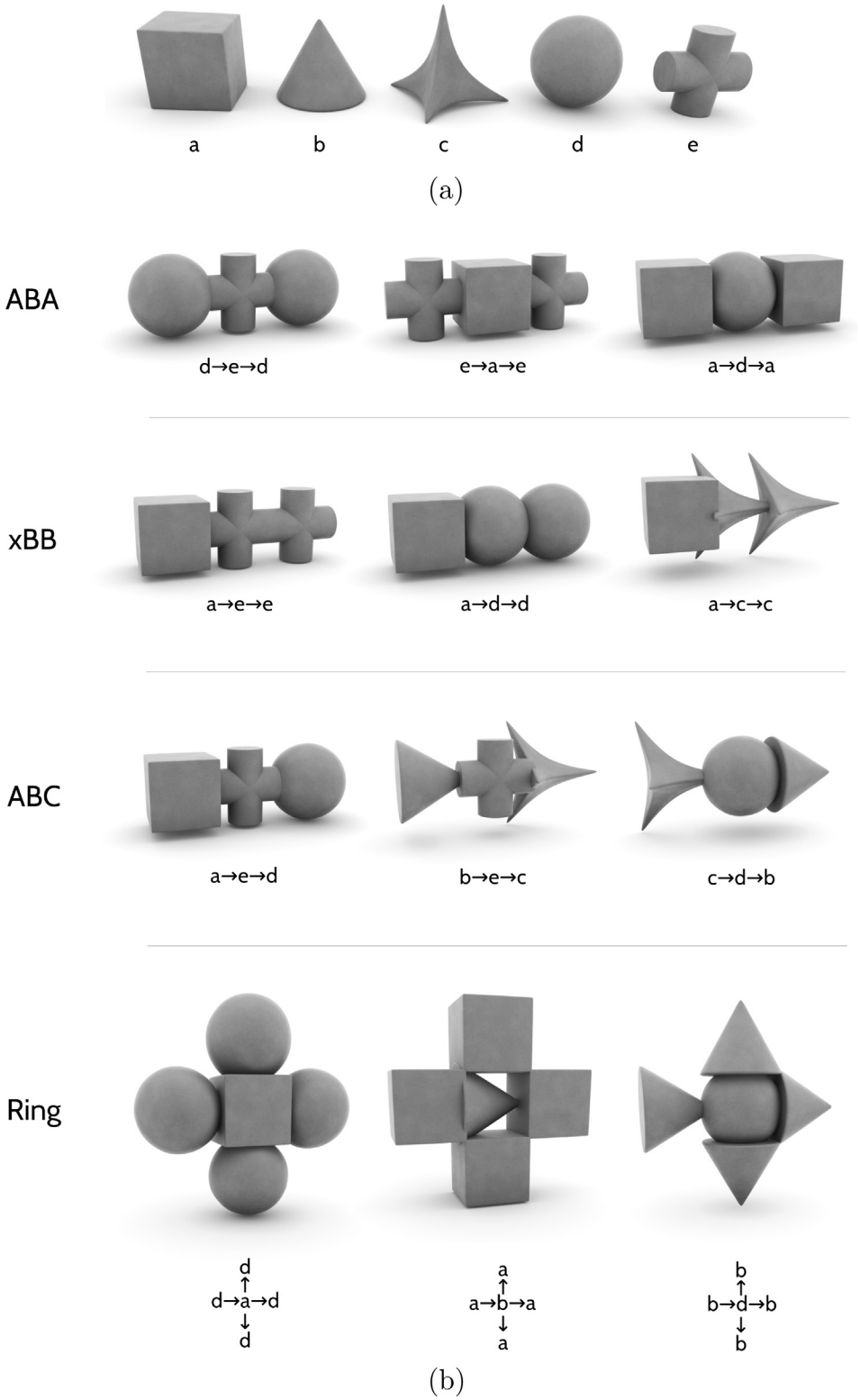

Fig. 1. (a) Object parts used in the experiment along with their string representation as used by the computational models. Participants viewed these parts (but not the string representations) during the instructions phase of the experiment. (b) Training exemplars for all experimental conditions along with their string representations as used by the models.

not enter the experiment with prior beliefs about the depicted objects—and naturalistic.

Importantly, visual concepts were defined based on objects' parts and the spatial relations among parts. Four concepts were used in the experiment:

- *ABA*: Concept *ABA* is analogous to the rule used in Marcus et al. (1999). In objects consistent with this concept, the leftmost and rightmost parts are identical, but different from the middle part (top row of Fig. 1b).

- *xBB*: Concept *xBB* is analogous to the rule used in Gerken (2006). Here, the leftmost part is always a specific part, whereas the middle and rightmost parts are different from the leftmost part, though are identical to each other (second row of Fig. 1b). These stimuli are consistent with both the *xBB* rule and the broader *ABB* rule in which the leftmost part can be any part that differs from the right two. However, the *xBB* concept is "narrower" than the *ABB* concept in that there are fewer possible objects that satisfy it—with n possible parts, there are $n-1$ *xBB* objects and $n(n-1)$ *ABB* objects. A comparison of subjects' generalizations after viewing exemplars from the *xBB* concept allows us to verify the prior literature suggesting that people learn a narrower concept when the evidence supports it.
- *ABC*: We also wanted to study people's generalizations when the evidence supports a "wider concept" which, to our knowledge, has not been previously studied. We therefore implemented an *ABC* concept in which all three parts differ (third row of Fig. 1b). This concept has $n(n-1)(n-2)$ satisfying objects.
- *Ring*: Previous experiments have primarily used stimuli with three parts arranged linearly (either in time, by necessity, or in space). To go beyond this context, we tested a concept with five-part objects where the parts are arranged in two dimensions. In the *Ring* concept, four of the five parts comprising an object are identical and are organized to form a ring. The fifth part is distinct and resides at the center of the ring (bottom row of Fig. 1b). Like the *ABA* concept, this concept has $n(n-1)$ satisfying objects.

To eliminate possible order or experience effects, each subject participated in a brief experimental session using a single trial with a single visual concept. The session consisted of three stages: an instruction stage, a training stage, and a testing stage. During the instruction stage, subjects were shown all five possible part shapes as in Fig. 1a. Next, subjects participated in a training stage where they were shown three exemplars from a concept. Each subject was allowed to view the exemplars for as long as he or she wished. Training was followed by testing. During testing, subjects were shown an array of 24 test items. Test items had the same general structure as training exemplars (three parts arranged linearly or in a ring for the *Ring* condition), but differed in which parts occupied each position in an item. For each condition, we chose a unique set of test items in order to broadly cover a range of patterns both consistent with and inconsistent with the intended concept. Participants chose 'yes' or 'no' for each test item to indicate whether it belonged to the same concept as the training exemplars. To eliminate memory demands, training exemplars remained available for viewing at the top of the web page for the duration of the test stage.

One hundred twenty subjects participated in experimental sessions, with thirty subjects assigned to sessions using each visual concept. However, data from some subjects were not used in the analyses reported below. At least one training exemplar was present in the test items. If a subject answered 'no' to a test item that was identical to an exemplar, his or her results were excluded from our analyses. Based on this criteria, the responses from 29, 28, 23, and 29 subjects were used in the analyses of the *ABA*, *xBB*, *ABC*, and *Ring* conditions, respectively.

3. Computational models

This section describes our proposed computational model, referred to as the Hierarchical Language of Thought (HLOT) model, as well as two alternative models.

3.1. Hierarchical Language of Thought (HLOT) model

The HLOT model is a rule-learning model. Unlike many rule-learning models (like those that employ Boolean logic, for example), the HLOT model assumes that observed data exemplifying some concept are the result of an unobserved, stochastic generative process (Feldman, 1997; Kemp, Bernstein, & Tenenbaum, 2005; Lake, Salakhutdinov, & Tenenbaum, 2015; Stuhlmüller, Tenenbaum, & Goodman, 2010; Yildirim & Jacobs, 2015). That is, it defines a concept as the output of a hidden sampling procedure that generates data items. Such generative approaches have a rich history in computational approaches to vision (Leyton, 1999; Stiny & Gips, 1972; Yuille & Kersten, 2006). Here, although we are working in a visual domain, we apply this approach to abstract concepts in a domain-general way.

To get a sense for the mechanics of such generative rules, consider the *ABA* concept used in our experiment. A generative rule for this concept might look as follows:

1. Randomly pick a part from the set of all possible parts
2. Remove that part from the set of all possible parts
3. Randomly pick a second part from this new set
4. Produce an object whose parts, from left to right, are the first part, followed by the second part, followed by the first part again

This procedure for generating three-part objects is capable of outputting all and only those that follow an *ABA* pattern. Therefore, a learner who has inferred this procedure as the underlying rule has knowledge of a possible causal origin of *ABA* exemplars. A learner can use this knowledge to classify a novel object as a member of the *ABA* concept by determining whether the object is a possible output of the procedure.

We represent generative sampling procedures as probabilistic programs. Probabilistic programs have two computational properties that make them particularly powerful for dealing with abstract concepts. First is stochasticity. While traditional programs are deterministic, ours are probabilistic, meaning they employ as computational primitives functions that sample from distributions. Therefore, rather than a single output, a program might be able to produce multiple outputs—a different output each time the program is executed. In the concept learning literature, a concept is defined by its extension, or the set of entities in the world which it refers to. The set of possible outputs of a program forms a natural representation of the extension of the concept it characterizes.[2] Second is variable binding. As discussed above, abstract concepts are those that require variable binding. Probabilistic programs are similar to conventional computer programs in the sense that they have variable binding built in. Consequently, the HLOT model can naturally accommodate variable binding and, therefore, generative procedures representing abstract concepts.

We base our representation of probabilistic programs on lambda calculus. Lambda calculus is a type of logic that characterizes computation using function abstraction (via variable binding) and function application (via symbolic substitution). These two operations are sufficient to make lambda calculus a universal model for computation (i.e., it is equivalent in power to a Turing machine). Since our implementation uses stochastic functions that sample from distributions, it can be considered a form of stochastic lambda calculus. In addition to variable binding and function application, programs are made up of existing (either innate or previously learned) cognitive operations referred to as *primitives*. Each

[2] In other settings, it may be best to only consider items whose output probability exceeds a threshold as members of a concept.

primitive is simple, but primitives can be composed to build arbitrarily complex programs.

For the purposes of accounting for our experimental data, we provided the HLOT model with three program primitives: `sample()` which samples uniformly from a set, – (minus) which removes an element from a set, and `concat()` which concatenates strings. We chose these primitives because they constitute a minimal set necessary to construct reasonable hypotheses in this domain. Given our model's simplicity bias (discussed shortly), it is reasonable to think that even with a larger set of primitives, learned hypotheses would tend to use a minimal set like the one we have provided. These primitives are both simple and cognitively plausible, corresponding to manipulations that could be learned early in development by manipulating objects, such as removing something from a group or placing two objects near one another.

Our programs output string representations of objects. Each letter of a string stands for a particular part, and arrows denote spatial relationships between parts. For simplicity, we assume a fixed, deterministic mapping between the string representation of an object and its visual image. For example, all parts have the same size and orientation in all exemplars and test items, and we assume the learner knows this mapping. We discuss later (Section 6) the potential effects of relaxing this assumption.

Pseudocode equivalent to the above procedure for representing the *ABA* concept is the following[3]:

Program 1.

```
let x1 = sample(Σ)
let x2 = sample(Σ − x1)
output x1 → x2 → x1
```

where Σ is the set of all possible parts. This probabilistic program represents the *ABA* concept because it outputs all and only those objects that follow an *ABA* pattern. Its possible outputs are exactly the extension of the concept.

The HLOT model is a Language of Thought model in that it defines an infinite space of possible hypotheses (i.e., possible stochastic programs or stochastic lambda calculus expressions) by using a probabilistic context-free grammar (PCFG) (Goodman et al., 2008). PCFGs are a well known formalism from Computer Science where they have been successfully employed for many natural language tasks. They are generalizations of (non-probabilistic) CFGs, which are used to define the syntax of programming languages. Just as these grammars are used to characterize natural and artificial languages, the HLOT model uses them to characterize conceptual languages. A PCFG is made up of a set of production rules that define how to build up arbitrarily complex expressions.

We defined a grammar to generate expressions of the following form:

1. Generate and store a part
2. Generate and store zero or more additional parts
3. Using those parts, output a string representation of an object

The parts can either be sampled from a set (via `sample()`) or specified directly (part '*a*', for example). Sets can consist of either the full alphabet or any subset thereof, constructed by removing parts from the full alphabet. The object's string representation is then constructed using the variables bound in the previous steps. The full grammar is shown in Fig. 2.

Given some data—one or more exemplars from a concept—the model uses Bayesian inference to learn a distribution over stochastic programs, indicating which programs were likely or unlikely to have generated the data through their sampling process. We can express this distribution using Bayes' rule:

$$P(h|x_1,\ldots,x_n) \propto P(x_1,\ldots,x_n|h)\,P(h)$$

where h denotes a hypothesized rule and $x_1,\ldots,x_n$ denotes the set of exemplars comprising the data set. The term $P(h|x_1,\ldots,x_n)$ is known as the posterior, and it is proportional to the product of the terms $P(x_1,\ldots,x_n|h)$ and $P(h)$ which are known as the likelihood and prior, respectively. The prior is a distribution defined before observing the data set, whereas the posterior is the updated distribution calculated after observing the data set. In our model, the likelihood and prior distributions correspond to the two levels of the generative hierarchy.

The likelihood function $P(x_1,\ldots,x_n|h)$ specifies the probability of seeing the observed evidence if hypothesis h is true. This expression has a natural interpretation in our model because h is represented by a probabilistic program which inherently defines a distribution over its outputs. For example, let h_1 denote Program 1, which generates all and only those objects consistent with the concept *ABA*. If there are five possible object parts, and a learner observes data item $x_1 = a \rightarrow b \rightarrow a$, then the likelihood of that data item $P(x_1|h_1) = \frac{1}{5} \times \frac{1}{4} = \frac{1}{20}$. Next, consider hypothesis h_2 that is identical to h_1 except that the middle part is not constrained to be different from the outer parts. This would be represented by the following program:

Program 2.

```
let x1 = sample(Σ)
let x2 = sample(Σ)
output x1 → x2 → x1
```

In this case, the likelihood is $P(x_1|h_2) = \frac{1}{5} \times \frac{1}{5} = \frac{1}{25}$. This example illustrates that the likelihood function implements the "size principle"[4] (Tenenbaum, 1999). Hypothesis h_2 yields a lower likelihood value than h_1 because it must assign some probability mass to every item it can generate, and since it is a "wider" hypothesis and can generate more items (25 versus 20), each item is allocated less of that overall mass. In this way, the likelihood function prefers "smaller" hypotheses because they make each individual data item more probable.

Of course it is unlikely that learners explicitly track the full extension of their hypotheses; our claim is not that this is the mechanism by which learners compute, but rather that the Bayesian likelihood describes an ideal or optimal value, even if it is intractable to compute exactly. In practice, both large-scale machine learning models and human learners must approximate such values, but it is useful for ideal-observer models such as ours to specify the optimal quantity so that further work may have a well-founded target when researching approximation strategies. (See Frank (2013) for a fuller discussion of different levels of analysis as it relates to the size principle.)

[3] For ease of reading, we present these stochastic lambda calculus expressions in a procedural style, with line breaks, an explicit `output` function, and variable binding written as "`let` $x = \cdot$". These are simply syntactic rewritings, and do not change the semantics. For example, Program 1 is equivalent to the expression: $(\lambda x_1.\lambda x_2.x_1 \rightarrow x_2 \rightarrow x_1)(\texttt{sample}(\Sigma - x_1))(\texttt{sample}(\Sigma))$.

[4] More precisely, the size principle emerges from a generative model's likelihood function in the special case where that function is uniform over exemplars.

$$
\begin{aligned}
\textbf{START} &\rightarrow \texttt{let } \texttt{<}\textbf{BV\_PART}\texttt{>}{:}x_1 = \textbf{FIRST\_PART}; \textbf{EXPR} && \\
\textbf{EXPR} &\rightarrow \texttt{let } \texttt{<}\textbf{BV\_PART}\texttt{>}{:}x_n = \textbf{PART}; \textbf{EXPR} && \\
&\rightarrow \textbf{STRING} && \\
\textbf{STRING} &\rightarrow \textbf{BV\_PART} && \\
&\rightarrow \textbf{STRING CONNECT STRING} && \\
&\rightarrow \{\textbf{STRING}\} && \\
\textbf{FIRST\_PART} &\rightarrow \texttt{sample(}\textbf{FIRST\_SET}\texttt{)} && 1 - p_{single} \\
&\rightarrow \textbf{SINGLE} && p_{single} \\
\textbf{PART} &\rightarrow \textbf{BV\_PART} && (1 - p_{single})/2 \\
&\rightarrow \texttt{sample(}\textbf{SET}\texttt{)} && (1 - p_{single})/2 \\
&\rightarrow \textbf{SINGLE} && p_{single} \\
\textbf{FIRST\_SET} &\rightarrow \Sigma && 1 - p_{minus} \\
&\rightarrow \texttt{minus(}\textbf{FIRST\_SET}\texttt{, }\textbf{FIRST\_PART}\texttt{)} && p_{minus} \\
\textbf{SET} &\rightarrow \Sigma && 1 - p_{minus} \\
&\rightarrow \texttt{minus(}\textbf{SET}\texttt{, }\textbf{BV\_PART}\texttt{)} && p_{minus} \\
\textbf{CONNECT} &\rightarrow \text{`}\uparrow\text{'} \mid \text{`}\downarrow\text{'} \mid \text{`}\leftarrow\text{'} \mid \text{`}\rightarrow\text{'} && \\
\textbf{SINGLE} &\rightarrow \text{`a'} \mid \ldots \mid \text{`e'} &&
\end{aligned}
$$

Fig. 2. The HLOT model uses a probabilistic context-free grammar to define the space of stochastic lambda calculus expressions. Here we show an equivalent grammar that is easier to read. Non-terminals are indicated by **BOLD-CAPS**. The notation "let **<TYPE>**: $x_n = \ldots$" means that when this rule is expanded in a derivation, a new rule is created in the grammar: **TYPE** $\rightarrow x_n$. Where indicated, rules have production probabilities which were fit to data. All other production probabilities are uniform, or maximally so without resulting in an improper grammar.

When a data set consists of multiple exemplars from a concept, the HLOT model assumes that the exemplars are conditionally independent. Thus, the likelihood function is given by:

$$P(x_1, \ldots, x_n|h) = \prod_i P(x_i|h).$$

The prior distribution $P(h)$, following Goodman et al. (2008), uses the probabilities from the PCFG to define a distribution over hypotheses. The grammar consists of a set of rules, each of which has an associated probability. Expressions are built up by repeated application of these rules. This results in a tree structure known as a parse tree, where the leaves of the tree represent the content of the expression. For example, Fig. 3 shows a parse tree for Program 1. For our simulations, we set the grammar's production probabilities as uniformly as possible under the constraint that the expected length of the resultant expressions should be finite. If there are p production rules associated with a given nonterminal, this value was generally $\frac{1}{p}$ for each of those rules, except in cases where this would result in an improper grammar (the expected length of its generated expressions is infinite).

There were two rules that we felt corresponded to biases that subjects may be bringing into the task. Therefore we set their production probabilities to free parameters and fit them to data as discussed in the Appendix. One parameter represented the prior probability of choosing a specific part as opposed to randomly sampling one (p_{single}) and the other parameter represented the prior probability of removing a part from a set (p_{minus}).

Importantly, this prior implements a bias toward hypotheses that are "simpler". Since each production rule in a PCFG has an associated probability (less than 1), the HLOT model's prior distribution favors simplicity by defining the probability of expression h as the product of the production probability for each rule in its parse tree:

$$P(h) = \prod_{r \in \mathscr{T}} P_G(r)$$

where r is a production rule used in parse tree $\mathscr{T}$, and $P_G(r)$ is the associated probability for rule r as given by grammar G. Thus, more complex expressions employing more applications of production rules have more factors in this product, and therefore have lower probability.

The combination of the likelihood as defined by the probabilistic program and the prior as defined by the grammar form a principled balance between fit to the observed data and generalization to unseen data items. The likelihood function prefers "smaller" hypotheses providing the best fit to data, even if those hypotheses are complex. The prior, because of its simplicity preference, counteracts "overfitting" and ensures that the hypothesis generalizes to novel data items. However as the model receives more data, the likelihood will begin to dominate the prior and the model may come to prefer more complex hypotheses. Thus, the hypotheses with the highest posterior probabilities will be those that best trade off model fit to the available data, as measured by the likelihood function, with model simplicity and generalizability as defined by the prior distribution.

This subsection has described the probabilistic rules and their generating process for the HLOT model. To complete the description of the model, we also describe two variables that influence how the model generalizes. In visual environments, people often show orientation invariance in their object recognition performances (Attneave, 1955; Liu & Kersten, 2003). The first variable determines whether hypotheses should be orientation invariant, meaning, for example, that a probabilistic rule that produces the object $a \rightarrow d \rightarrow d$ also produces $d \rightarrow d \rightarrow a$.[5] In addition, people may differ in terms of how broadly they generalize due to differing notions of the space of possible object parts. The second additional variable of the HLOT model determines whether the set of possible object parts is the full set of possible parts or if it is limited to just the parts that occur in a set of training exemplars. We model these

[5] This invariance is at the symbolic level, and may not apply at the visual level when the parts themselves are not symmetrical.

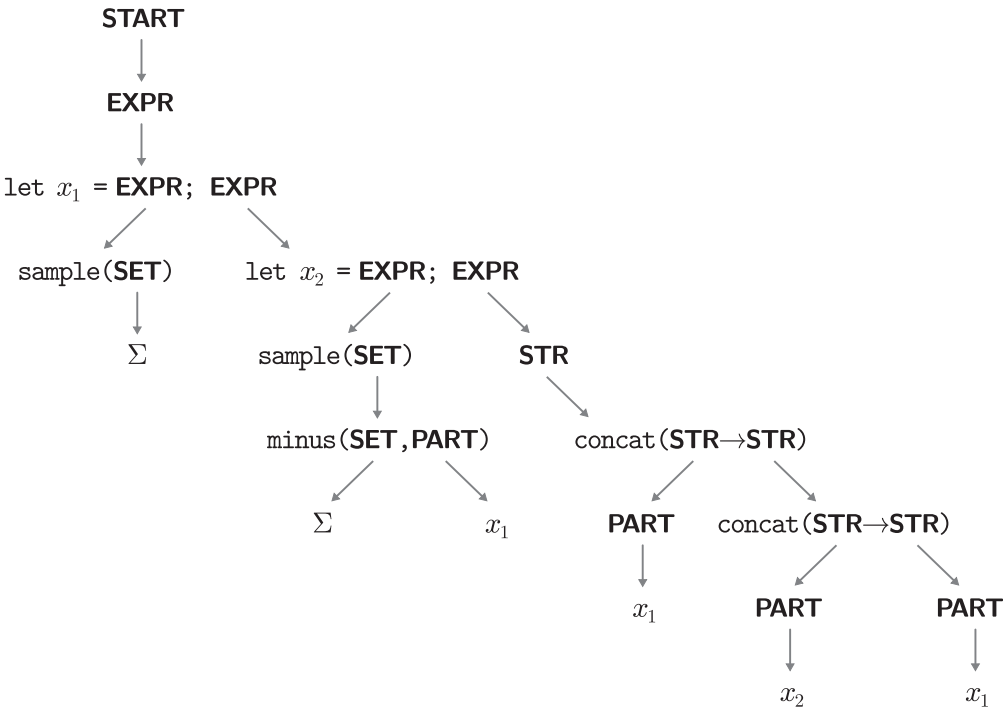

Fig. 3. A simplified parse tree for the pseudocode shown in Program 1 for the concept *ABA*. Each arrow corresponds to the application of a grammatical production rule. The production rules define constraints on what grammatical transformations are allowed (i.e., on the syntax of pseudocode in this example), along with an associated probability with each transformation (not shown here).

choices as part of the generative process via a vector of Bernoulli random variables, denoted $\vec{\theta}$. This makes the full posterior distribution:

$$P(h, \vec{\theta}\,|x_1, \ldots, x_n) \propto P(x_1, \ldots, x_n|h, \vec{\theta})P(h)P(\vec{\theta}).$$

We assume that the choices of orientation invariance and set content are independent, giving:

$$P(\vec{\theta}) = \prod_{i \in \{oi,\, rp\}} p_i^{\theta_i}(1 - p_i)^{1-\theta_i} \quad (1)$$

where *oi* stands for "orientation invariance" and *rp* stands for "restricted parts", and where p_{oi} and p_{rp} are free parameters that we fit to our subjects' responses.

The Appendix describes how we estimated the posterior distribution over hypotheses. In brief, we did so in two steps. The model has four free parameters: the two production probabilities p_{single} and p_{minus}, and the parameters p_{oi} and p_{rp}. First, the values of these parameters were set to an initial guess. Upon observing some exemplars from a target concept, the model then estimated a posterior distribution over hypotheses using a Markov chain Monte Carlo inference procedure. Samples generated by this procedure were placed on a list of viable hypotheses. Next, we fit the values of the free parameters by minimizing the sum of squared error between model prediction and subjects' responses as reported in Section 4.2. We then recomputed the posteriors of the viable hypotheses using these fitted values.

3.2. Competing models

We compared the HLOT model with two variants of an exemplar model based on the Generalized Context Model (GCM) (Nosofsky, 1986), a highly influential model of human categorization. The GCM is a similarity-based model—it determines the category membership of a test item based on its similarity (or inverse of distance) to the training exemplars. The GCM is a useful comparison model because it does not have a native mechanism for abstraction or variable binding.

The GCM's probability of responding 'yes' to test item k (that is, its probability of judging test item k as being a member of the same concept as the training exemplars), denoted r_k, is:

$$P(r_k = \text{'yes'}|x_1, \ldots, x_n) = \frac{\sum_i e^{-c \cdot d(y_k, x_i)}}{\max_j \sum_i e^{-c \cdot d(y_j, x_i)}}$$

where $d(\cdot, \cdot)$ is a distance function, c is a scaling parameter, x_i is the ith training exemplar, and y_j is the jth test item. In the simulations reported below, we used gradient descent to find a value for scaling parameter c that minimized the sum of squared error between the model's responses and subjects' responses in our behavioral experiment. This formulation of the GCM differs slightly from the standard one because the GCM was originally designed to discriminate between two classes, but there is only a single class in our experimental task. Therefore, we normalized the similarity scores by dividing them by the maximum similarity across all test items. This converts a raw similarity score to a pseudo-probability in the interval [0, 1] (see Stuhlmüller et al. (2010) for a related similarity score). We implemented two different versions of the GCM, each with a different distance function.

The first version, referred to as GCM-String, uses a "symbolic" distance function. For this function, we chose string edit distance, also known as Levenshtein distance (Levenshtein, 1966). This is a metric on strings that gives the minimum number of edits (single character insertions, deletions, or substitutions) required to transform one string into another. Applied to a string representation of our objects, this distance measure depends on the parts and their positions in the object. The lowest distance (and thus highest similarity) is assigned to pairs of strings that are exactly matching, as they require zero edits to transform one into the other. Higher distances are assigned to a pair of strings representing objects sharing fewer parts and where those parts are in differing positions.

ABA

```
let x1 = sample(Σ_R)
let x2 = sample(Σ_R − x1)
output x1 → x2 → x1
```

xBB

```
let x1 = sample(Σ)
let x2 = 'a'
output x2 → x1 → x1
```

ABC

```
let x1 = sample(Σ)
let x2 = sample(Σ − x1)
let x3 = sample(Σ − x2 − x1)
output x2 → x3 → x1
```

Ring

```
let x1 = sample(Σ_R)
let x2 = sample(Σ_R − x1)
output x1 → ((x2 ↑ x1) ↓ x1) → x1
```

Fig. 4. Hypotheses assigned the highest posterior probabilities by the HLOT model in each experimental condition. Σ denotes the set of all object-parts, and Σ_R denotes the set of parts that are present in the training exemplars.

The second version, referred to as GCM-CNN, implements a "visual" distance function. Here, we used a highly successful computer vision system based on a deep convolutional neural network (CNN). This system, AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), is an eight-layer (five convolutional, three fully connected layers) CNN trained on 1.2 million images in the ImageNet data set. We used the pre-trained network provided by the Caffe framework (Jia et al., 2014). It achieved the best performance on the 2012 ImageNet Large Scale Visual Recognition Challenge, and was in large part responsible for the recent surge of interest in deep neural networks for computer vision. To compute our distance function, we first used AlexNet to obtain a visual representation of our objects. For each object, we provide an image of that object to the input of the network. We then extract a vector representation of that image from the activations of the units in the network's last fully-connected layer before the output layer (layer *fc7* in the notation of Krizhevsky et al. (2012)). The distance between two objects was then calculated as cosine distance, or 1 minus the normalized dot product, between the vectors representing the objects. This technique of using visual features from deep CNNs for classification tasks has recently been shown to be equal to or better than other commonly used techniques (Razavian, Azizpour, Sullivan, & Carlsson, 2014).

4. Comparison of models and human generalizations

When simulating the HLOT model, we treated it as if it was a subject in our experiment: for each experimental condition, we provided it with the same three training exemplars as observed by our subjects, and we tested it using the same set of twenty-four test items as used to test our subjects.

4.1. Inferred hypotheses

We start by reporting the hypotheses to which the model assigned the highest posterior probability for each experimental condition (see Fig. 4). For the condition using concept *ABA*, the model assigned the largest probability to a hypothesis that follows the *ABA* pattern except that the set of possible object parts is restricted to the parts that appear in the training exemplars. The same hypothesis but without this restriction was assigned a similar probability. To us, these are both sensible hypotheses given the observed training exemplars. For the condition using concept *xBB*, the model assigned the highest probability to the hypothesis stating that objects consist of the specific part represented by the string '*a*' followed by any two identical parts. Here, the model correctly inferred a relatively "narrow" hypothesis stating that all objects belonging to the concept have part '*a*' as their leftmost part. For concept *ABC*, the model correctly inferred a hypothesis that all three parts differ. Lastly, for concept *Ring*, the model inferred a hypothesis that follows the *Ring* pattern except that the set of possible object parts is restricted to the parts appearing in the exemplars. This hypothesis is analogous to the one that it inferred in the *ABA* condition, verifying that the model is capable of operating with stimuli that go beyond short, linear strings. Overall, we find that the model, using only the data subjects observed, infers programs that are much like those we had intuitively expected when constructing the experiment.

4.2. Comparison with subjects' responses

Next we compare the HLOT model's responses with subjects' responses in our experiment. For test item k, we computed the model's marginal probability (marginalizing over hypotheses) that k would be in the concept by summing the posterior probabilities for all hypotheses that produce k:

$$P(r_k = \text{'yes'} | x_1, x_2, x_3) = \sum_{h, \vec{\theta}} P(h, \vec{\theta} \,|x_1, x_2, x_3)\, I_{ext(h, \vec{\theta})}(k)$$

where r_k is the response to test item k, $ext(h, \vec{\theta})$ is the extension of (i.e., the set of objects generated by) hypothesis h and generalization variables $\vec{\theta}$, and I is the indicator function equal to 1 if test item k is in the extension and equal to 0 otherwise. The results are shown in Fig. 5. Taken as a whole, the HLOT model captures the qualitative trends in subjects' generalizations. For example, in all conditions, both the model and subjects assigned the highest probabilities to test items that follow the target concept and the lowest probabilities to items that were strongly inconsistent with the target concept. This suggests that subjects learned an underlying abstract rule even with just three training exemplars.

The HLOT model also captures some of the finer gradations in subjects' generalizations. For example, in the *ABA* condition, subjects rated test items that have all the same parts (e.g., $a \to a \to a$) as more likely than items such as $e \to a \to a$ and $c \to d \to b$. The model captures this trend via hypotheses like the one shown in Program 2 which lacks the constraint that the middle part differs from the outer parts. This program generates all items that follow an *ABA* pattern, but additionally generates items in which all three parts are the same. This hypothesis has a lower likelihood score than that of Program 1 because it generalizes more widely, but it has a higher prior probability because it is less complex. The model correctly predicts that, due to hypotheses like Program 2, people will more readily generalize to items that have all the same parts than to items where the first and third parts differ.

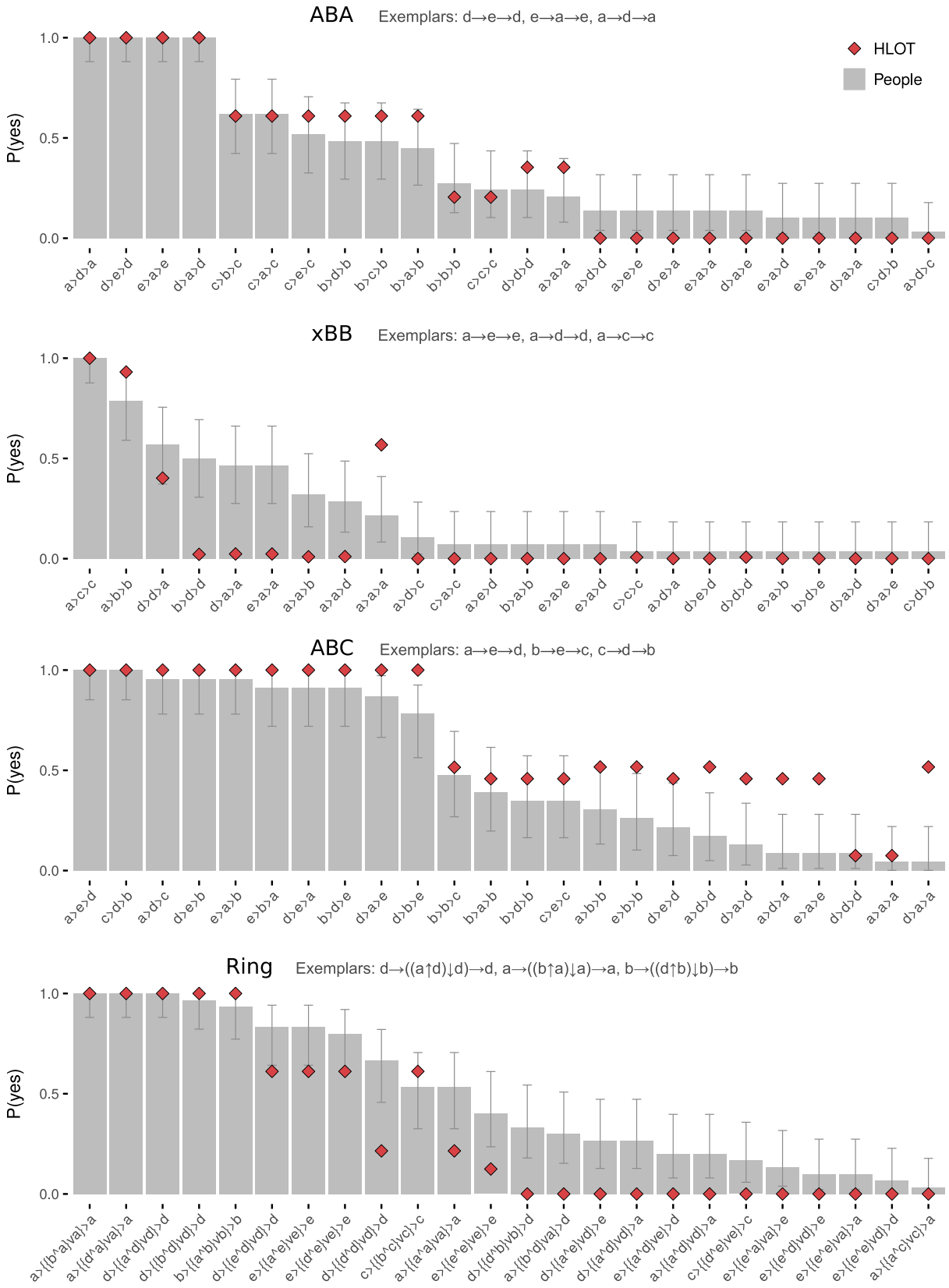

Fig. 5. HLOT model's and subjects' probabilities of responding 'yes' to each test item in the *ABA*, *xBB*, *ABC*, and *Ring* conditions, respectively. Test items are listed along the horizontal axis, and probability of responding 'yes' is plotted along the vertical axis. Subjects' probabilities are given by the gray bars, and the model's probabilities are given by the red diamonds. Error bars show 95% confidence intervals given by the exact binomial test (Clopper & Pearson, 1934). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

These finer aspects of subjects' generalization trends would not be captured by a simpler formulation of model response, such as the hypothesis or program with the largest posterior probability (the maximum *a posteriori* or MAP estimate, as in Stuhlmüller et al. (2010)). This is because even though hypotheses such as Program 2 are non-optimal, they still represent a notable portion of the total posterior mass and therefore play an important role in overall generalization trends. For another example, note that the MAP hypothesis in the *xBB* condition (Fig. 4) does not have the parameter for orientation invariance set to *true*. Nonetheless, test item $d \rightarrow d \rightarrow a$, which would most plausibly be generated by an orientation-invariant hypothesis, was chosen by people more than half of the time (Fig. 5). These examples suggest that, given limited evidence, people are able to discover a diverse range of plausible hypotheses, and therefore it is crucial for models to explore the full posterior and to use this posterior to average over all hypotheses.

Although the model gives a good account of subjects' responses, it sometimes generalizes in ways that differ from our subjects' generalizations. For instance, in the *xBB* condition, the model gives a lower probability than people to items that follow an *ABB* pattern but do not have an '*a*' in the first position. As expected based on the data of Gerken (2006), people give lower scores to those items than they do to those with an '*a*' in the first position, but the model gives near-zero probability, a much stronger effect. This is due to the much lower likelihood given to hypotheses that allow any part in the first position.

4.3. Comparison with alternative models

As described above, we also applied two competing models, GCM-String and GCM-CNN, to our experimental task. Fig. 6 shows, for all models and experimental conditions, the correlation between each model's predicted probability of responding 'yes' to a test item and subjects' actual probability. The HLOT model achieves the highest score in all conditions (indicated by the bold font). In all conditions, the difference between HLOT and the GCM models was statistically significant at the $p < .05$ level by a two-tailed *t*-test on the Fisher's *z*-transformed correlations.[6]

These correlations show that the HLOT model provides an excellent account of subjects' responses, with most correlations above 0.9. Despite the sophistication of the GCM-String and GCM-CNN models, they provide relatively poor accounts of subjects' responses in all conditions, explaining roughly half of the variance that the HLOT model accounts for.

To further characterize the differences in performances between the HLOT, GCM-String, and GCM-CNN models, we examined their responses to three groups of test items: (1) *in_restricted*: items that follow the intended pattern and have no parts other than those seen in the exemplars; (2) *in_novel*: items that follow the intended pattern but have one or more parts that were not seen in the exemplars; and (3) *other*: items that do not follow the example pattern. These results are shown in Fig. 7. In all experimental conditions, subjects' probabilities were highest for items that followed the given pattern and had only parts that were present in exemplars. The probabilities took an intermediate value for items that followed the given pattern but contained novel parts not seen in exemplars. Lastly, the probabilities were lowest for items that do not follow the given pattern. These results indicate that subjects did indeed learn a close approximation to the intended rule.

The responses of the HLOT model capture subjects' generalization trends, with high probability given to test items that follow the structural pattern of the exemplars and low probability given to items that do not. This pattern of generalization is markedly different from that of the GCM-String and GCM-CNN exemplar models. The GCM-String model prefers items that have the same parts in the same positions as the exemplars, and the GCM-CNN model prefers items that have the most shared visual features with the exemplars (where the features were determined by the deep convolutional neural network AlexNet). Since neither of their similarity functions are sensitive to the abstract, relational structure of the objects, these models do not generalize in the same way as people.

	ABA	*xBB*	*ABC*	*Ring*
HLOT	**0.99**	**0.79**	**0.94**	**0.94**
GCM-String	0.44	0.34	0.27	0.65
GCM-CNN	0.36	0.42	0.49	0.62

Fig. 6. Correlations with human responses for the HLOT model and the two similarity-based models. The best score in each condition is shown in bold.

In particular, the similarity-based models tend to underpredict "yes" responses for the *in_novel* set of test items, those that follow the same pattern as the exemplars but contain novel parts. For example, in the *ABA* condition, given exemplars $d \rightarrow e \rightarrow d$, $e \rightarrow a \rightarrow e$, and $a \rightarrow d \rightarrow a$, the HLOT model assigns a high probability to test items such as $b \rightarrow c \rightarrow b$ which, although it is solely made up of parts that were not seen in the exemplars, reflects the exemplars' abstract structure. Roughly half of the subjects responded 'yes' to this test item in this condition. The similarity-based models, however, assign that item a low probability, as it requires many edits to transform to a training exemplar and also shares few visual features with the exemplars. Thus, this test item shares its abstract, second-order structure with the exemplars, but it differs in its concrete and particular features.

This difference in model responses can go in the other direction as well. The similarity-based models tend to overpredict "yes" responses for out-of-pattern items, since these can have similar surface characteristics to the examples without following the abstract pattern. Continuing with the *ABA* condition, test item $d \rightarrow e \rightarrow a$ does not reflect the abstract, relational structure of the training exemplars, since all three of its parts are different from each other, but it does share at least some parts (and also visual features) with all of the exemplars. Therefore, the HLOT model gives it a low probability but the GCM models give it a high probability. The results of the experiment show that, as predicted by the HLOT model, subjects responded 'yes' to this test item with a low probability, suggesting that people can and do learn abstract, relational concepts when these concepts provide good accounts of the data.

4.4. Discussion

These results show that not only is the HLOT model able to infer programs with variables, it does so in a way that better matches human generalizations than standard existing models. We believe that the overall superior performance of the HLOT model is primarily due to its ability to infer the abstract compositional (or relational) structure underlying the training exemplars. Moreover, because the model does not infer a single hypothesis regarding this structure—it infers a distribution over all possible hypotheses—the model simultaneously considers multiple hypotheses, albeit each one to a lesser or greater extent (based on a hypothesis's posterior probability). These results suggest that subjects, even when exposed to only three training exemplars, generalize in a way that, like the HLOT model, reflects objects' abstract structure rather than their surface-level similarities.

[6] We accounted for the dependency induced by calculating correlations with respect to the same target (subjects' responses) by modifying the *z*-transform as per Steiger (1980).

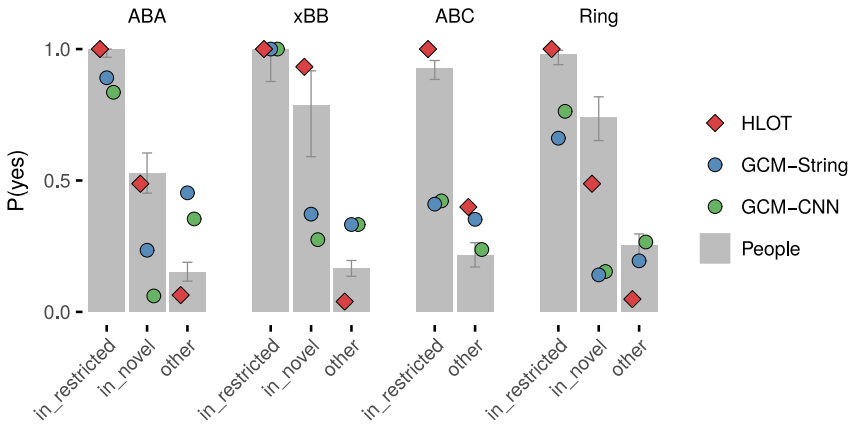

Fig. 7. Total subject and model probabilities of responding 'yes' across all test items in each of the three groups (see text for a description of the items in each group). (In the *ABC* condition, all five possible parts were present in the training exemplars and, thus, there are no test items in the *in_novel* group.) Shown are the HLOT model and the two models based on the GCM. Error bars show 95% confidence intervals given by the exact binomial test (Clopper & Pearson, 1934).

$$
\begin{aligned}
\textbf{START} &\rightarrow \textbf{EXPR} \\
\textbf{EXPR} &\rightarrow \texttt{let} <\textbf{PART}>{:}x_n = \textbf{PART}; \textbf{EXPR} \\
&\rightarrow \textbf{STRING} \\
\textbf{STRING} &\rightarrow \textbf{PART} \\
&\rightarrow \textbf{STRING CONNECT STRING} \\
&\rightarrow \{\textbf{STRING}\} \\
\textbf{PART} &\rightarrow \texttt{sample(SET)} && 1 - p_{single} \\
&\rightarrow \textbf{SINGLE} && p_{single} \\
\textbf{SET} &\rightarrow \Sigma && 1 - p_{minus} \\
&\rightarrow \texttt{minus(SET, PART)} && p_{minus} \\
\textbf{CONNECT} &\rightarrow \text{`}\uparrow\text{'} \mid \text{`}\downarrow\text{'} \mid \text{`}\leftarrow\text{'} \mid \text{`}\rightarrow\text{'} \\
\textbf{SINGLE} &\rightarrow \text{`a'} \mid \ldots \mid \text{`e'}
\end{aligned}
$$

Fig. 8. The grammar for the optional-abstraction model variant. See Fig. 2 for details of the syntax.

5. The nature of abstraction

An advantage of the HLOT model is that it uses computational primitives and grammatical structures that are psychologically interpretable. This gives us the ability to use the model to reason about psychological phenomena by implementing different assumptions and then investigating the effect that those assumptions have on the models' performances. We leveraged this feature to investigate the nature of abstraction in people's learned concepts.

For example, the basic HLOT model's grammar implements the constraint that the output string must be made up of previously bound variables (e.g., "$x_1 \rightarrow x_2 \rightarrow x_1$"). To probe the question of whether this assumption accurately reflects people's mental representations, we implemented a variant of the HLOT model in which this restriction is relaxed. That is, the model variant can produce strings, such as "$x_1 \rightarrow \texttt{sample}(\Sigma) \rightarrow \texttt{sample}(\Sigma - x_1)$", containing terms that are not bound variables (e.g., $\texttt{sample}(\Sigma)$). By endowing this model variant with a different prior structure, it obtains different biases, thereby favoring and disfavoring different hypotheses. Even though the basic model and its variant have different prior structures, both still impose an overall simplicity bias. This model variant, which we call the optional-abstraction variant, implements a sampling procedure of the following form:

1. Generate and store zero or more parts
2. Output a string representation of an object (using parts represented by bound variables or ones computed on the fly)

We defined the grammar in Fig. 8 to output such expressions.

A summary of the performance of the basic HLOT model and the optional-abstraction variant is shown in Fig. 9. While these performances are largely the same in the *ABA*, *xBB*, and *Ring* conditions, the differences between the versions are pronounced in the *ABC* condition. Here, both versions predict roughly the same high probabilities for test items consistent with the *ABC* pattern. For other test items, however, the optional-abstraction variant assigns higher probabilities than both people and the basic model.

Why does making abstraction (i.e., variable binding) optional result in broader generalization? The answer is illustrated by examining the hypotheses with the highest posterior probabilities inferred by each version. In the basic model, the maximum-a posteriori hypothesis generates the *ABC* pattern (bottom left of Fig. 4). For the optional-abstraction version, however, the maximum-a posteriori hypothesis is

$$\texttt{output}\ \texttt{sample}(\Sigma) \rightarrow \texttt{sample}(\Sigma) \rightarrow \texttt{sample}(\Sigma)$$

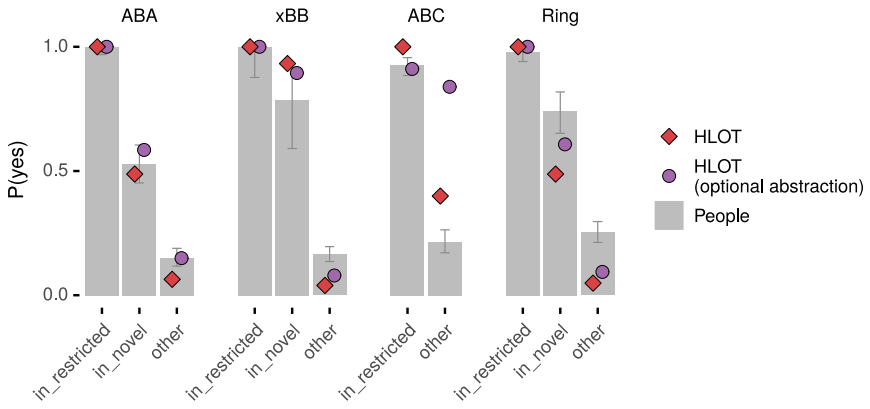

Fig. 9. Total subject and model probabilities of responding 'yes' across all test items in each of the three groups. (In the *ABC* condition, all five possible parts were present in the training exemplars and, thus, there are no test items in the *in_novel* group.) Shown are the basic HLOT model and its optional-abstraction variant. Error bars show 95% confidence intervals given by the exact binomial test (Clopper & Pearson, 1934).

which generates all possible three-part items. The reason the optional-abstraction version prefers this hypothesis is because it can be expressed using exactly zero abstraction operations. Since it imposes no constraints on an object's parts, it does not require any self-reference and, therefore, does not require any variables to refer to individual parts. Although this hypothesis has a lower likelihood score than the *ABC* hypothesis, that factor is offset by the much greater prior probability it obtains by avoiding the additional complexity of abstractions. Conversely, when abstraction happens automatically (as in the basic HLOT model), all valid hypotheses include those variable-binding operations. Consequently, the cost in the prior of that additional complexity is common to all hypotheses, and therefore this cost does not contribute to distinguishing between hypotheses (technically, this cost gets factored out in normalization).

A prior structure in which abstractions happen automatically has the unintuitive result of making abstraction operations appear to be "free" in the prior. Indeed, when looking at the results in the *ABC* condition, it appears that people very readily form hypotheses with three distinct parts, and that they give very low scores to all other hypotheses. This suggests that they impose very little prior penalty on hypotheses with additional variables, indicating that variable binding in people seems to be virtually free. This result is compatible with a simplicity prior over hypotheses only when variable binding happens automatically, as in the basic HLOT model.

Another way to characterize the difference between the hypothesis spaces of the two model variants is in terms of whether cognitive variables are *local* or *global*. The prior structure of the optional-abstraction variant allows for expressions that include computations but no abstractions (e.g., `output sample(`Σ`)` → `sample(`Σ`)` → `sample(`Σ`)`). Here, the result of an individual `sample(·)` operation is effectively local to its corresponding slot in the string. However, it seems intuitive that a cognitive system would prefer to make the results of any sub-computations globally available for reasons of efficiency, thereby avoiding redoing work. The basic HLOT model effectively implements this assumption by requiring that subcomputations be saved before they are used, just in case the results of those subcomputations are subsequently needed (as in `output` $x_1 \rightarrow x_2 \rightarrow x_1$).

5.1. Discussion

Our use of a formal computational model allows us to test variants that impose different assumptions on the nature of abstraction. We find that the best HLOT model is one in which abstraction happens automatically, meaning that any subcomputations performed in the construction of a rule are globally available to the rest of the rule. This suggests that even though hypothesis complexity is a major influence on rule-based concept learning (Feldman, 2000), the automatic nature of abstraction may essentially "cancel out" its contribution toward complexity, making it appear as if abstraction is "free". If so, this may explain why people fluidly and eagerly learn abstractions from simple data sets like a few instances of *ABA* strings.

6. General discussion

In summary, we have formalized the HLOT model, an inductive model of probabilistic rule and variable learning. Our approach follows the emerging framework known as the probabilistic language of thought (pLOT), which has sought to account for human perception and cognition through models that combine symbolic and statistical approaches (Erdogan et al., 2015; Goodman et al., 2008; Kemp, 2012; Piantadosi et al., 2012, 2016; Ullman et al., 2012; Yildirim & Jacobs, 2015). Due to its symbolic nature, the model can learn abstract rules containing variables. Due to its statistical nature, the model can use Bayesian inference and programs with stochastic primitives to learn distributions over rules indicating which rules are relatively likely or unlikely to underlie a set of data items.

To evaluate the HLOT model, we conducted an experiment in which human subjects viewed training items and then judged which test items belong to the same concept as the training items. We found that the model provides a close match to human generalization patterns, significantly outperforming two variants of the GCM model, one variant making judgments based on string similarity and the other based on visual similarity using features from a deep convolutional neural network. Although GCM models perform well in other tasks, our experiment highlights a key limitation of these systems: they do not have adequate mechanisms for handling variables. Our task was constructed to encourage variable use precisely because we hypothesized that, when given the opportunity, people would use variables in a way that cannot be captured by previous accounts. In combination, our results formalize, test, and strongly support the view that the ability to define and use variables is a central capacity in human thinking (Marcus, 2003). Moreover, by using variable assignment, a learner can account for rules that go beyond the concrete features in the stimuli and reflect abstract or second-order relationships.

To our knowledge, ours is one of the first models using the pLOT framework to infer probabilistic programs—programs with stochastic primitives—from data. The use of probabilistic programs has the benefit of generalizing some assumptions built into previous models. For example, with deterministic rules, in order to define a likelihood function it is common to assume uniformity over all items consistent with the rule (e.g. Goodman et al. (2008)). This simple assumption is clearly insufficient in any situation where people are sensitive to the underlying distribution of data. Stochastic rules, however, naturally define a distribution over data. This output distribution also generalizes the "size principle", as that effect emerges as a consequence of the likelihood.

Another benefit of our representation is that we connect the pLOT literature with emerging work on probabilistic program induction (where statistical inference is used to learn a probability distribution over programs). Our model is closely related to program induction models such as that of Lake et al. (2015). We build on such work by demonstrating how to learn not only the parameters of probabilistic programs, but also their structure.[7] Whereas the programs induced by Lake et al. (2015) had a fixed structure, our model discovers the best structure from the space of possible ones as defined by the grammar. The model of Stuhlmüller et al. (2010) also featured hypotheses of varied structure, but in that work they eschewed a true inference procedure and simply compared people's results to the known, true hypothesis. As discussed in Section 4.2, our experimental results showed that people can learn varied and diverse hypotheses from the same data, and therefore modeling a true posterior over structures is crucial. Although structure learning is computationally challenging, it has the benefit of being flexible and domain general. For instance, we could extend our model to new domains simply by incorporating more or different primitives and providing new grammar rules.

For example, we could, in principle, extend our model to account for "category-based" rules. The rule learning literature distinguishes between "pattern-based" rules, which are defined by perceptual similarities, and "category-based" rules, which are defined by similarities between abstract, unobservable properties (Gomez & Gerken, 2000). The *ABA*-like rules studied in this work are instances of the former, since the same/different relationship between the parts in our objects can be discerned purely perceptually. Category-based rules, however, are based on abstract or unobservable properties. For example, such a rule might be based on grammatical categories, such as when "Children throw food" and "Beatrix drives cars" are linked as instances of *noun-verb-noun*. Here, even though "Children" and "Beatrix" are not perceptually similar, they fulfill the same role in the rule due to their abstract property of being nouns. Of course, category-based rules pose a much harder learning problem. Assuming a `same-as` predicate, as in the Frank and Tenenbaum (2011) model, for a category-based rule-learning task would be begging the question, as it would be hiding all of the interesting work. As many have noted (Murphy & Medin, 1985), similarity (or sameness) is no longer a satisfying explanatory tool in contexts where the relevant features are abstract or latent. In this case, the interesting theoretical question becomes how the learning system discovers which features to compare from the infinitely many possible. However, a generative approach based on program induction offers alternate possibilities.

For example, in our model we assumed a fixed, deterministic mapping from an object's string representation to its image. However, if we instead assume that a symbol in the string represents a category of shapes rather than a single shape, then the model could be applied to category-based rules. In this setup, we could provide primitives and grammar rules that enable a stochastic mapping from symbols to images, and it could then be a learned part of the inductive process (similar to the "token-level" subprogram in Lake et al. (2015)). Such a model could represent *ABA*-like rules where the *A*'s and *B*'s are whole classes of parts rather than perceptually identical ones. For example, the parts in our stimuli were all the same size and orientation (Fig. 1). But it is reasonable to assume that people could learn analogous concepts that are (at least partially) invariant to parts' sizes and orientations. A model with a learned, stochastic mapping from an abstract representation to concrete instances would be capable of doing so.

An alternative computational approach—deep neural networks—has recently become popular in the Machine Learning community and is likely to become increasingly popular for cognitive modeling. Although we appreciate the many strengths of deep neural networks, we also believe that our work reiterates an important challenge for advocates of this approach. As pointed out by others with respect to an earlier generation of neural networks (Fodor & Pylyshyn, 1988; Marcus, 2003), although traditional neural networks such as multilayer-perceptrons can perform some forms of abstraction, they do not present a natural way to account for variables. Consequently, they do not perform variable binding, and they do not show human-like patterns of generalization. In our work, this is precisely why the GCM-CNN model, which computes visual similarity based on features from a deep convolutional neural network, did not provide an adequate account of our experimental data. This remains a difficulty even for the new generation of neural networks such as deep convolutional networks. A critical challenge for advocates of deep neural networks is to modify them so that they can perform additional forms of abstraction, including variable binding. Promising approaches in this direction include models with some form of writeable memory (Graves, Wayne, & Danihelka, 2014; Graves et al., 2016; Gregor, Danihelka, Graves, & Wierstra, 2014; Reed & de Freitas, 2016).

Lastly, the HLOT model's account of our experimental results suggests a novel view of psychological simplicity. While it is true that simplicity—as measured by a PCFG-based prior distribution (Goodman et al., 2008; Piantadosi et al., 2016) or by counting operations (Feldman, 2000)—is an important driver of generalization in concept learning, full psychological simplicity may not be so simple. As Section 5 showed, a PCFG prior for which abstraction happened automatically fared better in predicting human judgments than a model in which abstraction was optional. This had the effect of making abstraction appear to be free, at least with respect to our stimuli. This provides quantitative, empirical evidence that abstractions in the form of variables are special as logical operations—because they happen automatically, they may make seemingly unintuitive contributions to conceptual complexity. This fact is only discoverable by using the tools that we employed: an experiment where subjects learn concepts that use variables, and models that can formalize both people's inferences and their use of representations that include variable abstractions.

Understanding these properties of cognition will be important for moving beyond simple program-induction models of concepts. Abstraction—through variables, subroutines, or libraries—is an important capacity in computational systems because it extends the reach of short, simple programs. For instance, if a function or a variable can be defined once and re-used, that permits a much shorter program than if the function or variable must be re-described each time it is used (Dechter, Malmaud, Adams, & Tenenbaum, 2013). This means that abstractions increase the effective power of any resource-bounded computational system, making them an important target for cognitive theories. Here, the variables that we have studied are very simple. This is both a

[7] The structure of the program itself is distinct from the structure of the objects generated by the program. While both models can output objects of varied structure (i.e., with different numbers of parts in different configurations), our model learns the structure of the program itself (i.e., which functions are called and how they are composed).

strength and a weakness of our approach. By examining the simple case of *ABA*-style rules, we are able to conduct a detailed investigation that reveals basic properties of how variables are handled. At the same time, the variables required for these are very simple; more complex cognitive phenomena will require richer types of abstractions and rules. In principle, our approach can be extended to other choices of cognitive primitives and abstraction mechanisms in order to study increasingly complex types of representational systems with variables. Thus, this work provides an early step for understanding how probabilistic inference, logical complexity, and abstraction interface in human conceptual representation.

Appendix A. HLOT model posterior inference

The HLOT model has four free parameters. Two parameters are production probabilities for grammatical rules (see Fig. 2): p_{single} is the production probability for choosing a specific object part as opposed to randomly sampling one, and p_{minus} is the production probability of removing a part from a set. In addition, the parameter p_{oi} is the probability that the extension of a hypothesis is orientation invariant (e.g., if $a \rightarrow d \rightarrow d$ is part of the extension, then $d \rightarrow d \rightarrow a$ is also part of the extension), and p_{rp} is the probability that a hypothesis considers the full set of object parts versus a restricted set limited to just the parts that occur in the training exemplars.

The posterior distribution over hypotheses was inferred in two stages. In the first stage, the four free parameters were fixed to default values, and a Markov chain Monte Carlo (MCMC) procedure (in particular, a Metropolis-Hastings algorithm) was used to search through the space of hypotheses. MCMC is a form of random walk that visits each state (or hypothesis, in our case) a number of times that is proportional to a target posterior distribution (in the limit as the number of iterations goes to infinity).

We initialized each random walk with the "widest" or most general hypothesis for each concept. In the *ABA*, *xBB*, and *ABC* conditions this was

```
let x1 = sample(Σ)
let x2 = sample(Σ)
let x3 = sample(Σ)
output x1 → x2 → x3
```

and in the *Ring* condition it was the equivalent program for five-part objects. At each iteration of the random walk, the algorithm modified the current hypothesis to generate a new sample. We used a mixture of two proposal functions: one was a variant of the tree-regeneration proposal function from Goodman et al. (2008) that only regenerates a subset of non-terminals (in order to preserve the structure of the string), and the other randomly flipped the values of θ_{oi} and θ_{rc}.

Because the hypothesis space is discrete, we can use MCMC to identify viable hypotheses (i.e., hypotheses with non-negligible probabilities). We constructed a set of viable hypotheses and their un-normalized posterior probabilities. In our simulations, we stored the top 2000 unique hypotheses, where uniqueness was defined by the hypotheses' extensions. If two hypotheses had the same extension, we stored only the simplest hypothesis (the one with the highest prior probability). We found that the set of viable hypotheses was large enough to capture all of the reasonable hypotheses as well as many others (i.e., those with low, albeit non-zero, probability). We ran each MCMC chain until the viable set remained unchanged (no new hypotheses were found) for 2000 iterations. Once we had obtained the viable set, we computed an accurate approximation to the full posterior distribution by normalizing the un-normalized posteriors by dividing each by the sum of all stored un-normalized posteriors.

$\vec{\rho}^*$: HLOT		$\vec{\rho}^*$: Optional abstraction	
p_{oi}	0.30	p_{oi}	0.50
p_{rp}	0.03	p_{rp}	0.02
$1 - p_{minus}$	$e^{-11.2}$	$1 - p_{minus}$	$e^{-1.4}$
p_{single}	$e^{-10.5}$	p_{single}	$e^{-1.0}$

Fig. 10. Fitted values for the HLOT model's free parameters as well as for the abstraction-optional variant. The parameters are the production probabilities p_{single} and p_{minus}, and parameters p_{oi} (*oi* stands for orientation invariance) and p_{rp} (*rp* stands for restricted parts).

During the second stage, we fit the values of the four free parameters based on our experimental data. This was accomplished using gradient descent to minimize (across all conditions) the sum of squared error between subjects' responses and model prediction as reported in Section 4.2:

$$\vec{\rho}^* = \arg\min_{\vec{\rho}} \sum_k \left(\widehat{P}(r_k) - P(r_k|x_1, \ldots, x_n)\right)^2$$

where $\vec{\rho}$ is a vector representing the four fitted parameters, and $\widehat{P}(r_k)$ is the observed proportion of subjects who responded 'yes' to test item k. Fig. 10 shows the four fitted values for each model. Lastly, we recomputed the posteriors of the viable hypotheses using these fitted values.

Appendix B. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cognition.2017.07.005.

References

Attneave, F. (1955). Symmetry, information, and memory for patterns. *The American Journal of Psychology, 68*(2), 209–222.

Chater, N., & Vitanyi, P. (2003). Simplicity: A unifying principle in cognitive science? *Trends in Cognitive Sciences, 7*(1), 19–22. http://dx.doi.org/10.1016/S1364-6613(02)00005-0.

Clopper, C. J., & Pearson, E. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika, 26*(4), 404–413.

Dechter, E., Malmaud, J., Adams, R. P., & Tenenbaum, J. B. (2013). Bootstrap learning via modular concept discovery. In *Proceedings of the 23rd international joint conference on artificial intelligence*.

Erdogan, G., Yildirim, I., & Jacobs, R. A. (2015). From sensory signals to modality-independent conceptual representations: A probabilistic language of thought approach. *PLoS Computational Biology, (11)*(11), 1–32 (pp. 1–32). http://dx.doi.org/10.1371/journal.pcbi.1004610.

Feldman, J. (1997). The structure of perceptual categories. *Journal of Mathematical Psychology, 41*(2), 145–170. http://dx.doi.org/10.1006/jmps.1997.1154.

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature, 407*(6804), 630–633. http://dx.doi.org/10.1038/35036586.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*(1–2), 3–71.

Frank, M. C. (2013). Throwing out the Bayesian baby with the optimal bathwater: Response to Endress (2013). *Cognition, 128*(3), 417–423. http://dx.doi.org/10.1016/j.cognition.2013.04.010.

Frank, M. C., & Tenenbaum, J. B. (2011). Three ideal observer models for rule learning in simple languages. *Cognition, 120*(3), 360–371. http://dx.doi.org/10.1016/j.cognition.2010.10.005.

Gayler, R. (2004). Vector symbolic architectures are a viable alternative for Jackendoff's challenges. http://dx.doi.org/10.1017/S0140525X06309028.

Geisler, W. S. (2003). Ideal observer analysis. *The Visual Neurosciences, 10*(7), 12. http://dx.doi.org/10.1167/10.7.12.

Gerken, L. (2006). Decisions, decisions: Infant language learning when multiple generalizations are possible. *Cognition, 98*(3), B67–74. http://dx.doi.org/10.1016/j.cognition.2005.03.003.

Gomez, R. L., & Gerken, L. (2000). Infant artificial language learning and language acquisition. *Trends in Cognitive Sciences, 4*(5), 178–186. http://dx.doi.org/10.1016/S1364-6613(00)01467-4.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive Science: A Multidisciplinary Journal, 32*(1), 108–154. http://dx.doi.org/10.1080/03640210701802071.

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines (pp. 1–26). arXiv:1410.5401.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-barwiflska, A., ... Hassabis, D. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature, 538*(7626), 471–476. http://dx.doi.org/10.1038/nature20101.

Gregor, K., Danihelka, I., Graves, A., & Wierstra, D. (2014). DRAW: A recurrent neural network for image generation (pp. 1–16). arXiv:1502.04623.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review, 104*(3), 427–466. http://dx.doi.org/10.1037/0033-295X.104.3.427.

Jackendoff, R. (2003). *Foundations of language: Brain, meaning, grammar, evolution*. Oxford: Oxford University Press.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the acm international conference on multimedia* (pp. 675–678). http://dx.doi.org/10.1145/2647868.2654889. Available from 1408.5093.

Kemp, C. (2012). Exploring the conceptual universe. *Psychological Review, 119*(4), 685–722. http://dx.doi.org/10.1037/a0029347.

Kemp, C., Bernstein, A., & Tenenbaum, J. B. (2005). A generative theory of similarity. In *Proceedings of the twenty-seventh annual meeting of the cognitive science society* (pp. 1132–1137).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1–9.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science, 350*(6266), 1332–1338. http://dx.doi.org/10.1126/science.aab3050.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady, 10*(8), 707–710.

Leyton, M. (1999). *Symmetry, causality, mind*. MIT Press.

Liu, Z., & Kersten, D. (2003). Three-dimensional symmetric shapes are discriminated more efficiently than asymmetric ones. *Journal of the Optical Society of America A, 20*(7), 1331–1340.

Marcus, G. F. (2003). *The algebraic mind: Integrating connectionism and cognitive science*. MIT Press.

Marcus, G. F., Vijayan, S., Bandi Rao, S., & Vishton, P. M. (1999). Rule learning by seven-month-old infants. *Science, 283*(5398), 77–80. http://dx.doi.org/10.1126/science.283.5398.77.

Murphy, G. L., & Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological Review, 92*(3), 289–316. http://dx.doi.org/10.1037/0033-295X.92.3.289.

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology. General, 115*(1), 39–61. http://dx.doi.org/10.1037/0096-3445.115.1.39.

Piantadosi, S. T., & Jacobs, R. A. (2016). Four problems solved by the probabilistic language of thought. *Current Directions in Psychological Science, 25*(1), 54–59. http://dx.doi.org/10.1177/0963721415609581.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2012). Bootstrapping in a language of thought: A formal model of numerical concept learning. *Cognition, 123*(2), 199–217. http://dx.doi.org/10.1016/j.cognition.2011.11.005.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2016). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, 1–43. http://dx.doi.org/10.1037/a0039980.

Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 512–519). http://dx.doi.org/10.1109/CVPRW.2014.131. Available from 1403.6382.

Reed, S., & de Freitas, N. (2016). Neural programmer-interpreters. In *International conference on learning representations (iclr)*. Available from arXiv:1511.06279.

Siskind, J. M. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition, 61*, 39–91. http://dx.doi.org/10.1016/S0010-0277(96)00728-7.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence, 46*(1–2), 159–216. http://dx.doi.org/10.1016/0004-3702(90)90007-M.

Smolensky, P., & Legendre, G. (2006). *The harmonic mind: From neural computation to optimality-theoretic grammar* (vol. 1: Cognitive architecture) Cambridge, MA: MIT Press.

Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. *Psychological Bulletin, 87*(2), 245–251. http://dx.doi.org/10.1037/0033-2909.87.2.245.

Stiny, G., & Gips, J. (1972). Shape grammars and the generative specification of painting and sculpture. *Information processing 71 proceedings of the IFIP congress 1971* (vol. 2, 71, pp. 1460–1465).

Stuhlmüller, A., Tenenbaum, J. B., & Goodman, N. D. (2010). Learning structured generative concepts. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*.

Tenenbaum, J. B. (1999). *A Bayesian framework for concept learning* (Doctoral dissertation). Massachusetts Institute of Technology.

Tenenbaum, J. B., & Griffiths, T. L. (2001). Generalization, similarity, and Bayesian inference. *Sciences-New York, 24*, 629–630. http://dx.doi.org/10.1017/S0140525X01000061.

Ullman, T., Goodman, N. D., & Tenenbaum, J. B. (2012). Theory learning as stochastic search in the language of thought. *Cognitive Development, 27*, 455–480. http://dx.doi.org/10.1016/j.cogdev.2012.07.005.

van der Velde, F., & de Kamps, M. (2006). Neural blackboard architectures of combinatorial structures in cognition. *The Behavioral and Brain Sciences, 29*(1), 37–70. http://dx.doi.org/10.1017/S0140525X06009022.

Yildirim, I., & Jacobs, R. a. (2015). Learning multisensory representations for auditory-visual transfer of sequence category knowledge: A probabilistic language of thought approach. *Psychonomic Bulletin & Review, 22*(3), 673–686. http://dx.doi.org/10.3758/s13423-014-0734-y.

Yuille, A. L., & Kersten, D. (2006). Vision as Bayesian inference: Analysis by synthesis? *Trends in Cognitive Sciences, 10*(7), 301–308. http://dx.doi.org/10.1016/j.tics.2006.05.002.