A model of conceptual bootstrapping in human cognition

Bonan Zhao, Christopher G Lucas, Neil R Bramley
University of Edinburgh

To tackle a hard problem, it is often wise to re-use, re-combine, or re-purpose existing knowledge. Such an ability to bootstrap enables us to grow rich mental concepts that go beyond our limited cognitive resources. However, the computational mechanisms underpinning this ability in humans are yet to be fully explicated. Here we present a model of conceptual bootstrapping that can cache and later reuse elements of earlier insights in principled ways. At its core, this model uses a dynamic conceptual repertoire that is enriched over time, modeling learning as a series of compositional generalizations. This model predicts systematically different learned concepts when the same evidence is processed in different orders, without any extra assumptions about prior beliefs or background knowledge. Across four behavioral experiments, we demonstrate strong curriculum-order and conceptual garden-pathing effects, revealing that people's inductive concept inferences closely resemble our model's, and differ from those of alternative accounts. This work provides an explanation for why information selection alone is not enough to teach complex concepts, and offers a computational account of how past experiences shape future conceptual discoveries.

Keywords: bootstrapping, concept learning, compositional generalization, Bayesian-symbolic models, adaptor grammars, order effects, garden-pathing

People have a remarkable ability to develop rich and complex concepts despite limited cognitive capacities. On the one hand, there is abundant evidence that people are bounded reasoners (Griffiths et al., 2015; Kahneman et al., 1982; Newell & Simon, 1972; Van Rooij, 2008; Vul et al., 2009), entertain a rather small set of mental options at a time (Bonawitz et al., 2014; Cowan, 2001; Sanborn & Chater, 2016; Sanborn et al., 2010; Vul et al., 2014), and generally deviate from exhaustive search over large hypothesis spaces (Acerbi et al., 2014; Bramley et al., 2017; Chater, 2018; Fränken et al., 2022; Gelpi et al., 2020). On the other hand, these bounded reasoners can develop richly structured conceptual systems (Gopnik & Meltzoff, 1997; Kemp & Tenenbaum, 2008; Quine & Ullian, 1978), produce sophisticated explanations (Craik, 1952; Keil, 2006; Lombrozo, 2012), and push forward complex scientific theories (Kuhn, 1970). How are people able to create and grasp such complex concepts that seem so far beyond their reach?

Newton gave a famous answer to this question: "If I have seen further, it is by standing on the shoulders of giants." (Isaac Newton, 1675). This reflects the intuition that people are bounded yet blessed with a capacity to not just learn from others, but to extend and re-purpose existing knowledge to create new and more powerful ideas. Such ability is taken to be a cornerstone of cognitive development (Carey, 2004). For instance, by building from atomic concepts of small numbers one, two, three, and counting, young children seem to *bootstrap* to more general and abstract numerical concepts such as successor relationships and the infinite line of real numbers (Piantadosi et al., 2012). Via bootstrapping, extant hard-earned knowledge need not be re-discovered every time it is used, saving the learner time and effort in constructing new concepts that build on old concepts. Because of such effective re-representation of existing knowledge, people can arrive at rich mental constructs incrementally (Gobet et al., 2001; Klein, 2017; Krueger & Dayan, 2009), and grow a hierarchy of concepts naturally through levels of nested reuse (Kemp & Tenenbaum, 2008).

While bootstrapping is a key idea in theories of learning and development (Carey, 2004), both behavioral studies that examine bootstrapping directly, and cognitive models articulating its mechanisms are relatively rare. Piantadosi et al. (2012) pioneered a line of research that posited bootstrapping in a Bayesian concept learning framework. However, they focused on the discovery of a recursive function in learning numeric concepts, and left open the task of examining bootstrapping as a general model of online inductive inference. Dechter et al. (2013) formalized the idea that an artificial learner can start with solving simple search problems, and then reuse some of the solutions to make progress in more complex problems. This approach later developed into Bayesian library learning, a class of models that aim to extract shared functionalities from a collection of programs (Bowers et al., 2023; Ellis et al., 2020). These models have successfully solved a variety of tasks, and have been shown to capture aspects of human cognition (Tian et al., 2020; Wong et al., 2022). However, these works primarily aim at learning optimal libraries or solving challenging test problems, rather than explicating how resource limitations interact with the mechanisms of bootstrapping, and how exploiting such interactions may explain human patterns of reasoning errors as well as successes.

We here provide a computational model of how people bootstrap, and propose an algorithmic mechanism that progressively produces rich concepts, even with limited cognitive resources. Treating how people construct concepts as a computational problem, we model bootstrapping as a process-level learning algorithm (Marr, 1982) that effectively caches previous learned concepts, and reuses them for more complex concepts through principled re-representation. To achieve this, we extend standard Bayesian concept learning frameworks with a dynamic concept library that can be enriched over time, powered by a formalization drawn from adaptor grammars (Johnson et al., 2007; Liang et al., 2010). We then design experiments informed by this model to test and measure how people construct complex concepts, and how this process adapts to the order in which people encounter, or think about, evidence. We compare this bootstrap learning account to a variety of alternative models of concept learning, and demonstrate how a cache-and-reuse mechanism provides an account for human inferential limitations, as well as how it enables us to reach concepts that are initially beyond our grasp in facilitatory conditions.

Formalization

Consider a causal learning and generalization task depicted in Figure 1A: An agent object A (called a "magic egg" in our experiments) moves toward a recipient object R (called a "stick"), and upon touching each other, the agent object A causes changes to the number of segments on the recipient object R, producing what we call the result object R'. Here, an agent object has two numerical features—a number of stripes and a number of spots–and people are asked to hypothesize about the nature of the causal relationship between the agent and recipient objects and the result, or formally, the content of function $f(\text{stripe}(A), \text{spot}(A), \text{segment}(R))$ that produces segment(R'). Without ambiguity, we shorten this to $R' \leftarrow f(\text{stripe}(A), \text{spot}(A), R)$.

Despite its apparent simplicity, this task captures a key challenge of concept learning: the space of possible hypotheses is infinite. For instance, it could be that object A adds two segments to the recipient R, i.e., $R' \leftarrow R + 2$; or perhaps A doubles the number of segments of R, i.e., $R' \leftarrow 2 \times R$; or each stripe on A is a multiplier, i.e., $R' \leftarrow \texttt{stripe}(A) \times R$. The space of possible causal hypotheses is unbounded. One can use a generative model to express this infinite space using a small set of building blocks (Goodman et al., 2008). In this case, consider a probabilistic context-free grammar $\mathbf{G}$ with primitives $\texttt{stripe}$(A), $\texttt{spot}$(A), R, small integers 0, 1, 2, 3, and operations +, −, ×. Primitives $\texttt{stripe}$(A), $\texttt{spot}$(A) and R return corresponding numeric values. Operations like + bind two numeric values and return a numeric value following the corresponding operation. Grammar $\mathbf{G}$ recursively samples these primitives to construct concepts (functions). Specifically, each operation primitive, such as +, can either bind numeric primitives, or invoke another combination of operations, forming nested functions such as $\texttt{stripe}(\mathrm{A}) \times (\mathrm{R} - 1)$. Grammar $\mathbf{G}$ thus covers an infinite space of possible concepts, and can be used to assign a probability distribution over this space (Methods). For a concept z, its prior probability is given by $P_{\mathbf{G}}(z)$. As learners gather data D, they can check how likely it is for concept z to produce data D, known as the likelihood $P(D|z)$. According to Bayes' rule, learners are then informed by the posterior $P(z|D) \propto P(D|z) \cdot P_{\mathbf{G}}(z)$. While directly computing this posterior is infeasible because the normalization term involves infinity, many methods exist to approximate this calculation (Fränken et al., 2022; Goodman et al., 2008; Piantadosi et al., 2016; Thaker et al., 2017).

We build on this Bayesian symbolic concept learning framework to model conceptual bootstrapping. Specifically, we use adaptor grammars (AG, Johnson et al., 2007) as our generative grammar to assign prior probabilities. An adaptor grammar, by design, learns probabilistic mappings among sub-parts of a structure, capturing the intuition that when some concepts go together frequently, it makes sense to expect that the entire ensemble will be common in the future. Such a mechanism of caching concept ensembles and reusing them as a whole relaxes the context-free assumption of the context-free grammar $\mathbf{G}$ introduced above, and captures the essence of bootstrap learning—to effectively reuse learned concepts without having to re-discover them every time it is used. Liang et al. (2010) extends adaptor grammars with combinatory logic, offering an algorithm for learning programs that benefits from learning sub-program sharing and reuse. Here, we adapt the algorithm in Liang et al. (2010) to examine this cache-and-use mechanism as a process-level model of conceptual bootstrapping under resource constraints. Specifically, instead of sampling from a fixed set of primitives, we introduce a latent concept library that can be updated dynamically. Concept library L contains primitive concepts, as well as cached concept ensembles, weighted by how useful an ensemble has been (see next). Learners generate concepts using contents in library L, and adaptor grammar $\mathbf{AG}$ defines the probability for a library L to generate a concept z (Methods). This joint probability $p(z, L)$ provides a prior $P_{\mathbf{AG}}(z|L)$. We can then combine likelihood $p(D|z)$ with this prior, yielding the posterior $p(z|D, L)$.

The goal of inference is thus to infer the latent library L that can best account for learning data D. Following previous work suggesting that human learners make inferences by sampling from an approximate posterior instead of tracking the entire posterior space of possibilities (Bramley et al., 2017), we use known methods for sampling from Pitman-Yor processes (Pitman & Yor, 1997), such that conditional

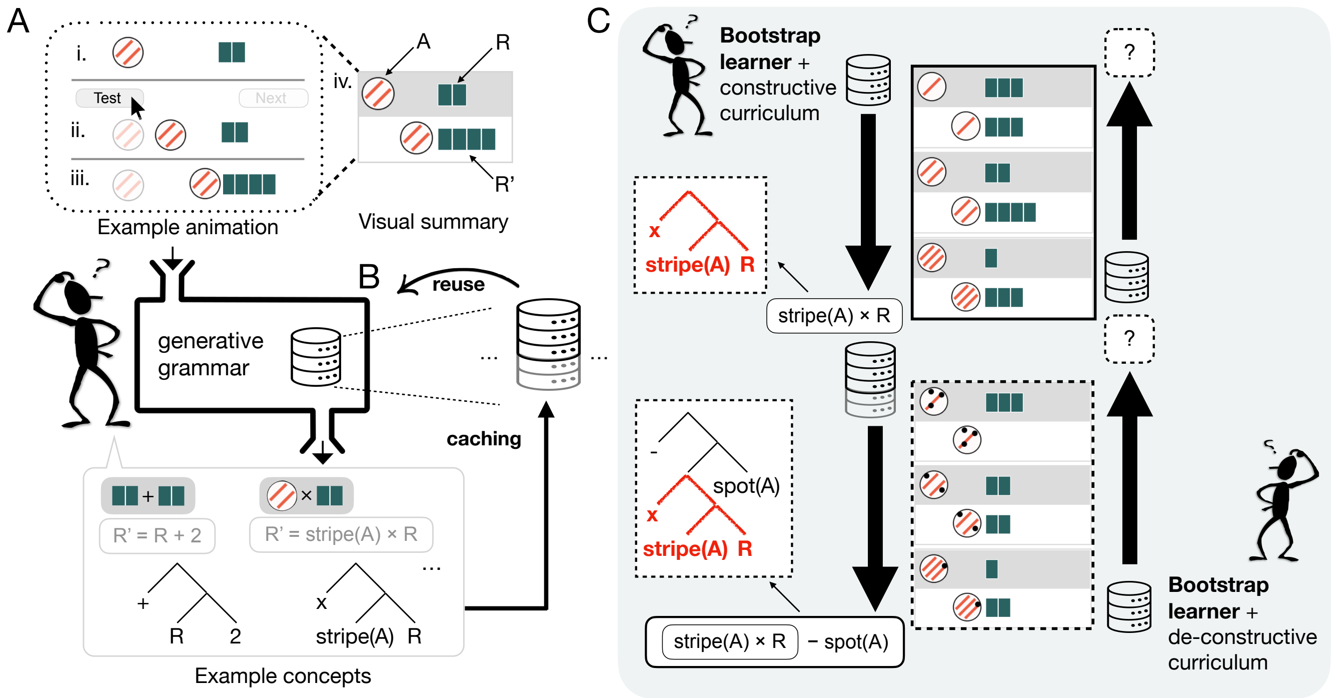

Figure 1. A. Example causal interaction with i. a causal agent (left, circle) and a recipient object (right), ii. agent moves rightward to the recipient, and iii. upon touching the recipient object, this changes into its result form. Translucent marker is only used here to illustrate the animation. iv., summary of this animation, with grey background showing the agent 'A' and recipient 'R' before the causal interaction, and white background the agent and result after the causal interaction 'R''. B. Schematic of bootstrap learning model. C. Example bootstrap learning trajectories over six observations, see main text for explanation.

on a library L at any given moment, learners can make appropriate inferences about the probabilities of different explanations for new or salient events. In particular, we use Gibbs sampling (Methods), a Markov Chain Monte Carlo (MCMC) method, over the joint distribution of concepts and libraries. At each iteration of Gibbs sampling, we sample concepts $z \sim P_{AG}(z|L)$, and combine them with the likelihood function to find out concepts favored by data. Then we sample up to 3 favored concepts and add them, as well as their sub-parts, to library L (caching, Figure 1B), producing a library sample L'. Note that in the next iteration, when sampling from $P_{AG}(z|L')$, those added contents are used as if they were primitives (reuse, Figure 1B), and therefore the learner can compose sophisticated combinations with rather few steps of composition (Methods).

This idea of a dynamic concept library is especially powerful when we take resource constraints into account. Take the six observations in Figure 1C for example, the ground truth concept involves different causal powers (math operations) per agent feature. Therefore, trying to find a concept consistent with all the six observations is a challenging problem. However, if one looks at the first three pairs that only involve stripes (bordered box, Figure 1C), the learner might discover that stripes may multiply segments, i.e., `R'` ← `stripe(A)` × R. With this idea in mind, now looking at all six pairs, the learner may now manage to construct a nested concept `R'` ← (`stripe(A)` × R) − `spot(A)` that explains all the observations by reusing the earlier concept as a sub-concept. If we swap the presentation order and first show the learner the last three pairs in Figure 1C (dashed-border box), the space of possible concept may overwhelm the learner, and without having cached any useful sub-concepts, the full observation set might be just as confusing. Under our bootstrap learning model, individual learners could develop a concept library L^* that is the result of two sequential episodes of posterior search and caching. Provided that the first search phase leads to the learner caching the crucial building block `stripe`$(A) \times R$, the second search phase is liable to result in their discovering and caching the ground truth, making this concept directly available when the learners go to make generalizations and explicit guesses.

Results

Our bootstrap learning model predicts that successful search for a complex target concept is heavily reliant on having good, previously-learned abstractions. We test these model predictions using a two-phase causal learning and gen-

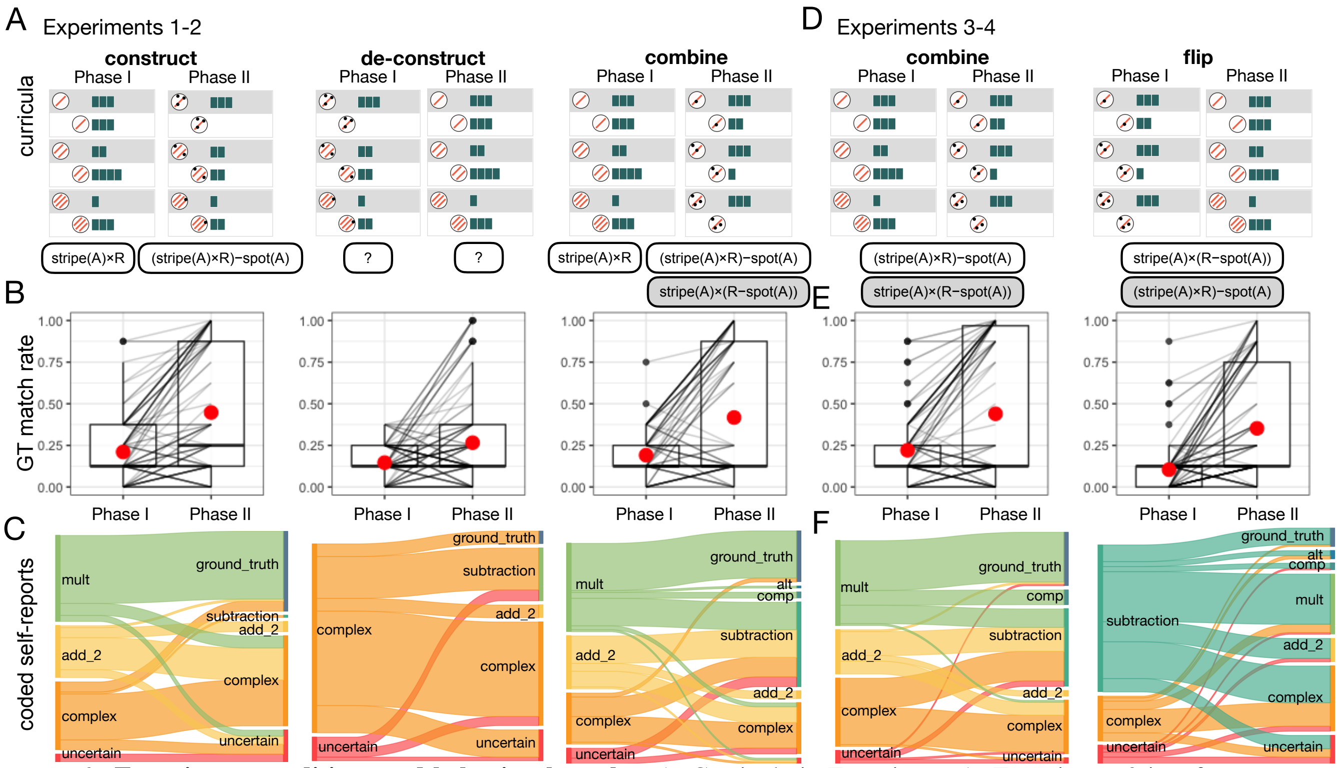

Figure 2. **Experiment conditions and behavioral results.** A. Curricula in Experiment 1. Experiment 2 is a feature counterbalance of this, available in SI. Texts below each phase are data-compatible causal concepts. Transparent text boxes are concepts favored by the model, and shaded boxes for equally complex and data-consistent alternative concepts. B. Participants generalization accuracy (match to ground truth) in Experiments 1 and 2. Box plots show the first and third quantiles with lines for the medians; red dots mark the means. C. Coded self-reports in Experiments 1 and 2. D. Curricula design in Experiment 3. Experiment 4 is a feature counterbalance of this and is available in SI. E. Participants' match to ground truth in Experiments 3 and 4. F. Coded self-reports in Experiments 3 and 4.

eralization task. In Phase I, learners observe three pairs of objects and their causal interactions (in fixed orders as illustrated in Figure 2A), write down their guessed causal function, and make generalization predictions on eight pairs of novel objects appearing in random orders. Right after, in Phase II, learners observe three more pairs of objects and their causal interactions (with the previous three pairs still visible above), provide an updated guess to account for all six pairs, and then make generalization predictions again on the same eight pairs as earlier, in new randomized orders (Methods)

Experiments 1 & 2: Curriculum-order effects

Experiment 1 ($N = 165$) examined three curricula. Curriculum *construct* and *de-construct* were as described in Figure 1C and discussed above. We further included a *combine* curriculum that shares the same Phase I as in *construct*, but in Phase II keeps $\texttt{stripe}(A) = 1$ throughout (Figure 2A), making it ambiguous how $\texttt{stripe}(A) \times R$ and $R - \texttt{spot}(A)$ should be combined. If people process Phase II with the cached sub-concept from Phase I, we expect to see $R' \leftarrow \texttt{stripe}(A) \times R - \texttt{spot}(A)$ more often than $R' \leftarrow \texttt{stripe}(A) \times (R - \texttt{spot}(A))$. In a follow-up Experiment 2 ($N = 165$), we flipped the roles of the stripes and spots of the agent object (Methods and SI). While all main results replicate robustly in Experiment 2, we report per-curriculum collapsed results in analysis here for simplicity.

First, we observed a significant difference in Phase II generalization accuracy[1]—defined as "match to ground truth"—between the *construct* and *de-construct* curricula. As illustrated in Fig. 2B, participants under the *construct* curriculum achieved an accuracy of $44.7\% \pm 38.3\%$, significantly higher than those with the *de-construct* curriculum of only $22.6\% \pm 27.5\%$, $t(1717) = 8.13, p < .001$, Cohen's $d = 0.4$, 95%CI = $[0.14, 0.24]$ (chance accuracy: $1/17 = 5.88\%$). The large standard deviations here imply a wide-spread in-

[1]Strictly-speaking, there are no wrong answers for the generalization tasks, because they are all novel out-of-distribution pairs, such that any generalization prediction is justifiable under some inferred concept.

dividual difference in causal generalizations, showcasing the openness and creativity of how people conceptualize causal relationships. Such individual difference crystallizes when looking at participants' self-reports (Fig. 2C). For Phase II self-reported guesses, 37.8% of participants under the *construct* curriculum were classified as describing the ground truth (Fig. 2C), and in *de-construct* condition only 6% did so, $t(151) = 6.05, p < .001$, Cohen's $d = 0.8$, 95%CI = [0.21, 0.42]. A deeper dive into those self-reports revealed that, for those who induced that one feature multiplies in Phase I, 79% subsequently landed on ground truth in Phase II, showing a clear bootstrap learning trajectory. Recall that at the end of Phase II in both *construct* and *de-construct* curricula, participants have seen identical learning information (Fig. 2A), hence this substantial difference in final learning performance coheres with our main claim that people reuse sub-concepts to compose more complex ones. Merely observing evidence that favors a target concept is not sufficient to induce this concept.

The low matches with the ground truth in self-reports in the *de-construct* curriculum also reflects a strong garden-pathing effect (Bever, 1970). We coded participants' self-reports according to whether the content matches the ground truth, describes an operation such as multiplication, subtraction, or addition, is uncertain, or involves complex reasoning patterns drawing upon conditionals, positions of spots or relative quantities (Methods). Notably, 89% of participants in the *de-construct* condition came up with guesses classified as "complex" in Phase I. For example, one participant wrote: "*If there are more stripes than dots the stick is reduced in length. If there are equal stripes and dots then the stick stays the same. If there are more dots than stripes the stick increases in length.*" This is a significantly higher proportion than the complex rule reported in the *construct* Phase I (31.7%), $t(183.56) = -10.61, p < .001$, Cohen's $d = 1.4$, 95%CI = [−0.68, −0.46]. The average length of Phase I guesses for the *de-construct* curriculum was 168 ± 145 characters, significantly longer than answers in the *construct* curriculum's 112 ± 68.1 characters, $t(168.09) = -3.76, p < .001$, Cohen's $d = 0.5$, 95%CI = [−85.65, −26.72]. These longer and more complex initial guesses appeared to influence the second phase of the experiment. In *de-construct* Phase II, after seeing the simpler examples, 50% of the complex-concept reporters either stuck with their initial complex guesses or embellished them even more, resulting in 48.7% complicated self-reported causal concepts in Phase II. Furthermore, only 24.8% of participants in Phase II of the *de-construct* curriculum described that one feature multiplies, significantly lower than the 40.2% of *construct* curriculum participants after Phase I , $t(212.13) = 2.47, p = .01$, Cohen's $d = 0.3$, 95%CI = [0.03, 0.28]. These results show that people frequently fall prey to learning traps in which initial complex examples prohibit them from arriving at the ground truth (Gelpi et al., 2020; Rich & Gureckis, 2018). Again, this pattern is consistent with the hypothesis that participants reuse their own phase I ideas in order to bootstrap learning in phase II.

Finally, participants in the *combine* condition overwhelmingly favored ground truth over the alternative, despite them being equally complex and compatible with the data. In Phase II self-reports, 24.5% of participants in the *combine* condition reported the ground truth, with only one reported the alternative concept (0.94%) (Fig. 2C). The Phase II generalization accuracy of the *combine* curriculum ($41.7\% \pm 38.5\%$) did not differ significantly from that in the *construct* curriculum ($44.7\% \pm 38.3\%$), $t(1702) = 1.25, p = .2$. This aligns with our predictions that people reuse Phase I learned concept as a primitive in Phase II, and more strongly it shows that such tendency leads people to systematically favor certain concepts over alternatives of the same level of accuracy and complexity.

Experiments 3 & 4: Biases in compositional form

Results of the *combine* curriculum seem to support the idea that people reuse previous construction as conceptual primitives. However, it could also be compatible with the idea that people just "glued" the two sub-concepts together additively. That is, $(\texttt{stripe}(A) \times R) + (-\texttt{spot}(A))$ is logically equivalent to the ground truth. Furthermore, this "multiply-first" function fits more naturally with conventional order of mathematical operations, in which multiplication is performed before addition in absence of brackets. To disentangle these concerns, we further designed a new curriculum, *flip*, which swaps Phase I and Phase II of *combine* (Fig. 2D). In this *flip* curriculum, if people reuse the concept they inferred in Phase I as a conceptual primitive in Phase II, they should conclude $R' \leftarrow \texttt{stripe}(A) \times (R - \texttt{spot}(A))$, the data-consistent alternative not favored by the *combine* condition. If people rather use addition as their default or dominant compositional mode, then in *flip* Phase II we would expect that they will still favor the original ground truth. Experiment 3 ($N = 120$) tested this *flip* curriculum, together with the *combine* curriculum as in Experiment 1, using material exactly as shown in Fig. 2D. Experiment 4 ($N = 120$) reversed the causal powers between the stripe and spot features and otherwise replicated Experiment 3 (Methods and SI).

We found that people indeed favored the ground truth less often in the *flip* curriculum (Fig. 2E-F). For generalization accuracy, here defined as match to the original ground truth, participants in *flip* Phase II was at $35.2\% \pm 34.3\%$, while participants in *combine* achieved $44\% \pm 41.8\%$, $t(1881.9) = 3.93, p < .001$, Cohen's $d = 0.2$, 95%CI = [0.04, 0.13]. In addition, only 8.7% of participants in the *flip* curriculum reported ground truth in Phase II, compared to 25.4% in the *combine* condition, $t(190.31) = 3.47, p < .001$, Cohen's $d = 0.5$, 95%CI = [0.07, 0.26]. These results are in line with our

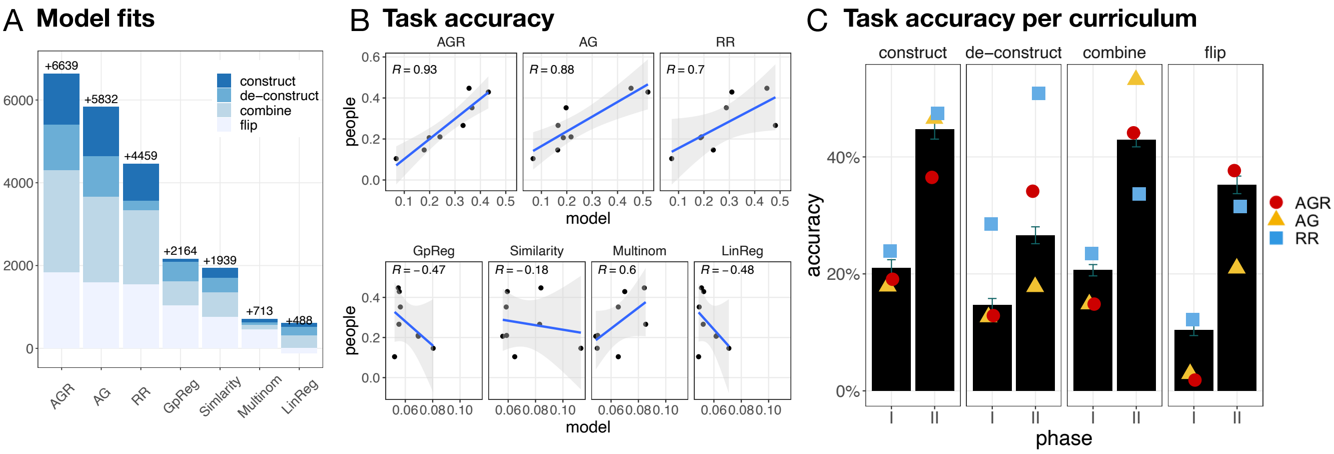

Figure 3. **Modeling results**. A. Model fit (total log likelihood) improvement over random baseline (y=0), log scale. B. Generalization accuracy per curriculum and phase. X-axis are model predictions, y-axis people's. C. Generalization accuracy between people (black bars) and four symbolic models.

previous finding that constructing, caching and later reusing the key sub-concept is crucial for acquiring the complex target concept.

However, a further examination suggests that the drop in synthesizing ground truth in *flip* was not primarily driven by turning to the alternative. Participants' generalization accuracy in terms of matching the alternative concept was 28.8% ± 17.3%, lower than the level of agreement with the predictions of the original ground truth. As illustrated in Fig. 2F, five participant in *flip* Phase II reported the alternative concept (2.08%), in comparison with 16.7% guessing the ground truth, $\chi^2(1) = 27.2, p < .001$, Cramer's $V = 0.8$. This suggests that additive compositional form is still quite a prevalent inductive bias, and it interacts with sequential bootstrap learning in phased reasoning tasks. Putting it another way, people may be choosing which phase to chunk according to their inductive bias on compositional form, and this might override the order that evidence was actually presented in the experiments.

In our experimental interface, at the end of Phase II, all six pairs of learning examples were available on the screen, and participants could freely scroll up and down to revisit any earlier pairs. Such revisiting could induce orders of cache-and-reuse that are different from the ones designed by the experimenters. In fact, since we encouraged participants to synthesize causal relationships that can explain all six pairs, this may consequently encourage deliberate revisits. By revisiting evidence, in the *flip* curriculum, a strong inductive bias on additive compositional form could lead to preferring ground-truth over the alternative. In the *de-construct* curricula in Experiments 1 and 2, some participants may have revisited Phase I after observing Phase II, and therefore discovered the ground truth accordingly, reflected by the slight increase in Phase II generalization accuracy compared to Phase I in *de-construct* (Fig. 2B).

Model comparison

We now examine predictions and simulations from a range of computational models comparing their ability to reproduce participants' generalization patterns. First, we considered a bootstrap learning model based on adaptor grammars AG as described in the Formalization section. Model AG first processes Phase I learning examples, acquiring an updated library, and then processes Phase I and II altogether with the updated library. Next, to account for the fact that participants were able to scroll up and down and re-access Phase I after reasoning about Phase II, we considered a variant of AG, Adaptor Grammar with Re-processing (AGR). This model mixes predictions $\hat{y}_{\rightarrow}$ from Phase I to II, and predictions $\hat{y}_{\leftarrow}$from Phase II to I, with a weight parameter $\theta \in [0,1]$, getting a mixed prediction $\hat{y}_r \propto \theta \cdot \hat{y}_{\rightarrow} + (1-\theta) \cdot \hat{y}_{\leftarrow}$. For hyper-parameters in models AG and AGR, we used the same values as in Liang et al. (2010). From the estimated posterior libraries we can collect a large number of generated concepts. Since concepts here are functions specifying R' for any agent-recipient object pairs, evaluating these concepts on novel object pairs and marginalizing on the predictions gives a distribution of R' for any novel object pair (Methods).

For comparison, we also examined a "rational rules" model (RR) based on Goodman et al. (2008). This assumed the same conceptual primitives as the adaptor grammar models, but uses a probabilistic context-free grammar to get prior concepts, as specified by grammar **G** in the Formalization section (see also Methods). Since we evaluate models using generalizations, we also implemented several sub-symbolic models capable of generalization but not explicit rule guesses. Here we included a similarity-based categorization model (Tversky, 1977), a linear regression

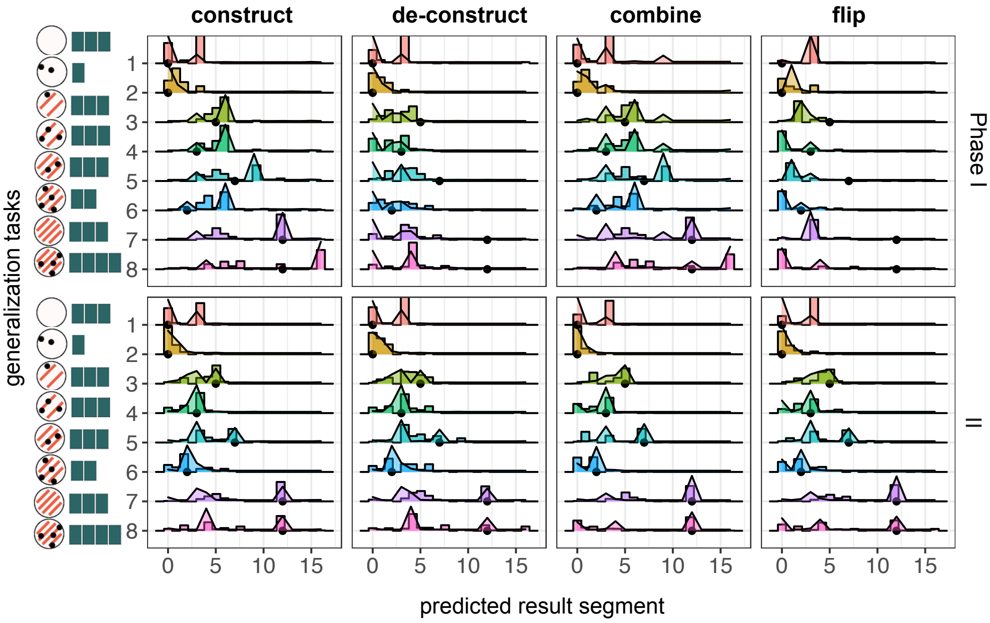

Figure 4. Generalization predictions by people (bars) and the best fitting AGR model (area). Rows of panels are for experimental phases, columns for conditions. In each panel, x-axis are predicted number of segments (0-16), y-axis are tasks.

model (LinReg), and a multinomial regression model (Multinom). We further considered a Gaussian process regression (GpReg) model with radial basis function kernels (one per feature), since these models exhibit human-like performance in function learning and few-shot generalizations (Lucas et al., 2015; Wu et al., 2018). For these categorization and regression models, parameters were fitted to the learning examples predicting R' using $\texttt{stripe}(A)$, $\texttt{spot}(A)$, and R. We then made predictions about the novel objects with those fitted models, and evaluated model predictions in terms of their log-likelihood LL of producing participants' predictions (Methods).

Figure 3A shows each model's improvement over baseline, $\Delta_{\text{model}} = LL_{\text{model}} - LL_{\text{rand}}$. Model AGR achieves the greatest improvement, with the three Bayesian-symbolic models (AGR, AG, RR) easily outperforming similarity-based or regression models. With fitted model parameters, Fig. 3B plots generalization accuracy in each phase for each curriculum between model and people. In line with overall model fits, AGR best predicts people's performance across all cases, and the non-symbolic models fail to match people's predictions.

Notably, while model RR can learn that some primitives are more common or useful than others, it is unable to discover and re-use concepts, as illustrated in Fig. 3A. We further plot generalization accuracies for models AGR, AG, and RR against behavioral data in Fig. 3C, showing that RR fails to reproduce the curriculum-order effects between the *construct* and *de-construct* curricula. This is because model RR is likely to have figured out ground truth after seeing all the data, even for the *de-construct* curriculum, and thus deviating from how people process phases of information. Model AG, on the other hand, is defeated by the learning trap as many people were, exhibiting no accuracy improvement in Phase II relative to Phase I. Model AGR mixes model AG with some re-processing, and is therefore able to capture participants' modest improvement in *de-construct* Phase II generalizations. Furthermore, RR achieves lower accuracy than people in the *combine* Phase II, because it assigns as much posterior probability to the intended ground truth as to the equivalent-consistent alternatives.

Figure 4 shows the best fitting AGR model's predictions in each generalization task with participant data showing a close match. We note one interesting discrepancy in generalization task 1, which asked about an agent with no spots or stripes: While many participants predicted the disappearance of segments, since $R' \leftarrow \texttt{stripe}(A) \times R$ and $0 \times 3 = 0$, many participants also predicted that the resulting number of segments would stay the same. This could be due to participants concluding that absent features meant that nothing would happen. Future work could investigate how people reason about these kinds of edge cases.

Overall, the adaptor grammar models AG and AGR provided a much better account of people's behavioral patterns in the experiments than the other models we considered. More generally, this means that curriculum-order effects and garden-pathing effects exhibited by people, can be explained as consequences of a cache-and-reuse mechanism expanding the reach of a bounded learning system. Critically, these phenomena cannot be explained by a standard Bayesian-symbolic model out of the box, or by familiar sub-symbolic categorization models, showcasing that a cache-and-reuse mechanism is central to human-like inductive inference to compositional concepts.

Discussion

We proposed a formalization of bootstrap learning that supercharges Bayesian-symbolic concept learning frameworks with an effective cache-and-reuse mechanism. This model replaces a fixed set of conceptual primitives with a dynamic concept library enabled by adaptor grammars, facilitating incremental discovery of complex concepts under helpful curricula in spite of finite computational resources. We showed how compositional concepts evolve as cognitively-bounded learners bootstrap from earlier conclusions over batches of data, and how this process gives rise to systematically different interpretations of the same evidence depending on the order it is processed. Being a Bayesian-symbolic model, our approach accounts for both the causal concepts people synthesized, and the generalization predictions they made.

People often exhibit a general path-dependence in their progression of ideas (Mahoney & Schensul, 2006). We showed that this follows naturally when a bootstrap learner progresses in a space of compositional concepts, constructing complex ideas "piece by piece" with limited cognitive resources. Crucially, we focused on how reusing earlier concepts bootstraps the discovery of more complex compositional concepts using sampling-based inference. This builds on other sampling-based approximations to rational models (e.g. Sanborn et al., 2010), that demonstrate how memory and computational constraints create focal hypotheses in the early stages of learning, and impair a learner's ability to accommodate data they encounter later (Gelpi et al., 2020; Thaker et al., 2017). Going beyond this earlier work, we showed how people exceed their immediate inferential limitations via reusing and composing earlier discoveries through an evolving library of concepts. Our proposal also relates to Gershman and Goodman (2014)'s observation that amortized inference can explain how solving a sub-query improves performance in solving complex nested queries. While our model instantiates reuse in a compositional space by caching conceptual building blocks in a latent concept library, there is potential to explore the connection between our formalization with amortized inference in terms of how reusing partial computation may shape the approximation of the full posterior.

We also offered additional process-level explanations of why and how people often come up with diverse understandings of the same evidence. People are known to develop biased interpretations of features (Searcy & Shafto, 2016), and fall easily for various learning traps in category-based generalization related to selective attention or assumptions about stochasticity and similarity (Rich & Gureckis, 2018). Jern et al. (2014) argued that different evaluations of the same evidence is due to the different prior beliefs people hold. Tian et al. (2020) corroborated that equipped with different concept libraries, people can derive different solutions to the same problem set. Our formalization, however, demonstrates that drastically different conceptualization of the same evidence can arise among learners with the same learning mechanisms and even the same priors, systematically deviating from a normative approach to library learning. Note that our experiments tested causal learning and generalization in abstract settings, rather than over subjective opinions such as political attitudes, and therefore serves a friendly reminder that an objective interpretation is not guaranteed to prevail, even among capable cognizers scrutinizing the same data.

This interaction between our evolving concepts and our trajectory through environment they seek to reflect lends itself to several interesting future directions. Culbertson and Schuler (2019) reviewed children's performance in artificial language learning and stressed that learning is tightly bounded by cognitive constraints. We further found that inductive biases, like those about compositional forms we identified in Experiments 3 and 4, shape the order in which people process information. That is, rather than passive information receivers, it seems far more plausible that people have inductive biases of attention and action that shape how they select which subset of a complex situation to process first, and then build on to make sense of the whole picture. Future work may extend our framework to active learning scenarios to study such information-seeking behaviors and self-directed curriculum-design patterns in the domain of concept learning (e.g. Bramley & Xu, 2023). Moreover, cache-and-reuse is a useful way to refactor representations. Liang et al. (2010) introduced a sub-tree refactoring method for discovering shared sub-structures, providing natural future extensions for studying refactoring as a cognitive inference algorithm involved in the development of concepts (Rule et al., 2020).

Recent research in neuroscience is starting to unravel how the brain may perform non-parametric Bayesian computations and latent causal inference (Tomov et al., 2018), and has uncovered representational similarities between artificial neural networks and brain activity (e.g. Flesch et al., 2022; Sorscher et al., 2022). Along these lines, neural evidence for reuse of computational pathways across tasks (cf. Dasgupta & Gershman, 2021) would seem to support our thesis, and further enrich our understanding of how brain grows its conceptual systems and world models. One challenge for the symbolic framing we adopted here comes from the fact that our conceptual representations are intimately tied in with their embodied sensorimotor features and consequences (Fernandino et al., 2022). We look forward to more integrated models that capture how symbolic operations of composition and caching interface with such deeply embodied representations.

Our current work has several limitations that future work could address. For instance, we assumed a deterministic likelihood function, but this does not handle vague concepts like *the stick decreases* or *increases* very well. A grammar and likelihood able to capture concepts that constrain rather than uniquely predicting generalizations could capture a larger range of people's guesses and predictions. Since, for simplicity, we did not include conceptual primitives for conditionals, our model could not express all of the "divide-and-conquer" self-reports people made when attempting to make sense of overwhelmingly complex information. This would be a straightforward extension, achievable by starting with more basic primitives, or assuming an `ifElse` base concept. Piantadosi (2021) argued that base primitives in combinatory logic suffice to ground any Turing-machine computable mental representation and computation. We used natural-language-like base terms simply for computational and expressive convenience, and all of the base primitives and learned concepts we assumed can be decomposed into solely combinatory logic bases. In addition, there exist many other options than combinatory logic to formalize our tasks. If we view variable objects A and R as hard-coded primitives, for example, a first order logic formalization could have sufficed. We however preferred combinatory logic for its convenience and flexibility in routing variables, as this makes it easier to share and reuse any generated program. One other limitation of our current model is that it does not handle forgetting by default, a critical feature of human memory and learning (Della Sala, 2010; Gravitz, 2019; Nørby, 2015). To extend our formalization to model life-long learning, it would be important to incorporate a mechanism through which concepts are forgotten, either through decay or being overwritten or out-competed (Brown et al., 2007).

In sum, we argued for the central role of bootstrap learning in human inductive inference, and proposed a process-level computational account of conceptual bootstrapping. Our work puts forward cache-and-reuse as a key cognitive inference algorithm, and elucidates the importance of active information parsing for bounded reasoners grappling with a complex environment. Our findings stress the importance of curriculum-design in teaching, and to facilitate communication of scientific theories. We hope this work will inspire not only social and cognitive sciences, but also development of more data-efficient and human-like artificial learning algorithms.

Methods

Experiment 1

Participants. 165 participants ($M_{\text{age}} = 31.8 \pm 9.9$) were recruited from Prolific Academic, according to a power analysis for three between-subject conditions seeking at least 0.95 power to detect a medium size (≈ 0.35) fixed effect. Participants received a base payment of £1.25 and performance-based bonuses (highest paid £1.93). The task took 9.69 ± 4.47 minutes. No participant was excluded from analysis. All experiments were pre-registered and performed with ethical approval from the University of Edinburgh.

Stimuli. The agent object A was visualized as a circle that moved in from the left of screen and collided with the recipient R (Fig. 1A). The agent object A varied in its number of stripes and randomly positioned spots. The recipient object R took the form of a stick made up of a number of cube-shaped segments. During learning, all feature values were between 0 and 3. The rule we used to determine the recipient's final number of segments was $R' \leftarrow \texttt{stripe}(A) \times R - \texttt{spot}(A)$. Learning materials were shown as in Fig. 2A. For generalization tasks, an arbitrary segment number (0 to 16) could be selected putting a nominal eyes-closed floor level of performance at $1/17 = 5.88\%$. Generalization trials were selected via a greedy entropy minimizing search in order to select a set that well distinguishes between a set of hypotheses favored by model AG (see SI). Live demos are available at https://bramleylab.ppls.ed.ac.uk/experiments/bootstrapping/p/welcome.html, and pre-registration at https://osf.io/ud7jc.

Procedure. Each participant was randomly assigned to one of the three learning conditions, *construct*, *de-construct*, and *combine*. After reading instructions and passing a comprehension quiz, participants went through experiment Phase I and then Phase II. In each phase, a participant tested three learning examples in the corresponding phase as shown in Fig. 2A, each appearing sequentially and as ordered in Fig. 2A. Participants watched the animated causal interactions by clicking a "Test" button. Once tested, a visual summary of the learning example including the initial and final state of the recipient was added to the screen and remained visible until the end of the experiment. After the learning stage, participants were asked to write down their guesses about the underlying causal relationships, and make generalization predictions for eight pairs of novel objects. Generalization trials appeared sequentially. Once a prediction was made the trial was replaced by the next one. The pairs of generalization objects in both Phase I and Phase II are the same, but their presentation orders were randomized for each participant and in each phase.

Experiments 2-4

Experiment 2 is a feature counterbalanced replication of Experiment 1, using true rule $R' \leftarrow \texttt{spot}(A) \times R - \texttt{stripe}(A)$. Another 165 participants ($M_{\text{age}} = 33.8 \pm 10.1$) who did not participate in Experiment 1 were recruited from Prolific Academic. The task took 9.8 ± 5.2 minutes. No participant was excluded from analysis. Payment scale (highest paid £1.95) and procedure are identical to Experiment 1. Stimuli and pre-registration are available at https://osf.io/k5dc3 and in SI. We conducted a two-way ANOVA to analyze the effect of feature-counterbalancing and curriculum-design on Phase II generalization accuracy. While both factors had significant main effects (curriculum-design: $F(2, 2) = 9.2, p < .001$, feature-counterbalancing: $F(1, 2) = 8.5, p < .001$), there is no significant interaction, $F(2, 324) = 0.15, p = .9$. This indicates that people may be treating stripe and spot features differently, but this difference does not drastically interfere with our results about curriculum designs.

Experiment 3 recruited another 120 participants ($M_{\text{age}} = 35.4 \pm 10.9$) to test the *combine* and *flip* curricula in Fig. 2D. We initially recruited $165 \div 3 \times 2 = 110$ participants to match group sizes in Experiments 1 and 2, but was faced with an imbalance between the two curricula (*combine*: 47, *flip*: 63) due to the random number generator the experiment used to assign participants. To even out the samples, we recruited another 10 participants on Prolific on the same day, all to the *combine* curriculum, and ensured that these extra batch did not contain participants from Experiments 1, 2 and current Experiment 3. All 120 participants were paid at the same scale as in Experiments 1 and 2 (highest paid £1.85). The task took 10.7 ± 4.5 minutes. The procedure was otherwise identical to Experiments 1 and 2. No participant was excluded from analysis. Pre-registration for this experiment is available at https://osf.io/mfxa6, and full stimuli is available in SI.

Experiment 4 was a feature counterbalanced replication of Experiment 3. We recruited another 120 participants ($M_{\text{age}} = 34.0 \pm 12.6$) on Prolific, who did not participate in Experiments 1-3. Here the roles of the stripe and spot features was reversed as in Fig. 2D. Participants were paid at the same scale as in Experiments 1-3 (highest paid £1.83). The task took 9.2 ± 4.4 minutes. The procedure was identical to Experiments 1-3. No participant was excluded from analysis. Pre-registration is available at https://osf.io/swde5. As above, a two-way ANOVA on feature-counterbalancing and curriculum-design predicting Phase II generalization accuracy revealed main effects on both factors (feature-counterbalancing: $F(1, 1) = 15.12, p < .001$; curriculum-design: $F(1, 1) = 11.1, p = .001$), but no interaction, $F(1, 236) = 0.77, p = .4$. While people indeed treat stripe and spot features differently, our results about curriculum design hold for both experiments.

Coding scheme

Two coders categorized participants self-reports independently. The first coder categorized all free responses, and 15% of the categorized self-reports were then compared against the second coder's. Agreement level was 97.6%.

We identified eight codes. `Ground truth`: equivalent to the ground truth causal relation in each experiment. Eg., *the length is multiplied by the number of lines and then the number of dots is subtracted* (Participant 43, Exp. 1). `Alternative`: equivalent to the alternative causal relation in each experiment. Eg., *the dots subtract from the segments by their number, and the number of lines is multiplied by the amount of segments* (Participant 461, Exp. 3). `Comp`: unclear or implicit about how two sub causal concepts should be combined. Eg., *the lines multiply the segments and the dots subtract the segments.* (Participant 451, Exp. 3). `Add 2`: add two segments to the recipient object, under the assumption that nothing happens if the agent object's feature value is 1 (stripe in Exps. 1 and 3, and spots in Exps. 2 and 4). Eg., *adds 2 segments to the stick only if there are 2 or more stripes on the egg* (Participant 35, Exp. 1). `Mult`: one feature of the agent object multiplies the recipient object. Eg., *the number of stripes multiplies the number of segments* (Participant 59, Exp. 1). `Subtraction`: one feature of the agent object is a subtractor to the recipient object. Eg., *each spot on the egg takes away one stick* (Participant 100, Exp. 1). `Complex`: describe the stimuli without generalizing a rule, or report a different rule for each observation. Eg., *3 dots means the sticks disappear, 2 dots means 2 sticks, 1 dot means add another stick.* (Participant 161, Exp. 1); *if there are more lines than dots it will increase in size. if there are more dots than lines it will decrease in size. an equal number of dots and lines will results in no change* (Participant 134, Exp. 1). `Uncertain`: not knowing, unsure, or confused about the learning stimuli. E.g., *i don't have a clue!* (Participant 57, Exp. 1).

Algorithm 1 Adaptor Grammar AG(τ, X)

Require: Type $\tau = t_0 \to \ldots \to t_k$
Require: Variables $X = \{x_0, \ldots, x_n\}$
  Sample $\lambda \sim U(0, 1)$
  if $\lambda \leq \lambda_1$ **then** ▹ Construct new hypothesis
    $z_L \sim \{z | t(z)_{\text{output}} = t_k\}$ ▹ Sample a term, e.g., `mult`
    $r \sim \mathbf{r}^{|X|}$ ▹ Sample a router, e.g., **SC**
    $i \leftarrow |t(z_L)|$ ▹ Grow RHS branches
    while $i > 0$ **do**
      $X' = r(X)$ ▹ Get routed variables
      $\tau' = t(X') \to t(z_L)_{i-1}$ ▹ Get type constraints
      AG(τ', X') ▹ Compose recursively
      $i \leftarrow i - 1$
    end while
  else ▹ Fetch existing hypothesis
    Return* $z \in C_\tau$ with probability λ_2
  end if

Adaptor grammar models

Causal programs. AG expects modular reuse of program fragments, so we formalize programs in combinatory logic (Schönfinkel, 1924). This solves the variable binding problem in generating functional programs (Crank & Felleisen, 1991), and is supported by recent work by Piantadosi (2021) arguing that combinatory logic provides a unified low-level coding system for human mental representations.

We start with defining a basic set of terms and types relevant to the task.[2] In combinatory logic, each term z is treated

[2]This choice to start with the right types for the task is for ex-

as a function, constrained by its input domain type and output co-domain type, written in the form of $t_{\text{input}} \rightarrow t_{\text{output}}$, with right-association by convention. Here, we default the last type t_n in a type $t_1 \rightarrow \ldots \rightarrow t_n$ to be the output type. Let agent and recipient objects be variables with type *obj*, we consider basic terms `getSpot`, `getStripe`, `getSegment`, each with type *object* $\rightarrow$ *int*, term `setSegment`, with type *obj* $\rightarrow$ *int* $\rightarrow$ *obj*, and terms `add`, `sub`, `mult`, each with type *int* $\rightarrow$ *int* $\rightarrow$ *int*. Term $\texttt{getSpot}_{obj \rightarrow int}$ takes an object as input, and returns the integer number of spots on this object. Term $\texttt{add}_{int \rightarrow int \rightarrow int}$ takes two integers as input, and return the sum of them as output. Likewise for the other terms above. We additionally consider four primitive integers 0, 1, 2 and 3 because these are the quantities appeared in the learning examples. Conveniently, we use $t(z)$ to read off the type of term z, e.g., $t(\texttt{getSpot})$ is *obj* $\rightarrow$ *int*. In addition, combinatory logic utilizes router terms such as **B**, **C**, **S** and **I** for variable binding. For a tree-like structure $[router, z_L, z_R]$, router **B** sends variable x first to the right-hand side z_R, and the result of this is then sent to the left-hand side z_L. In other words, $[\mathbf{B}, z_L, z_R](x)$ is executed as $z_L(z_R(x))$. Similarly, router **C** sends x to the left then right, router **S** sends x to both sides, and router **I** is an identity function that returns an input as it is. For N input variables, we concatenate N routers in corresponding order.

Program generation. We employ a tail recursion for composing terms as in Dechter et al. (2013) in order to efficiently satisfy type constraints. As demonstrated in Algo. 1, for a given target type $\tau = t_o \rightarrow \ldots t_k$, and a set of input variables $X = \{x_0, \ldots, x_n\}$, with probability λ_1 (Eq. 1) it goes into the construction step, and with probability λ_2 (Eq. 1) it returns a term with type τ and add this returned term to the cache (hence the Return* in Algo 1). The construction step starts by sampling a left-hand side term *LHS* whose output type is the same as the output type of τ, $t_{\text{output}}(\tau)$, which is t_k because we default the last element in a type to be the return type.

Following the notation in Liang et al. (2010), let N be the number of distinct elements in a collection of programs C, and M_z the number of times program z occurs in collection C:

$$\lambda_1 = \frac{\alpha_0 + Nd}{\alpha_0 + |C|}, \quad \lambda_2 = \frac{M_z - d}{|C| - Nd}. \tag{1}$$

Hyper-parameters $\alpha_0 > 0$ and $0 < d < 1$ in Eq. 1 control the degree of sharing and reuse. Since λ_1 is proportional to $\alpha_0 + Nd$, the smaller α_0 and d are, the less construction and more sharing we have. Similarly, λ_2 is proportional to M_z, hence the more frequently a program is cached, the higher weight it gets, regardless of its internal complexity. This definition of λ_2 instantiates the idea of boostrapping—the prior generation complexity of a cached program is overridden by its usefulness for composing future concepts. At its core, AG reuses cached programs as if they were conceptual primitives.

For simplicity, we assumed a flat prior initially such that terms sharing the same types have the same prior probability. Based on how many variables are fed to this stage, $|X|$, it then samples a router r of corresponding length from the set of all possible routers $\mathbf{r}^{|X|}$. This again is assumed to be a uniform distribution. For example, two variables corresponds to $4^2 = 16$ routers $\{\mathbf{BB}, \mathbf{BC}, \mathbf{BS}, \mathbf{BI}, \ldots\}$, and the probability of samling each router is $1/16 = 0.0625$. Router r then sends input variables to the branches. Now, the target type for the right-hand side of the tree is fully specified, because it has all the input types (routed by r) and a required output type (to feed into *LHS*). Therefore, we apply the same procedure iteratively to get this right-hand side subprogram *RHS*, returning the final program [*RT LHS RHS*]. The constructed program [*RT LHS RHS*] is then added to the program library $\mathcal{L}$ (caching). Note that after caching, the counter for a term z in library L could change, i.e. M_z in Eq. 1 gets updated, and preference for useful terms will then play a role in future program generation.

Inference. Given this probabilistic model, we are faced with the challenge of efficiently approximating a posterior distribution over latent programs. Here, we use known methods for sampling from Pitman-Yor processes (Liang et al., 2010; Pitman & Yor, 1997), such that conditional on a program library at any given moment, learners can make appropriate inferences about the probabilities of different explanations for new or salient events. This can be done via Gibbs sampling (Geman & Geman, 1984): for the i-th iteration, conditional on the library from previous iteration L_{i-1}, sample an updated library L_i and add it to the collection of samples.

During each iteration of Gibbs sampling, when searching for programs consistent with learning data, we adopted a breadth-first beam search under resource constraints. Since the search space grows exponentially as depth increases, we hypothesize that people are more likely to search shallowly than deeply. Therefore we draw generation depth $d \propto e^{-bd}$, where b is a parameter controlling how steep this exponential decay is. With generation depth d, we first enumerate a set of frames $\mathcal{F}$, where instead of applying Algo 1 recursively, we use typed program placeholders for *LHS*. We then sample a frame from $\mathcal{F}$ according to frame generation probabilities. The sampled frame is then "unfolded", replacing each placeholder with a program of the required type from the current library, yielding a set of fully-articulated programs M. If any program(s) $M^* \subseteq M$ produce learning data with likelihood 1, we stop the search, and sample $n = 3$ programs to enrich the library; otherwise, we sample another frame from $\mathcal{F}$ and repeat. If no programs are perfectly consistent

planatory convenience, and does not undermine our method's ability to grow new types and new basic terms. We could imagine starting with cognitively salient operators like those used here, or more basic operations as in Piantadosi (2021).

with the data after checking every frame from $\mathcal{F}$, we return with a `Nothing found` marker and skip to the next iteration. Because of memory constraints, we were able to enumerate frames up to depth $d = 2$, but this can easily produce deeply nested concepts as a result of iterated caching and reuse. We ran a grid search over integers 0-10 for parameter b in e^{-bd} on top of other model fitting procedures. When $b = 0$, depths $d = 1$ and 2 searches are equally likely, and as b increases, the model prefers depth $d = 1$. The best fitting $b = 6$, implying a stronger preference for depth $d = 1$ (see SI for additional analysis on search depth).

Thanks to the comprehensive search-check-sample procedure, we expect our Gibbs sampler to approximate the true posterior quickly and without the need for extensive burn-in. Since extensive Gibbs sampling is computationally expensive, and there is little value to running more than a handful of steps, we assume learners perform very little search within each phase. We thus approximate the population-level library distribution by running 1,000 simulations for chains of length h. During model fitting, we compare simulations for length $h = 1, 2, 3, 4$, and 5, and found that the best fitting model runs on a $h = 2$ chain (together with depth weight $b = 6$), suggesting strongly bounded use of resources (see SI for additional analysis on chain length).

Generalizations. We run the generative procedure of grammar $\mathcal{G}$ using the sampled libraries to approximates a distribution $Dist_M$ over latent causal programs, and make generalization predictions about new partially observed data $D^* = \langle A^*, R^*, ?\rangle$, producing a predicted distribution $Dist_P$ over generalizations. Since we compare our models to the aggregated behavioral data and the generalization process is not as computationally expensive as inference processes, we ran the generation process 10,000 times for a posterior predictive of generalization predictions that is reasonably representative of the population. Note that these implementations are needed to set up a fair comparison between models and aggregated participant data. While generating 10,000 hypotheses is certainly computationally demanding, this is not required for a single participant, but only to enable us to approximate a population-level distribution.

Rational rules model

Following (Bramley et al., 2018; Goodman et al., 2008; Zhao, Lucas, et al., 2022), we implemented a Probabilistic Context-Free Grammar $\mathcal{G}_r = \{S, T, N, \Theta\}$, where S is the starting symbol, T a set of production rules, N the set of terminal nodes, and Θ the production probabilities. In order to retain a close match with the adaptor grammar's initial concept library, we considered production rules as follows:

$$S \to \text{add}(A, A) \mid \text{sub}(A, A) \mid \text{mult}(A, A)$$

$$A \to S \mid B$$

$$B \to C \mid D$$

$$C \to \text{stripe} \mid \text{spot} \mid \text{segment}$$

$$D \to 0 \mid 1 \mid 2 \mid 3$$

The pipe symbol | represents "or", meaning that the symbol on the left-hand side of the arrow symbol $\to$ can transform to either of the symbols on the right-hand side of $\to$. As with the adaptor grammar models, we assigned uniform prior production probabilities: let Γ_L be the set of production rules all starting with L, i.e. any production rule $\gamma \in \Gamma_L$ is of the form $L \to K$, where K can be any symbol in grammar $\mathcal{G}_r$, the production probability for each $\gamma \in \Gamma_L$ is $\frac{1}{|\Gamma_L|}$. Since grammar $\mathcal{G}_r$ can produce infinitely complex causal concepts, we fixed a generation depth $d = 40$ in our implementation to cover the ground truth concepts. If d is set too small, like the same constraint we set to the adaptor grammar models, $\mathcal{G}_r$ cannot land on the ground truth by design and therefore not so useful in model comparison (see Zhao, Bramley, et al., 2022). As in the adaptor grammar models, we used a deterministic likelihood function to evaluate each concept generated by grammar $\mathcal{G}_r$, essentially discarding all generated concepts that fail to explain all the evidence. We set $n = 100,000$ to have good coverage of rules up to and beyond the degree of complexity seen in human responses. Generalization predictions are made following the same procedure as the adaptor grammar models: Apply the approximated posterior rules with the partially observed data $D^* = \langle A^*, R^*, ?\rangle$ in generalization tasks, and marginalize over the predicted R'^* as an approximated posterior predictive.

Similarity-based model

Let d_l be a learning example data point, consisting of an agent, a recipient object, and a result object; d_g a generalization task data point, consisting of only an agent and a recipient objects. Let $\texttt{stripe}(x)$ be the number of stripes of object x, we can measure the similarity between learning example d_l and generalization task d_g in terms of stripes by taking the absolute difference $||\texttt{stripes}(A)_{d_l} - \texttt{stripes}(A)_{d_g}||$, denoted by $\delta_{\texttt{stripes}}(d_l, d_g)$. Taking all three features stripes, spots and segments into account, the feature difference Δ between learning example d_l and generalization task d_g can be measured by $\Delta(d_l, d_g) = a \cdot \delta_{\texttt{stripe}}(d_l, d_g) + b \cdot \delta_{\texttt{spot}}(d_l, d_g) + c \cdot \delta_{\texttt{segment}}(d_l, d_g)$. With these measures, we can define a similarity score

$$\sigma_{\text{sim}}(d_l, d_g) = e^{-\Delta(d_l, d_g)}$$

such that the more similar d_l and d_g are (smaller distance Δ), the higher the similarity σ_{sim}. When the two data points share the same agent and recipient objects, similarity score σ_{sim} reaches its max of = 1. When making generalization predictions, this model first computes similarity score σ_{sim} between the current generalization task g_i with all the available learning examples $\{l_1, \ldots, l_k\}$, resulting in

$S = \{\sigma_{\text{sim}}(d_{l_1}, d_{g_i}), \ldots, \sigma_{\text{sim}}(d_{l_k}, d_{g_i})\}$. Now for this generalization task g_i, it mimics $\texttt{result}(d_{l_k})$ with confidence $\sigma_{\text{sim}}(d_{l_k}, d_{g_i})$. Let $n = \texttt{result}(d_{l_k})$, task g_i predicts $p(n) = \texttt{result}(d_{l_k}) \cdot \sigma_{\text{sim}}(d_{l_k}, d_{g_i})$. Marginalizing over all possible result segment values n gives the distribution over task g_i's predicted result segment values.

Linear regression model

Let the number of stripes, spots and segments in each learning example be the independent variables, and the resulting stick length R') be the dependent variable. We fit a linear regression model after each phase of the experiment with formula

$$R' \sim a \cdot \texttt{stripe}(A) + b \cdot \texttt{spot}(A) + c \cdot R + \epsilon.$$

We made generalization predictions using fitted parameters and the requisite generalization task's feature values. We rounded the predicted result segment number to the two nearest integers in order to match the required prediction output.

Multinomial logistic regression model

We treated each possible result segment value as categorical value (instead of continuous as in the linear regression case), and fit a multinomial logistic regression model to predict the probability of each result segment value using a formula same as the one used in the linear regression model, with the `nnet` package in R. By fitting the model, we call the `pred` function to gather probabilistic predictions about the possible result segment values for each trial. We normalize this probabilistic prediction to ensure this is a probabilistic distribution.

Gaussian process model

Treating each learning example as three-dimensional input (stripes, spots, segments) with a one-dimensional output (result segments), we fit a Gaussian Process (GP) regression model with radial basis function kernels, each per feature x_f:

$$K(x_f, x'_f) = \exp(-\frac{||x_f - x'_f||}{2\sigma^2})$$

We used the `GPy` package in Python to fit the model. Conditioning on the three dimensional input for each generalization task, the fitted GP regression model outputs a Gaussian distribution over possible segment lengths $\mathcal{N}(\mu, \sigma^2)$. We then bin this distribution over the possible discrete segment values for comparison with empirical data.

Cross validation

We used cross validation to evaluate models against behavioral data in generalization tasks on log likelihood fits. To do this, we collapsed data from all four experiments by curriculum c, keeping how many people n chose which segment number $y \in [0, 16]$ in each task i, resulting in data $\mathcal{D} = \{n_{ciy}\}$. We then let each computational model generate a distribution P_{ci} over all possible segment numbers $Y = \{0, 1, \ldots, 16\}$ for task i in curriculum c. Since many model predictions are point estimates, or centered on only a few segment numbers, we considered a trembling hand noise parameter $h \in (0, \frac{1}{|Y|})$ such that for a probability distribution $P(Y)$:

$$P^h(Y = y) = \frac{P(Y = y) + h}{1 + h|Y|}. \quad (2)$$

Essentially, we add noise h to each random variable in set Y to avoid 0 likelihoods. The denominator ensures $P^h(Y)$ is still a probability. Different from softmax functions, $P^h(Y)$ stays close to the shape of $P(Y)$ when h is small, and therefore best maintains each model's "raw" degree of confidence on those 1 or 2 predictions. Log likelihood of a model producing data $\mathcal{D}$ is thus given by:

$$LL = \sum_{c=c_1}^{c_k} \sum_{i=t_1}^{t_j} \sum_{y=y_1}^{y_m} \ln(P^h_{ci}(Y = y)) \cdot n_{ciy}. \quad (3)$$

For each run of the cross validation, we hold out one curriculum c_{test}, and fit the noise parameter h on the other three curricula using maximum likelihood estimation (MLE) with the `optim` function in R. Note that for model AGR, an additional weight parameter λ is jointly fitted. Then we compute LL_{test} on curriculum c_{test} with the fitted parameters. Summing over LL_{test} for all four curricula serves as the total log likelihood fit LL for the model. As a baseline, choosing randomly yields $LL_{\text{rand}} = 570 \times 16 \times \ln(\frac{1}{17}) = -25838.91$, for there were 570 participants, each completing $8 \times 2 = 16$ tasks, where in each task there were 17 possible responses (final stick lengths, including 0) to choose from. Any value smaller than LL_{rand} is improvement over an eyes-closed baseline.

Data availability

Data reported in this study are available on the Open Science Framework https://osf.io/9awhj/.

Code availability

Implementations of all the above mentioned models and analysis are freely-accessible at https://github.com/bramleyccslab/causal_bootstrapping and https://osf.io/9awhj/.

Acknowledgements

This work was supported by by an EPSRC New Investigator Grant (EP/T033967/1) to NB and CL. Thanks to XinXin Zhu for help with coding the free text responses. Thanks to Frank Mollica, Tadeg Quillien, Simon Valentin, Charles

Kemp, Noah Goodman, Eric Schulz, Robert Hawkins, the reviewers and editors for valuable feedback on the manuscript.

Author contributions

BZ, NB and CL designed the studies. BZ and CL devised the main and alternative model. BZ and NB designed the experiments. BZ implemented the model, collected data, performed analyses and drafted the manuscript. NB and CL supervised all aspects of the project. All authors discussed the results and revised the manuscript.

Competing interests

The authors declare no competing interests.

References

Acerbi, L., Vijayakumar, S., & Wolpert, D. M. (2014). On the origins of suboptimality in human probabilistic inference. *PLoS computational biology*, *10*(6), e1003661.

Bever, T. G. (1970). The cognitive basis for linguistic structures. *Cognition and the development of language*.

Bonawitz, E., Denison, S., Gopnik, A., & Griffiths, T. L. (2014). Win-stay, lose-sample: A simple sequential algorithm for approximating Bayesian inference. *Cognitive Psychology*, *74*, 35–65.

Bowers, M., Olausson, T. X., Wong, L., Grand, G., Tenenbaum, J. B., Ellis, K., & Solar-Lezama, A. (2023). Top-down synthesis for library learning. *Proceedings of the ACM on Programming Languages*, *7*(POPL), 1182–1213.

Bramley, N. R., Dayan, P., Griffiths, T. L., & Lagnado, D. A. (2017). Formalizing Neurath's ship: Approximate algorithms for online causal learning. *Psychological Review*, *124*(3), 301.

Bramley, N. R., Rothe, A., Tenenbaum, J., Xu, F., & Gureckis, T. (2018). Grounding compositional hypothesis generation in specific instances. *Proceedings of the 40th Annual Meeting of the Cognitive Science Society*.

Bramley, N. R., & Xu, F. (2023). Active inductive inference in children and adults: A constructivist perspective. *Cognition*.

Brown, G. D., Neath, I., & Chater, N. (2007). A temporal ratio model of memory. *Psychological Review*, *114*(3), 539–576.

Carey, S. (2004). Bootstrapping & the origin of concepts. *Daedalus*, *133*(1), 59–68.

Chater, N. (2018). *The mind is flat: The illusion of mental depth and the improvised mind*. Penguin UK.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, *24*(1), 87–114.

Craik, K. J. W. (1952). *The nature of explanation* (Vol. 445). CUP Archive.

Crank, E., & Felleisen, M. (1991). Parameter-passing and the lambda calculus. *Proceedings of the 18th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 233–244.

Culbertson, J., & Schuler, K. (2019). Artificial language learning in children. *Annual Review of Linguistics*, *5*, 353–373.

Dasgupta, I., & Gershman, S. J. (2021). Memory as a computational resource. *Trends in Cognitive Sciences*, *25*(3), 240–251.

Dechter, E., Malmaud, J., Adams, R. P., & Tenenbaum, J. B. (2013). Bootstrap learning via modular concept discovery. *Twenty-Third International Joint Conference on Artificial Intelligence*.

Della Sala, S. (2010). *Forgetting*. Psychology Press.

Ellis, K., Wong, C., Nye, M., Sable-Meyer, M., Cary, L., Morales, L., Hewitt, L., Solar-Lezama, A., & Tenenbaum, J. B. (2020). Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning. *arXiv preprint arXiv:2006.08381*.

Fernandino, L., Tong, J.-Q., Conant, L. L., Humphries, C. J., & Binder, J. R. (2022). Decoding the information structure underlying the neural representation of concepts. *Proceedings of the National Academy of Sciences*, *119*(6), e2108091119.

Flesch, T., Juechems, K., Dumbalska, T., Saxe, A., & Summerfield, C. (2022). Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, *110*(7), 1258–1270.

Fränken, J.-P., Theodoropoulos, N. C., & Bramley, N. R. (2022). Algorithms of adaptation in inductive inference. *Cognitive Psychology*, *137*, 101506.

Gelpi, R., Prystawski, B., Lucas, C. G., & Buchsbaum, D. (2020). Incremental hypothesis revision in causal reasoning across development. *Proceedings of the 42th Annual Conference of the Cognitive Science Society*.

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 721–741.

Gershman, S., & Goodman, N. (2014). Amortized inference in probabilistic reasoning. *Proceedings of the 36th annual meeting of the cognitive science society*.

Gobet, F., Lane, P. C., Croker, S., Cheng, P. C., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, *5*(6), 236–243.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive Science*, *32*(1), 108–154.

Gopnik, A., & Meltzoff, A. N. (1997). *Words, thoughts, and theories*. MIT Press.

Gravitz, L. (2019). The forgotten part of memory. *Nature*, *571*(7766), S12–S12.

Griffiths, T. L., Lieder, F., & Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science*, *7*(2), 217–229.

Isaac Newton. (1675). Letter to robert hooke.

Jern, A., Chang, K.-M. K., & Kemp, C. (2014). Belief polarization is not always irrational. *Psychological Review*, *121*(2), 206.

Johnson, M., Griffiths, T. L., Goldwater, S., et al. (2007). Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. *Advances in neural information processing systems*, *19*, 641.

Kahneman, D., Slovic, S. P., Slovic, P., & Tversky, A. (1982). *Judgment under uncertainty: Heuristics and biases*. Cambridge university press.

Keil, F. C. (2006). Explanation and understanding. *Annual Review of Psychology*, *57*, 227–254.

Kemp, C., & Tenenbaum, J. B. (2008). The discovery of structural form. *Proceedings of the National Academy of Sciences*, *105*(31), 10687–10692.

Klein, G. A. (2017). *Sources of power: How people make decisions*. MIT press.

Krueger, K. A., & Dayan, P. (2009). Flexible shaping: How learning in small steps helps. *Cognition*, *110*(3), 380–394.

Kuhn, T. S. (1970). *The structure of scientific revolutions* (Vol. 111). Chicago University of Chicago Press.

Liang, P., Jordan, M. I., & Klein, D. (2010). Learning programs: A hierarchical Bayesian approach. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 639–646.

Lombrozo, T. (2012). Explanation and abductive inference. In K. J. Holyoak & R. G. Morrison (Eds.), *The oxford handbook of thinking and reasoning*. Oxford University Press.

Lucas, C. G., Griffiths, T. L., Williams, J. J., & Kalish, M. L. (2015). A rational model of function learning. *Psychonomic Bulletin & Review*, *22*(5), 1193–1215.

Mahoney, J., & Schensul, D. (2006). Historical context and path dependence. In *The oxford handbook of contextual political analysis*. Oxford University Press.

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

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Prentice-Hall.

Nørby, S. (2015). Why forget? on the adaptive value of memory loss. *Perspectives on Psychological Science*, *10*(5), 551–578.

Piantadosi, S. T. (2021). The computational origin of representation. *Minds and machines*, *31*(1), 1–58.

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.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2016). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, *123*(4), 392–424.

Pitman, J., & Yor, M. (1997). The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *Annals of Probability*, *25*, 855–900.

Quine, W. V. O., & Ullian, J. S. (1978). *The web of belief* (Vol. 2). Random house New York.

Rich, A. S., & Gureckis, T. M. (2018). The limits of learning: Exploration, generalization, and the development of learning traps. *Journal of Experimental Psychology: General*, *147*(11), 1553–1570.

Rule, J. S., Tenenbaum, J. B., & Piantadosi, S. T. (2020). The child as hacker. *Trends in Cognitive Sciences*.

Sanborn, A. N., & Chater, N. (2016). Bayesian brains without probabilities. *Trends in Cognitive Sciences*, *20*(12), 883–893.

Sanborn, A. N., Griffiths, T. L., & Navarro, D. J. (2010). Rational approximations to rational models: Alternative algorithms for category learning. *Psychological Review*, *117*(4), 1144–1167.

Schönfinkel, M. (1924). Über die bausteine der mathematischen logik. *Mathematische Annalen*, (92), 305–316.

Searcy, S. R., & Shafto, P. (2016). Cooperative inference: Features, objects, and collections. *Psychological Review*, *123*(5), 510–533.

Sorscher, B., Ganguli, S., & Sompolinsky, H. (2022). Neural representational geometry underlies few-shot concept learning. *Proceedings of the National Academy of Sciences*, *119*(43), e2200800119.

Thaker, P., Tenenbaum, J. B., & Gershman, S. J. (2017). Online learning of symbolic concepts. *Journal of Mathematical Psychology*, *77*, 10–20.

Tian, L., Ellis, K., Kryven, M., & Tenenbaum, J. (2020). Learning abstract structure for drawing by efficient motor program induction. *Advances in Neural Information Processing Systems*, *33*, 2686–2697.

Tomov, M. S., Dorfman, H. M., & Gershman, S. J. (2018). Neural computations underlying causal structure learning. *Journal of Neuroscience*, *38*(32), 7143–7157.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327–352.

Van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science*, *32*(6), 939–984.

Vul, E., Goodman, N., Griffiths, T. L., & Tenenbaum, J. B. (2014). One and done? optimal decisions from very few samples. *Cognitive Science*, *38*(4), 599–637.

Vul, E., Griffiths, T., Levy, R., Steyvers, M., & McKenzie, C. R. (2009). Rational process models. *Proceedings of the Thirty-first Annual Meeting of the Cognitive Science Society*.

Wong, C., McCarthy, W. P., Grand, G., Friedman, Y., Tenenbaum, J. B., Andreas, J., Hawkins, R. D., & Fan, J. E. (2022). Identifying concept libraries from language about object structure. *arXiv preprint arXiv:2205.05666*.

Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., & Meder, B. (2018). Generalization guides human exploration in vast decision spaces. *Nature Human Behaviour*, *2*(12), 915–924.

Zhao, B., Lucas, C. G., & Bramley, N. R. (2022). How do people generalize causal relations over objects? a non-parametric bayesian account. *Computational Brain & Behavior*, *5*, 22–44.

Zhao, B., Bramley, N. R., & Lucas, C. G. (2022). Powering up causal generalization: A model of human conceptual bootstrapping with adaptor grammars. *Proceedings of the 44th Annual Meeting of the Cognitive Science Society*, 1819–1826.