Learning abstract structure in drawings by motor-efficient program induction

Anonymous CogSci submission

Abstract

Humans learn quickly and flexibly by building new knowledge out of old. This capacity requires a prior inventory of concepts, which can serve as building blocks in a 'language of thought'. How do people acquire good concepts that will become useful for learning later on? Here we study the learning of abstract structure for drawing symbolic figures. We find that subjects spontaneously learn new abstract procedural rules after brief experience drawing new symbolic stimuli. We propose a generative model that learns *programs* for drawing by flexibly recombining parts from a *library of primitives*. Given the same limited training as humans, no direct supervision, and a bias for efficient motor actions, this model learns new useful abstractions that successfully transfer to test tasks. The similarity of model and human behavior on test tasks suggests that humans may be extending their language of thought by "inventing" new programming subroutines that combine with low-level motor constraints to guide the future interpretation and generation of drawings.

Keywords: drawing; learning abstract structure; language of thought; program induction; motor efficiency; generalization.

Introduction

A remarkable feature of cognition is the capacity to understand, learn, and create new concepts by building new knowledge out of old. In many cases, this depends on having in place flexible systems of knowledge that guide our current explanation and future learning (Chi, Glaser, & Farr, 2014; Harlow, 1949; Bartlett, 1932). For example, expert dancers can leverage prior knowledge of dance moves and sequencing rules, abstracted from the motor or sensory specifics, to quickly imitate a new dance.

How is abstract structure learned? One promising approach to modeling reasoning and learning is to frame abstract knowledge as derived from a probabilistic *language of thought* (Fodor, 1975; Goodman, Tenenbaum, Feldman, & Griffiths, 2008). Here concepts are represented as programs, or symbolic procedures, generated from a mental library of *primitives* and *means of combination*. This approach has proved successful in modeling a diversity of abstract knowledge, including in people's rapid learning of new handwritten characters (Lake, Salakhutdinov, & Tenenbaum, 2015), in inferring 3D structure from 2D images (Erdogan, Yildirim, & Jacobs, 2015), and in inferring the underlying parts and relations that define categories of objects in visual scenes (Jaekel, Savova, & Tenenbaum, 2009).

In the probabilistic language of thought, learning can arise from modifying probabilities or weights assigned to different primitives, or even by "inventing" new primitives by recombining older ones (Ellis, Morales, Sablé-Meyer, Solar-Lezama, & Tenenbaum, 2018). While powerful in theory, few empirical studies so far have tested whether this approach can explain human learning of abstract structure [although see (Cheyette & Piantadosi, 2017; Rule, Schulz, Piantadosi, & Tenenbaum, 2018)].

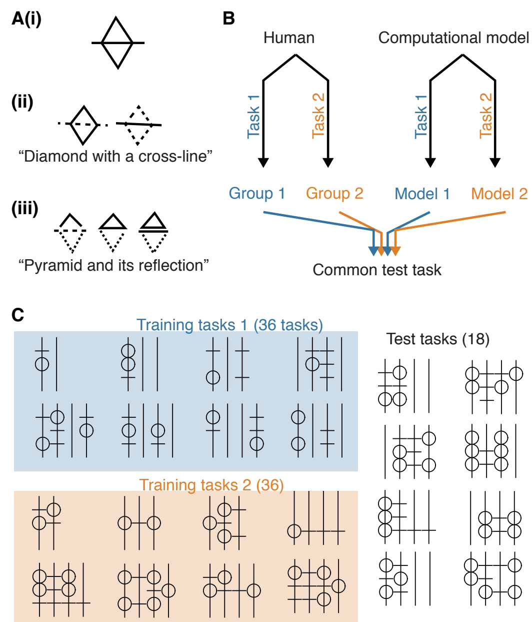

Figure 1: Background and overview. **(A)** Abstract structural descriptions are revealed in how we draw. Try drawing the object in **A(i)** before reading on. **(Aii and Aiii)** depict the most common ways people draw (line style indicates a continuous stroke) given different descriptions of what the object represents. (Van Sommers, 1984). **(B)** Humans and models were trained and tested on the same sets of drawing tasks. **(C)** Representative tasks in Training sets 1 and 2, and in the common Test set.

Here we introduce an experimental paradigm to study the learning of program-like abstract structure for *drawing*. Drawing is well-suited for this problem because it depends on diverse systems of knowledge (Goodnow, 1977; Forbus, Usher, Lovett, Lockwood, & Wetzel, 2011) - ranging from low-level motor efficiency constraints to high-level program-like structure relevant for real-life tasks - and is tractable since it is grounded in sensory-motor space. Our study is motivated by the observation that there is inherent ambiguity in how to copy an image: What strokes should be used, and in what sequence? People resolve this ambiguity via drawing biases. Distinct from prior studies of concept-learning for handwritten characters (Lake et al., 2015), our study focuses on biases related to abstract procedures, such as those that depend on an image's relation to real-world objects (Van Sommers, 1984; Karmiloff-Smith, 1990; Forbus et al., 2011; Bartlett, 1932; Cheng, McFadzean, & Copeland, 2001; Fan, Yamins, & Turk-Browne, 2018).

For example, imagine drawing Figure 1A(i). If told it is a "diamond with a cross-line", people tend to draw it according to the sequence in Figure 1A(ii). If told it is an "Egyptian pyramid and its reflection on water", people tend to follow

Figure 1A(iii) (Van Sommers, 1984). We hypothesize that humans may encode drawings as generative programs, such that, given an image, a person infers a program, in a process termed *program induction*, that captures the variety of abstract structure underlying the drawing.

To test this hypothesis, we performed behavioral experiments and computational modeling. We found that people spontaneously learned new composite drawing concepts from a single session of drawing. We modeled learning by combining program induction with motor efficiency constraints. Our results indicate that people extend their language for thinking by "inventing" new abstract program subroutines, which interact with low-level motor constraints to guide the subsequent interpretation and generation of drawings.

Computational Models

We reasoned that drawing is subject to two primary constraints related to low-level motor efficiency and high-level abstract structure. We model motor efficiency by assuming that choices of stroke sequence minimize a *motor cost* that approximates both musculo-skeletal constraints and motor effort (Goodnow, 1977; Van Sommers, 1984). All possible motor trajectories that can generate a drawing are assigned a cost and lower-cost trajectories are selected with higher probability. Parameters describing subjects' motor constraints are inferred from their motor trajectories during training tasks.

We model high-level abstract structure as programs that combine drawing primitives using higher-order rules like repetition, symmetry, and hierarchy. Given an image, drawing depends on probabilistic inference of an underlying program, or *program induction*. Models learn to infer programs from images with no access to motor trajectories, directly mirroring the conditions experienced by humans.

We formalize in the Hybrid Model (HM) (Table 1) the hypothesis that a combination of motor efficiency constraints and program induction would best capture subjects' behaviors. The Motor Cost (MC) model captures only motor efficiency costs. The Motor + Switching Cost (MC+) model extends MC by adding additional costs related to switching between particular drawing components. The Program Induction (PI) model uses program induction with no motor constraints. Lastly, the Null model produces trajectories by permuting strokes in a random order.

Table 1: Models used in this study

Model	Training Set	Abbrev.
Null	none	Null
Motor Cost	1, 2 (motor data)	MC1, MC2
Motor+Switching Cost	1, 2 (motor data)	MC1+, MC2+
Program Induction	1, 2 (images)	PI1, PI2
Hybrid	1, 2 (mot. & im.)[1]	HM1, HM2

Motor Cost Model (MC) Let t represent a motor trajectory - a sequence of discrete segmented strokes. All models and analyses used segmented strokes (see "Analysis of behavior"). We define a *feature extractor* $\phi(\cdot)$ that maps a trajectory t to a real-valued vector $\phi(t)$ of low-level trajectory features.[2] Given an input image I, the model predicts a drawing trajectory t with probability

$$P(t|I) = 1\left[t \text{ draws } I\right] \frac{\exp\left(-\theta \cdot \phi(t)\right)}{\sum_{t'} 1\left[t' \text{ draws } I\right] \exp\left(-\theta \cdot \phi(t')\right)} \quad (1)$$

for weight-vector θ, where $\theta \cdot \phi(t)$ is the cost of trajectory t, and where $1[t \text{ draws } I]$ is 1 when t draws I, 0 otherwise. Given a set of N training images $\{I_n\}_{n=1}^N$ and paired motor trajectories $\{t_n^s\}_{n=1}^N$ for subject s, the model estimates for each subject θ^s via regularized maximum likelihood, i.e. minimizing

$$-\sum_{n=1}^{N} \log P(t_n^s|I_n^s) + \lambda ||\theta^s||_2^2 \quad (2)$$

with a suitable coefficient of regularization λ. In practice, instead of enumerating all t' to calculate the denominator in Eq. (1), we randomly sample 150 t' such that $1[t \text{ draws } I] = 1$. Performing this entire procedure multiple times with independent random samples gave similar estimates of θ.

Motor + Switching Cost Model (MC+). This model is identical to MC, except for two additional parameters that capture two different mental switching costs.[3]

Program-Induction Model (PI). We treat image understanding as Bayesian inference over the most likely program ρ that generated each image, given a library of primitives L. The model recovers ρ maximizing:

$$P(\rho|I,L) \propto \underbrace{P(I|\rho)}_{\text{likelihood: } 1[\rho \text{ draws } I]} \quad \underbrace{P(\rho|L)}_{\text{prior}} \quad (3)$$

For test images we relax the likelihood function to pixel-wise L2 distance.

The library of primitives L is initialized with primitives shown in Figure 4A: **line()**, returns a unit line. **circle()**, returns a unit circle. **reflect(*P*, θ)**, returns the drawing program P, reflected across the axis defined by the angle θ. **transform(*P*, *T*)**, returns P after the affine transformation defined by T. **connect(*P*1, *P*2)**, returns the superposition of drawings by programs $P1$ and $P2$. **repeat(*P*, *N*, *T*)**, returns a program that repeatedly draws and transforms P by T, N times. **get_transform(*x*, *y*, θ, *s*)**, returns affine transformation T, which translates by (x, y), rotates by θ and scales by s. T does not produce a drawing on its own, but is an argument used in the primitives *transform* and *repeat*.[4] These

[1]Program induction used images; Motor costs used motor data.

[2]**Start**, position of first stroke, in distance from the top-left corner. **Distance**, cumulative distance between strokes. **Direction**, cumulative distance between strokes, after projection onto the top-left to bottom-right axis. **Verticality**, cumulative horizontal distance between strokes.

[3]**Typechunks**, number of switches between stroke types (e.g., C to Lh); **Skewers**, tendency to finish a "skewer" (vertical line and strokes on the line) before moving to the next skewer. This contrasts with first making the vertical grating before drawing other strokes.

[4]Parameters N, θ, x, y, and s, are discretized and drawn from multinomial distributions.

primitives encode the support of a prior distribution over programs.

Given training images $\{I_n\}_{n=1}^N$ the model updates its library L by searching to maximize:

$$P(L|\{I_n\}_{n=1}^N) \propto P(L)\prod_{n=1}^{N}\sum_{\rho} P(I_n|\rho)P(\rho|L) \tag{4}$$

This maximization is achieved by both reweighing a distribution over primitives in L (e.g., increasing probability of **line**) and by adding new primitives to L by combining old ones (e.g., combining **repeat**, **line** and **x=1** to invent **repeat(line, N, x=1)**). Equations 3 and 4 are intractable, because they require computing the infinite set of all possible programs. We approximate inference using the DreamCoder program synthesis algorithm (Ellis et al., 2018). DreamCoder alternates between inferring a program for each image, updating the library L to assign more probability mass to those programs, and training a neural network, $Q(\rho|I)$, to predict a probability distribution over programs ρ likely to explain image I. This corresponds to iterating the following equations:[5]

$$\rho_n = \underset{\substack{\rho:\\ Q(\rho|I_n)\text{ is large}}}{\arg\max}\; P(I_n|\rho)P(\rho|L) \tag{5}$$

$$L = \arg\max_L P(L)\prod_{n=1}^{N} P(\rho|L) \tag{6}$$

$$\text{Train } Q(\rho|I) \approx P(\rho|I,L) \propto P(I|\rho)P(\rho|L) \tag{7}$$

The PI model programs do not represent ordering of strokes. For example, a program that translates a vertical line four times could correspond to either a left-to-right or a right-to-left drawing sequence. Thus we "ground" these programs into possible motor trajectories t. Each program ρ generates a set of *admissible* t, which includes all programs that can be generated from independently and recursively permuting the order of all its subprograms[6]. The probability of t given program ρ in the PI model is therefore given by:

$$P(t|\rho) = \frac{1\,[t \text{ admissible for } \rho]}{\sum_{t'} 1\,[t' \text{ admissible for } \rho]} \tag{8}$$

The Hybrid Model (HM). This model combines motor efficiency and programmatic abstractions. Like PI, the HM model performs program induction, but here the model weighs the admissible trajectories not uniformly, but instead by motor costs. We use the most generic motor costs θ^{gen} (*Start*, *Direction*, and *Distance*) averaged over all subjects; thus any learning-related differences in behavior reflect only learned program structure. The probability of trajectory t given program ρ in the HM model is:

$$P(t|\rho,\theta) = \frac{1\,[t \text{ admissible for } \rho]\exp(-\theta^{gen}\cdot\phi(t))}{\sum_{t'} 1\,[t' \text{ admissible for } \rho]\exp(-\theta^{gen}\cdot\phi(t'))} \tag{9}$$

Scoring model-human similarity. For each combination of test image, subject, and model, we scored similarity of model predictions to human behavior as the expected Damerau–Levenshtein edit distance between the human trajectory t^{h} and the model's predicted trajectory t, with the expectation taken over the model's distribution over t: $distance = \sum_{t'} \text{Damerau–Levenshtein}(t', t^h)P(t'|\rho,\theta)$

Methods

We test whether humans can extend their concept library from modest experience drawing a controlled set of training stimuli.

The drawing task. Two groups of subjects were trained on different training stimuli, but were tested on the exact same test stimuli (Figure 1B,C).

Subjects. Subjects (N = 104 (58 M, 44 F, 2 failed to respond), Age = 35.0 +/- 9.3 (mean/SD)) were recruited on Amazon Mechanical Turk and paid $3.00 for 15-20 minutes. Subjects gave informed consent. The study was approved by our institution's Institutional Review Board.

Stimuli. See examples in Figure 1C. Two sets of stimuli were generated with differentprobabilistic algorithms. Both had repeated vertical lines (2, 3, or 4), but differed in the other components. Set 1 had vertically grouped strokes (lines and circles) superimposed on the vertical lines, while Set 2 had horizontally-oriented groups of strokes, sampled from an library of objects (e.g. dumbbells (o–o), lollipops (–o) and poles (—)). We randomly generated 250 stimuli for each training set, from which we manually selected 36 representative samples. The common Test set included 18 manually designed ambiguous images.

Procedure. The experiment was presented in a web browser using PsiTurk (Gureckis et al., 2016) on a touchscreen device (phone, tablet, or laptops). The instructions read: *You will learn to write letters from an alphabet of an alien civilization recently discovered by astronauts. Scientists would like to study how people learn to write new alphabets. Your task is to copy the letters. Try to be quick, but it is also important to be accurate! Letters are taken from the same alphabet. But letters get harder over time, so try to learn from the earlier trials!*

On each trial a single stimulus was presented top-center of the screen. The subjects copied it on "sketchpad" directly below the stimulus without explicit time constraints or evaluative feedback. Subjects first saw seven simple stimuli (e.g the first three stimuli in Figure 1C), followed by 13 more training stimuli of varying difficulty. Next subjects copied the 18 testing intermixed with the remaining 16 training stimuli, in orders randomized for each subject.

Analysis of behavior. Motor trajectories were segmented into discrete *strokes* based on the times when the finger initially touched the screen and when it was removed. Each stroke was classified as either a "vertical line" (LL), "horizontal line" (Lh), or "circle" (C).

To calculate frequencies of horizontal/vertical transitions, each stroke was first "snapped onto" its position in a $3\times N$ (row $\times$ column) grid, where N is the number of vertical lines in the "grating", thereby identifying each stroke by its type and coordinate. Stroke transitions were classified as either horizontal (same row, different column), vertical (same column), or undefined.

[5] Programs are represented in λ-calculus. Eq. 5 corresponds to proposing programs ρ in order of decreasing probability $Q(\rho|I)$, Eq. 6 updates L via probabilistic grammar induction over $\{\rho_n\}_{n=1}^N$; and Eq. 7 describes training Q to predict programs conditioned on an image. DreamCoder trains Q on (I, ρ) pairs drawn both from 'replays' of programs found when solving Eq. 5, and on 'dreams', or programs sampled from its generative model L, inspired by the Helmholtz machine (Hinton, Dayan, Frey, & Neal, 1995). Together, the three equations maximize a lower bound on Eq. 4.

[6] For example, the program *connect(connect(line, circle), connect(circle, circle))* has two subprograms (the inner connects($\cdot$)), each of which is allowed to emit any ordering of its own subprograms (e.g., *line* and *circle* for the first inner *connect*).

Results

Single-session learning of abstract structure in drawings. We first assessed the relative frequencies of horizontal vs. vertical transitions between strokes in the two groups of subjects. We expected subjects trained on Task 2 (horizontally structured objects) to produce a relatively higher frequency of horizontal transitions compared to subjects trained on Task 1 (vertically structured objects). Indeed, behavior during training exhibited this difference between the two groups (Figure 2 C, "Training"). Importantly, this behavioral difference persisted on Test stimuli (Figure 2C, "Test"), which were identical across all subjects. Thus, subjects learned new biases that influenced their interpretation of the Test set.

Drawing behavior exhibits a variety of statistical regularities, reflecting, in part, motor efficiency constraints (Van Sommers, 1984; Goodnow, 1977). We tested whether the changes to drawing behavior in the two groups of subjects reflected changes to such constraints. Using the MC model we estimated each subject's weights placed on four motor constraints: *Start, Distance, Direction*, and *Verticality* [see Methods]. We found a strong difference between the two groups in their *Verticality* weights, in the direction consistent with stronger vertical biases for Group 1, consistent with the analysis of transition frequencies above (Figure 2D). We also found slight differences in *Start* and *Distance*, suggesting that learning may have been more complex than simply updating biases towards vertical or horizontal transitions. Consistent with people having learned a variety of structures, MC with parameters fit to subjects' own Training sequences better predicted Test behavior than did models fit to others from the same Training group (Figure 2E).

Program-like structure in behavior. Visual inspection of pilot data suggested a variety of learned structures (Figure 3B). For example, the "skewers" strategy involved drawing a vertical line, immediately followed by the objects "skewered onto it" (Subject 1). Several other strategies involved drawing the vertical gratings followed by either: all strokes of the same type, such as circles or lines (Subject 2), or consecutive drawing of vertical (Subject 3), or horizontal (Subject 4) components. We therefore extended the MC model to the the MC+ model, which included two additional "switching costs": one that captures a bias to finish each vertical skewer before moving on to the next (*Skewer*) and another that captures a bias to group all strokes of a given type (*Typechunk*).

We found that covariation along these two dimensions with respect to *Verticality* (previously shown to separate the two groups of subjects) captured meaningful variation in behavior (Figure 3A, B). Subjects at the extremes of the weight distributions mapped onto the structures illustrated in Figure 3B, while people closer to the centers of the distributions exhibited a mixed structure. Interestingly, subjects with large *Skewer* weights were nearly exclusive to Group 1, while those with large *Typechunk* weights were predominantly in Group 2. Thus, people trained on the same stimulus sets spontaneously discovered a variety of abstract drawing procedures.

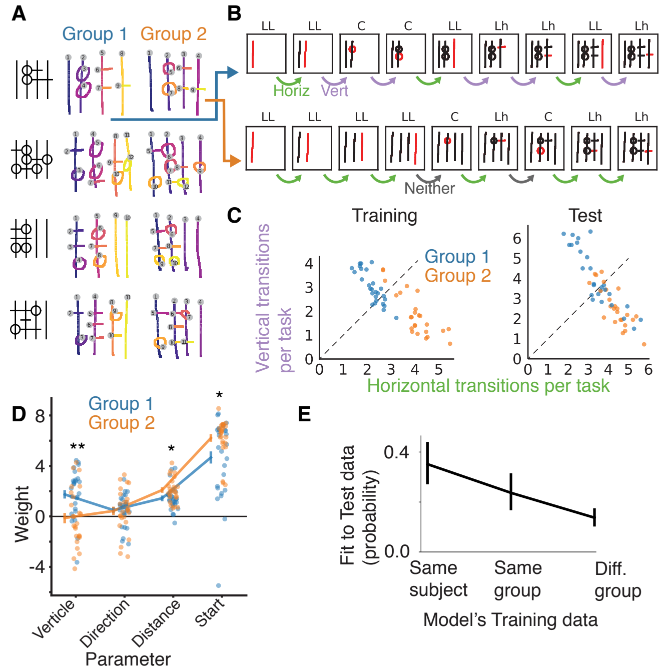

Figure 2: Single-session learning of structure in drawings. **(A)** Example drawing trajectories for two subjects (columns) on four test tasks (rows). Stroke order is indicated by both color (purple to yellow) and the number in grey dot. Grey dots also indicate start positions. **(B)** Example segmentations (LL, vertical line; C, circle; Lh, horizontal line). **(C)** Stroke transitions were directionally biased. **(D)** MC feature weights fit to behavior on Test stimuli. Positive corresponds to a bias for transitions that are vertical, towards bottom-left, low distance, and for first stroke at top-left. *, **, p<.05, .005, t-test. **(E)** Probability (mean, SE) of Test behavior given MC models either fit to Training behavior for the *same subject*, subjects from the *same group*, or *different group*.

Modeling drawing as program induction with efficiency constraints. These analyses suggested that subjects used abstract procedures for drawings, such as *connect a line to a circle and repeat it two times*. We trained a pair of Hybrid models on either Training sets 1 (HM1) or 2 (HM2), initialized with the primitives in Figure 4A (Table 1). Hybrid models combine learned abstract program structure, allowed to differ between groups, with population-averaged motor efficiency constraints. Through training, the model learned new primitives by abstracting out and storing for later reuse useful new primitives (examples in Figure 4B). After training, models performed well on the test tasks [mean/SD of 1.9/1.7 (HM1) and 0.35/0.76 (HM2) mistakes (missed or extra strokes) per task (out of 11.6)]. We compared model behavior to humans on test tasks and found that Human group 1 was better fit by HM1 than by HM2, and vice versa for Human group 2, indicating an overall good fit of the Hybrid model (Figure 4C, inset).

We compared the Hybrid model to alternative models (see Methods, Figure 4C). HM was better than Program Induc-

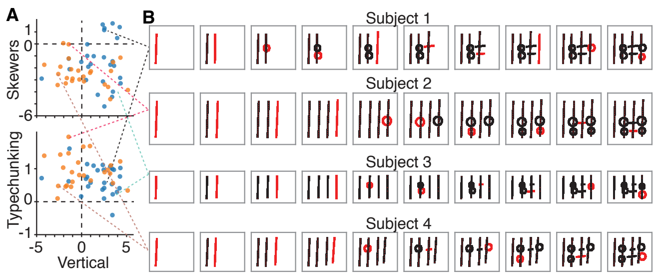

Figure 3: Program-like structure in behavior. **(A)** Subjects' weights for three different MC+ coefficients. Blue, group 1; orange, group 2. **(B)** Example program-like trajectories of four subjects with different behaviors. The dashed lines link these subjects to their weights in panel (A).

tion alone (PI), indicating an important role for motor biases. Moreover, HM performed better than the Motor Costs (MC) and Motor + Switching Costs (MC+) models. Thus the best model (HM) combined learning of abstract programs with low-level efficiency biases.

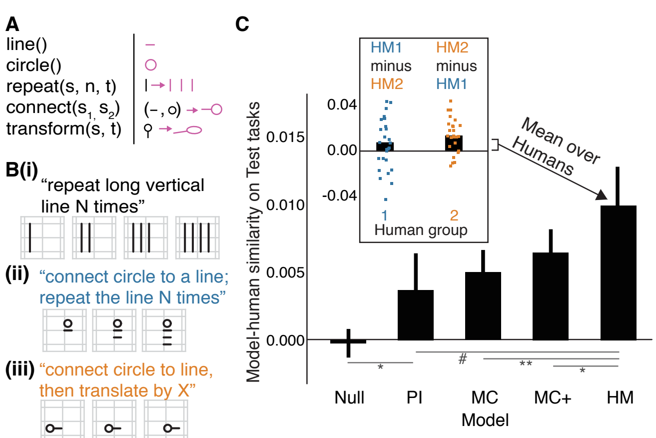

Figure 4: Modeling drawing as program induction with a bias for efficient actions. **(A)** Starting primitives (left) and example drawings (right). *s, n* and *t* are variables representing strokes, natural number, and transformation (see Methods). **(B)** Example primitives learned by both PI1 and PI2 (Bi), PI1 only (Bii), and PI2 only (Biii). **(C)**. Model scores in predicting human behavior on Test tasks. *, **, $p<.05$, $.005$; #, $p=.06$, paired t-test. Null model $p<.05$ vs. other models.

A follow-up experiment. If people indeed learned new drawing abstractions, as indicated by modeling results, then these abstractions should persist if the Test stimuli are rotated, as in Figure 5A. If, in contrast, subjects simply learn to increase the probability of vertical or horizontal transitions, then we would expect them to express that bias regardless of the stimulus orientation.

We found, in a followup experiment, that directional biases in behavior rotated if we rotated the test stimuli (Figure 5B,C). In the original experiment people in Group 1 exhibited a stronger vertical bias than those in Group 2 during Training; this effect carried over to Testing (Figure 5C). However, in the follow-up experiment, while Group 1 subjects still exhibited a stronger vertical bias during training, they preferred horizontal transitions for the rotated test stimuli (Figure 5C). This flexible adaptation of directional biases in a manner that mimics the orientation of the stimuli corroborates the modeling-based conclusion that people learned abstract drawing procedures.

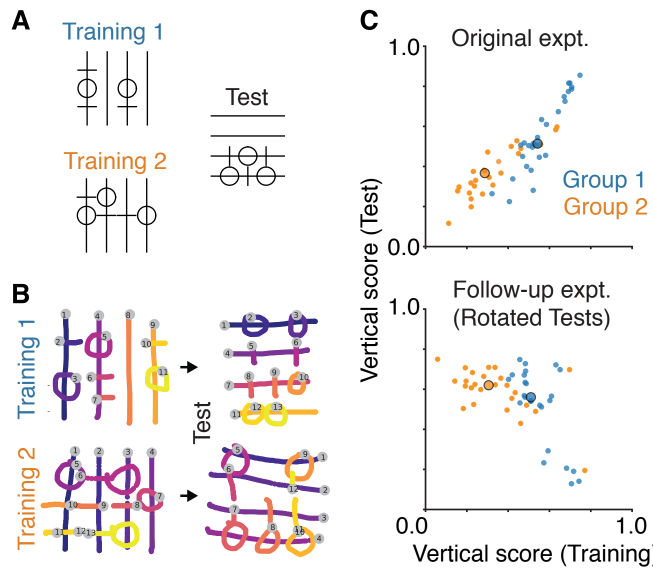

Figure 5: A follow-up experimental test for program-like structure. **(A)** Test stimuli are rotated relative to the original experiment in Figure 2 by 90°. **(B)** Example drawings for two subjects showing that they retained high-level structural biases evident in Training when performing rotated Test tasks. **(C)** Summary analysis. "Vertical score" is computed as V/(V + H), where V and H are average vertical and horizontal transitions per task. Large dots indicate medians. See text for description.

Discussion

Our experiments demonstrate that humans can acquire new concepts for drawing in a manner that is *abstract* (beyond low-level motor biases) and *unsupervised* (without any stroke-level instruction). We describe a modeling framework to capture this learning as the acquisition of new compositional abstract procedures for generating motor programs that underlie visual structure. Trained on the data given to human participants and without strong supervision, it flexibly learns new priors over programs by incorporating new symbolic abstractions into its language of thought.

Our modeling account is based on hierarchical Bayesian accounts of *learning-to-learn* (Kemp, Perfors, & Tenenbaum, 2007; Lake et al., 2015) and related arguments that learning involves a drive to find parsimonious model-based explanations of our experiences (Feldman, 2016). The speed with which both humans and models learned (with only 18 drawing experiences) - in contrast to deep networks, like (Ha &

Eck, 2017; Mellor et al., 2019), which require large datasets - highlights the importance of structured prior knowledge (or *inductive biases*) in the rapid learning of concepts.

Analogous to "chunking," (Diedrichsen & Kornysheva, 2015), an expanding inventory of primitives may support faster mental retrieval and manipulation, while minimizing effort related to working memory and switching costs. This chunking may be conscious or may be implicitly embedded into the neural systems supporting reasoning and learning.

While our study is focused on learning over short timescales, drawing is surely affected by a life-time of experience with visual symbols in systems of writing (e.g. alphabets or musical notation), art, and other real-world artifacts (e.g. "tomatoes and sausages on a grill", "dumbbells arranged on a rack"). The variation in drawing strategies not explained by our program-induction model is likely to partly arise from variation in prior experiences like these. Future work may assess people's prior knowledge through diagnostic baseline tasks, and study the interaction between people's existing concepts and online learning from experience.

Beyond prior experience with visual symbols, drawing depends on other diverse systems of biases and knowledge, such as musculo-skeletal constraints, efficiency biases, syntactic rules, world-knowledge, and geometry (Goodnow, 1977; Van Sommers, 1984; Karmiloff-Smith, 1990; Depeweg, Rothkopf, & Jäkel, 2018; Forbus et al., 2011). Future modeling may consider integrating these other systems of knowledge. We believe that these natural and creative aspects of drawing – all realized within a grounded sensorimotor space – make it an excellent domain for continued study. Insights will inform our understanding of how humans acquire, integrate, and deploy knowledge in languages for thinking.

References

Bartlett, F. C. (1932). *Remembering: A study in experimental and social psychology*. Cambridge University Press.

Cheng, P., McFadzean, J., & Copeland, L. (2001). Drawing on the temporal signature of induced perceptual chunks. In *Proceedings of the twenty-third annual meeting of the cognitive science society, edinburgh, uk* (p. 200).

Cheyette, S., & Piantadosi, S. (2017). Knowledge transfer in a probabilistic language of thought. In *Cogsci.*

Chi, M. T., Glaser, R., & Farr, M. J. (2014). *The nature of expertise*. Psychology Press.

Depeweg, S., Rothkopf, C. A., & Jäkel, F. (2018). Solving bongard problems with a visual language and pragmatic reasoning. *arXiv preprint arXiv:1804.04452*.

Diedrichsen, J., & Kornysheva, K. (2015). Motor skill learning between selection and execution. *Trends in cognitive sciences*, *19*(4), 227–233.

Ellis, K., Morales, L., Sablé-Meyer, M., Solar-Lezama, A., & Tenenbaum, J. (2018). Library learning for neurally-guided bayesian program induction. In *Neurips.*

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), e1004610.

Fan, J. E., Yamins, D. L., & Turk-Browne, N. B. (2018). Common object representations for visual production and recognition. *Cognitive science*, *42*(8), 2670–2698.

Feldman, J. (2016). The simplicity principle in perception and cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, *7*(5), 330–340.

Fodor, J. A. (1975). *The language of thought* (Vol. 5). Harvard university press.

Forbus, K., Usher, J., Lovett, A., Lockwood, K., & Wetzel, J. (2011). Cogsketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science*, *3*(4), 648–666.

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.

Goodnow, J. J. (1977). *Children drawing*. Harvard University Press.

Gureckis, T. M., Martin, J., McDonnell, J., Rich, A. S., Markant, D., Coenen, A., ... Chan, P. (2016). psiturk: An open-source framework for conducting replicable behavioral experiments online. *Behavior research methods*, *48*(3), 829–842.

Ha, D., & Eck, D. (2017). A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*.

Harlow, H. F. (1949). The formation of learning sets. *Psychological review*, *56*(1), 51.

Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science*, *268*(5214), 1158–1161.

Jaekel, F., Savova, V., & Tenenbaum, J. (2009). Grammar-based object representations in a scene parsing task. In *Cogsci.*

Karmiloff-Smith, A. (1990). Constraints on representational change: Evidence from children's drawing. *Cognition*, *34*(1), 57–83.

Kemp, C., Perfors, A., & Tenenbaum, J. B. (2007). Learning overhypotheses with hierarchical bayesian models. *Developmental science*, *10*(3), 307–321.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332–1338.

Mellor, J. F., Park, E., Ganin, Y., Babuschkin, I., Kulkarni, T., Rosenbaum, D., ... Eslami, S. (2019). Unsupervised doodling and painting with improved spiral. *arXiv preprint arXiv:1910.01007*.

Rule, J., Schulz, E., Piantadosi, S. T., & Tenenbaum, J. B. (2018). Learning list concepts through program induction. *BioRxiv*, 321505.

Van Sommers, P. (1984). *Drawing and cognition: Descriptive and experimental studies of graphic production processes.* Cambridge University Press.