An experimental and theoretical tool for studying the language of geometric concepts

by Manuj Dhariwal

B.Des, Indian Institute of Technology Guwahati (2008)

Submitted to the Integrated Design & Management Program
and Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering and Management and
Master of Science in Electrical Engineering and Computer Science
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2018

Signature redacted

Author . . . .

Department of Electrical Engineering and Computer Science
Integrated Design & Management Program
June 11, 2018

Signature redacted

Certified by . . . .

Laura Schulz
Professor of Cognitive Science
Thesis Supervisor

Signature redacted

Certified by . . . .

Joshua Tenenbaum
Professor of Computational Cognitive Science
Thesis Supervisor

Signature redacted

Certified by . . . .

Suvrit Sra
Assistant Professor of Electrical Engineering and Computer Science
Thesis Reader

Signature redacted

Accepted by . . . .

Matthew Kressy
Senior Lecturer
Executive Director, Integrated Design & Management Program

Signature redacted

Accepted by . . . .

Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

An experimental and theoretical tool for studying the language of geometric concepts

by Manuj Dhariwal

Submitted to the Integrated Design & Management Program and Department of Electrical Engineering and Computer Science on June 11, 2018 in Partial Fulfilment of the Requirements for the Degree of Master of Science in Engineering and Management and the Degree of Master of Science in Electrical Engineering and Computer Science

Abstract

In this thesis, I propose concretizing the Piagetian view of children as 'gifted learners' to children as 'gifted language builders', who construct and learn many languages to reduce their uncertainty about the world. These include languages such as, the language of geometry, the language of music & rhythm, even a child playing with blocks (eg: LEGO) is actually learning or rather building a language for themselves. As a specific case, I introduce an experimental paradigm and tool, *Finding GoDot*, for studying the cognitive language of geometry. Using the above lens, I model constructive actions as a language, specifically looking at the task of drawing shapes.

Next, majority of this thesis deals with the problem of calculating the entropy and redundancy of such a language for which there is no readily available language data. For this, I utilize Shannon's insight of accessing our implicit statistical knowledge of the structure of a language by converting it to a reduced text form, through a prediction experiment. I generalize Shannon's experiment design to make it applicable for a wide variety of languages, beyond just text-based, especially those lacking existing language data.

Finally, I compute entropy (average information per letter) values for individual shapes to show evidence of subjects using a rich forward model to mentally simulate incomplete shapes, thus gaining information about the underlying shape more than is visible. I also share results on bounds for the entropy and redundancy of the proposed language of actions for generating shape drawings.

Thesis Supervisor: Joshua Tenenbaum
Title: Professor of Computational Cognitive Science

Thesis Supervisor: Laura Schulz
Title: Professor of Cognitive Science

Acknowledgements

First to Saraswati Maa — for giving me faith in my faith that everything happens for the good; for time and again so clearly proving her constant presence; for keeping her promise of giving me the best possible education by being in my life in the form of my Mummy and now also my partner Gama.

@To family & friends back home

To my Mumma, who is for me a real incarnation of every possible motherly form of God in one person, who sacrificed her career as a top of her class gynecologist and made it possible for a dreamy hyperactive child like me to get interested in Math and Science and clear IITJEE, literally piggybacking me and running most of my major races in life, and giving me all the confidence & credit for winning them; for giving away so much of her life to make sure I start doing better things with mine. I would not have been anything without her both literally and figuratively!

To Gama, who has been my constant sound board, my unofficial thesis advisor, and my bestest friend whose sacrifice, love and support reflects in every page of this thesis and every course I took at MIT. Thank you for ensuring that I be good and do good in my life!

To the hardest working person, my Papa, for always taking care of a careless child like me who has a tendency of having some or the other minor illness. He is undoubtedly the best pediatrician in the world! Thank you for standing behind me so I could let go of my scholarship from Singapore and stay at MIT instead to do a SM in Computer Science. It is because of your support, that I could be so carefree and work on things that interest me the most!

To both my amazing star brothers, Jay and Rajat, one for being like a parent, for teaching me how to play Contra ☺ and also Linear Regression; and the other for being my closest friend, for getting me!, for so patiently teaching me $(a\text{-}b)^2$, for helping me finish my undergrad!, for..... To both for being so patient with me and for spending countless hours clearing my 10001 doubts! To Payal bhabhi & Chabi for making me feel that I have two real sisters!

To Rapa and Tamma for all their support in helping us get settled, for helping me recover and for always being there! To both Devesh & Arun Jijaji for being so inspiring! And of course to Ruchi Didi, without her this journey would not even have started!!!
To all my friends from DOD, I have the most fun and I am the most relaxed when I am around you guys! (Kissi-Red-B-BB)
@To my teachers and mentors
To Prof. Schulz for being so approachable, caring, tremendously encouraging and most importantly, being patient with me and for playing a part in helping me start my PhD program.
To Prof. Tenenbaum for introducing me to the hypothesis space; for being an inspiration & for being an awesome combination of brilliant ☺ and nice; and for teaching me the meaning of and an appreciation for the word 'Science'.
To Prof. Wornell whose class introduced me to the simplex and most most importantly to the thoughts and ideas of Claude Shannon - I willfully spent over a month reading and thinking through his papers. If I could click only one picture at MIT, it would be with his statue at LIDS.
To Parth (the enthusiastic 5-year-old), for being restless and bored and goading me to make a game for him - which kickstarted this thesis!
To Prof. Kressy, for teaching me both by example and in his unique way, the importance of exercising. I would be lucky if I could be half as fit as him (touchwood!). And of course, for so kindly granting me an extension, so I could finish this thesis with sanity and satisfaction. To Prof. Eppinger, for promptly helping me by giving me access to the conference rooms at Sloan.
To Prof. Belmonte, for introducing me to the thought of studying at a school like MIT, for supporting all my graduate school applications through the years, and for being there for me.

And of course, to Swami Sivananda, Swami Krishnananda for their teaching of "Matter doesn't matter". And finally to my living legend, most wise, most brilliant and always cheerful!!! → Guruji Jain Maharaj, who by having a 100% prediction accuracy about every future incidence of my life, is the secret knower of the Master Algorithm - that everyone these days seems to be after! :)
Thank you so much everyone for your patience, time, and for making me thrive, survive, and for letting me keep playing!

Content

0 Key ideas and their flow

The current version of this thesis lives at:

manujdhariwal.github.io/SMThesis

In this thesis, I propose looking at a lot of different kinds of human learning as form of language learning. The flow of thoughts and ideas in this thesis are as follows:

- One of the key hypothesis of Piaget was seeing children as gifted learners, building their own intellectual structures.

- Learning in general can be seen as reducing uncertainty.

- Shannon gave us a method to objectively measure uncertainty through his notions of Entropy and Redundancy.

- Looking at the results from the experiments that I did with both young children and adults as shared in this thesis, I propose looking at a lot of different kinds of human learning through the lens of language learning. For instance, using this lens, one can view a child playing with LEGO blocks as → learning a language. To tackle the task of learning to reduce their uncertainty about the world, children cognitively construct their own languages, identifying and creating both the alphabet for a language and iteratively building its probabilistic grammars. The alphabet of these various languages can be composed of not just typical letters, but also actions, sounds, and various other sets of building blocks. As a specific case in this thesis, I introduce an experimental paradigm, *Finding GoDot*, for studying one of these languages → the cognitive language of geometric concepts.

- Here I specifically tackle the task of approximating the entropy and redundancy that this language might have. It is a non-trivial problem to define and verify the specifics of our cognitive language of geometry. So here I create and propose a possible sub-language, 'Sketch-O' (as a language of actions to generate shape drawings) and argue why it might be more apt than other sub-language possibilities.

- One of Shannon's key insights was about translating the English language into a reduced text form through his prediction experiment and using that to calculate the bounds on the entropy and redundancy of English. Although we can directly calculate these values for a language like English with its ton of readily available language data, I note that the real value of Shannon's experiment is for languages for which there is no such readily available data or for which the only source for this kind of language statistics is our own cognitive machinery! And the first step to access this is to have a broader view of a lot of human learning as being a kind of language learning. Next, to be able to extract the statistics for these languages, I generalize Shannon's experimental method to be applicable to a wide variety of other languages. As a specific test case, I use it for calculating the entropy and redundancy of a universal language of actions for generating shape drawings.

- Lastly, I use entropy values for individual shapes, to show evidence of participants using a rich forward model to mentally simulate incomplete shapes, thus gaining information about the underlying shape more than is visible. I further prove it by showing that subjects were not able to mentally simulate random non-sensical shapes and thus limit their information to what is visible.

- As my next steps, I briefly argue why a global prediction experiment, as proposed by Shannon, and extended in this thesis, is a stronger indicator of one's knowledge of a language than the Turing test which relies on testing a learner (a language model ~ AI system) by evaluating the instances created by them, using the alphabet of that language.

Children as gifted learners

Children as gifted 'language' learners

Children as gifted 'language builders'

1 Testing a game with Shannon, Piaget, and a 5-year-old

1.1 Piaget++

I believe we build/learn a hundred and one languages during our childhood, and the one we use for reading and writing is just one of them. These include languages such as, the language of geometry or the language of forms, the language of music and rhythm, even a child playing with blocks (eg: LEGO) is learning a language or rather building a language for themselves. In fact, a child taking in the myriad forms of inputs in the form of visuals, sounds, words, objects and their forms, colors, textures, faces, other agents and their goals and behaviors etc. is constantly building up many languages and sub-languages to make sense of it all. The activity of constructing a language can be thought of as both - identifying the building blocks of that language and the inductive constraints that govern the composition of those building blocks i.e. the grammar of the language. I would go as far as to claim that the insightful hypothesis made by Piaget, Papert, and others about → viewing *children as gifted learners*, can be *equated* to viewing *children as gifted language builders!* The activity of building and learning languages for oneself is the ultimate hallmark of early childhood development.

1.2 Shannon not nonnahS

Claude Shannon in his seminal paper [3] (A mathematical theory of communication), introduced the notion of redundancy and his measure of uncertainty of an information source - 'Entropy'. The concept of entropy provides an objective measure to study the statistical structure of a language. A simple way to get a picture about these notions is:

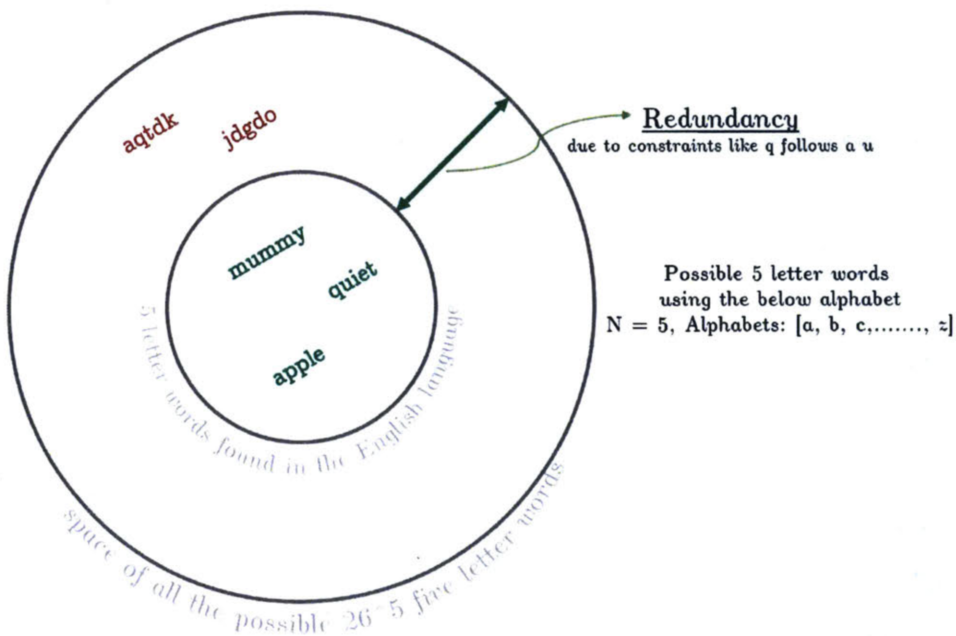


Figure 1 Visual representation of the concept of Redundancy in the English Language

The bigger circle in Figure 1 denotes the space of all the possible 5 letter words that can be made using the alphabet of English. The smaller circle is the five letter words one would find in an English dictionary. The measure of this compression in the possible sequences in a language to the ones used by the learners of a language is the Redundancy of the language. It is caused by the constraints placed on the alphabet of the language due to its grammar.

The Entropy of a language can be thought of as the optimal number of questions one needs to ask an expert in that language, to learn/know the sequence they have in mind. For a detailed outlook on these topics, please refer to the chapter on Entropy and Redundancy.

1.3 The Child

It is astonishing how a child takes in all the babble i.e. the sounds and words being generated in their environment, and slowly build for themselves the underlying structure of the language, its possible alphabet and probabilistic grammar, and thus start to speak approximate words, then words, and soon full meaningful sentences by the early age of 2-2.5 years.

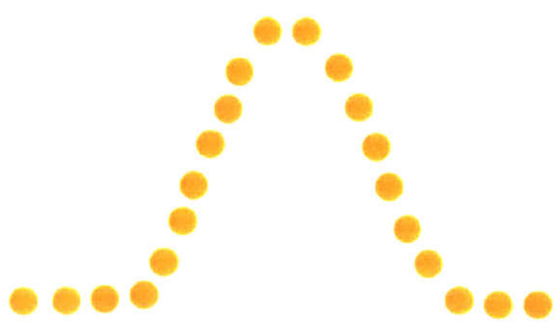
The question that I was curious about was $\rightarrow$ what kind of language is a child building from all the visual input they take in constantly day in and out — input in the form of shapes, color, textures, patterns of objects and tangible material around them both living, non-living, natural and man-made; what might be the possible alphabet sets and the rules/grammar they abstract in the process of building their cognitive language of geometry, as they see (and/or feel for children without eyesight) thousands of material instances every single day.

I was lucky to have an entire afternoon to spend with a five-year-old, and I started by asking him to draw instances of common shapes and objects, observing both his stroke order and abstract properties in his instances.

Having drawn 7-10 shapes for me, I saw him getting bored of the activity. Given my experience of previously building dozens of board games and card games for children, I flipped the activity on its head, and made a game out of it. This is the base mechanic for the experimental paradigm that I share in this thesis.

1.4 The Game

I showed the child a grid made of over 100 removable blocks and told him: "I have hidden a shape underneath these blocks, it could be any shape in the world, you have to find the shape by removing the least number of blocks!".



I had hidden a gaussian curve, made from N=21 dots. I discretized the shape into dots as otherwise, finding a continuous shape (given its starting point) would lead to no errors, zero entropy → no fun! both as a game and as an experiment.

The way the five-year-old went about this search task was quite intriguing. Below are some snapshots of him uncovering the gaussian shape (from Feb 2017):

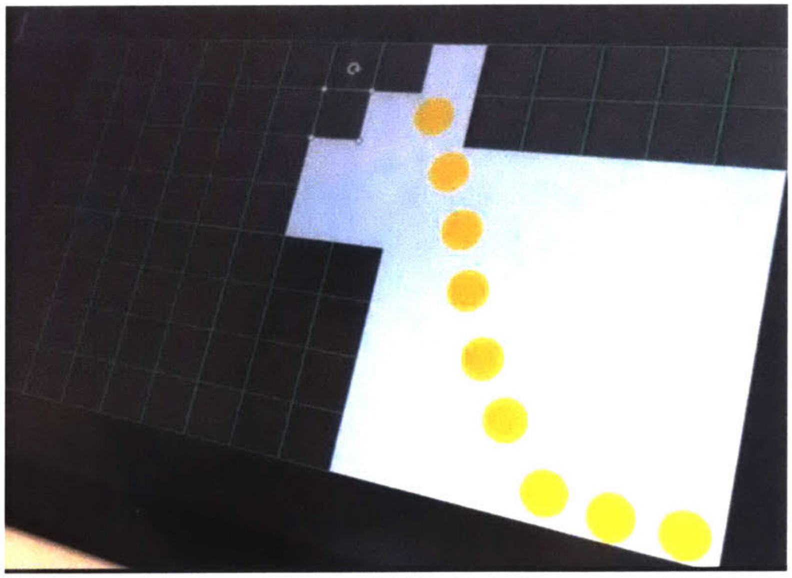


Figure 2 Makes his 1st prediction – "I think you hid a 'L'"

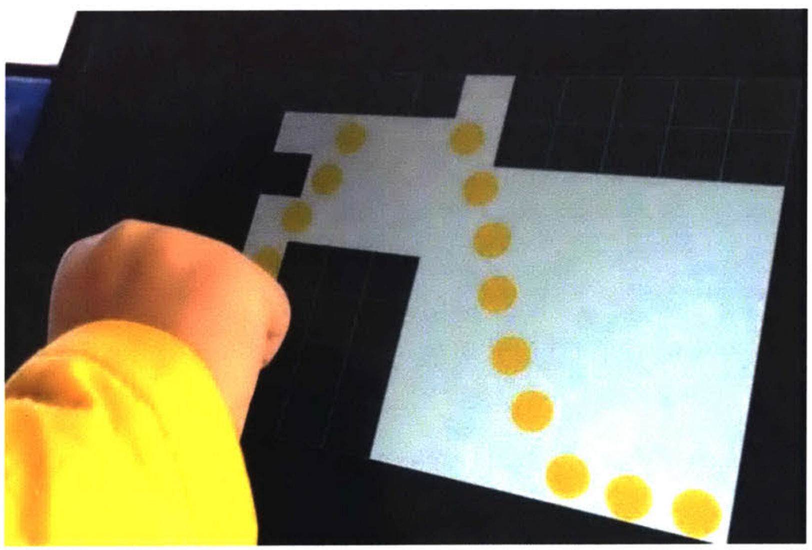


Figure 3 He seems to know that most shapes around him are symmetrical, and uses that to reduce his uncertainty

He started the game by randomly clicking around, until he fell upon the 1st yellow dot. Then, his search area became much more concentrated around that dot (he expects shapes to be continuous entities). Then he finds the next dot and continues looking for others along that direction (direction of momentum). After uncovering 5-7 dots he starts making predictions about what the shape might be – "you hid a 'L'". By the time he uncovers ~half of the shape (although he does not know the number of dots in the shape beforehand) something quite surprising happens → he knows most shapes are symmetrical! and starts to click at symmetrically opposite blocks along the mirrored curve of the right half of the shape.

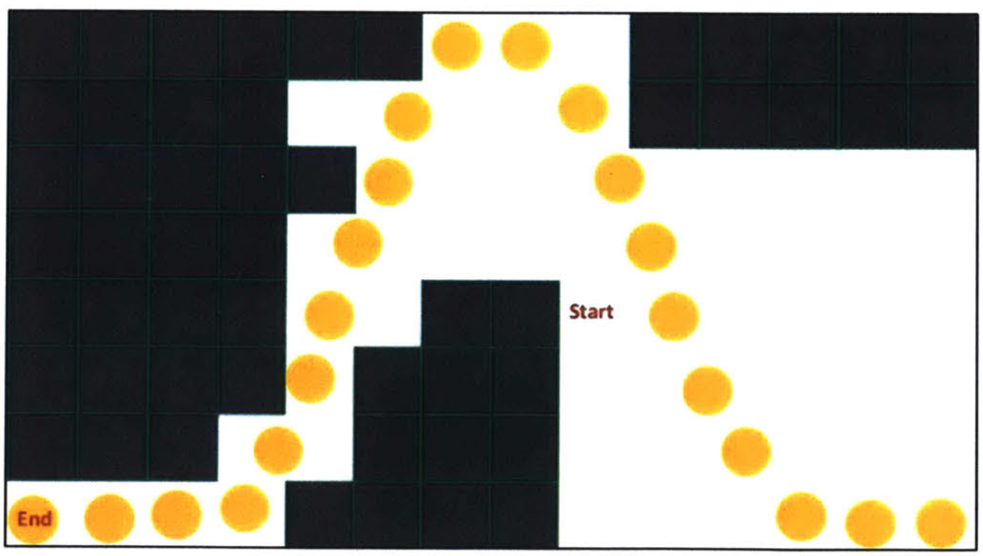
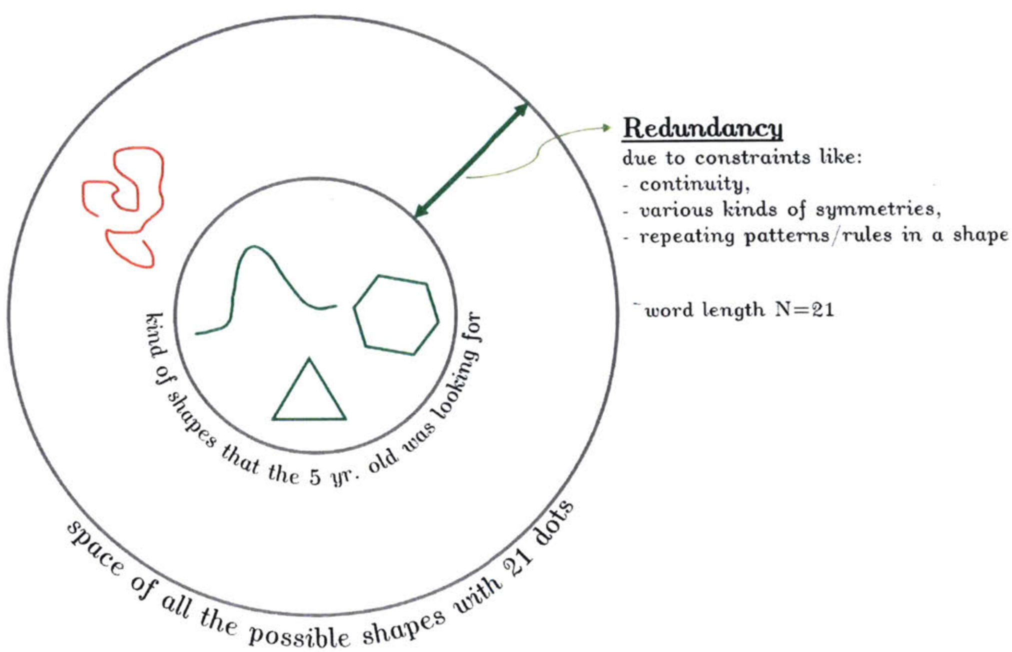


Figure 4 His final result→ 21 Dots hidden amongst 112 blocks, he removed 60 blocks to reveal the entire shape.

Starting from the right half and ending at the left most bottom corner, we can see how he becomes increasingly sure about the next dot in the shape and finds the last 7 dots without making any errors!

1.5 Children as gifted 'language' builders!

We could now essentially apply the same lens of 'Redundancy in a language' as discussed in section 1.2 and think of the child doing the above experiment as using their cognitive language of geometry to make predictions about the next dot conditional on the earlier ones already discovered.



So, of all the possible millions of shapes made of N dots (N=21 here), we see the child searching for regular shapes as shown in the smaller circle above. This compression of his search space, or the redundancy in the language, is caused by the probabilistic rules he has built for his language of geometry. This grammar includes properties like the various kinds of symmetries, continuity of shapes, expecting regularity or repeating patterns in shapes etc.

To summarize, Piaget and Papert gave us the notion of children as gifted learners building their own intellectual structures. Shannon provided us with an objective way to quantify uncertainty of an information source and use his measure to capture the statistical structure in the English language (1951). And lastly, the child playing the game - showed us how he used these built/learnt cognitive structures to significantly lower his uncertainty in searching for a hidden shape out of million possibilities. So, we can think of learning as a way of reducing uncertainty. Specifically, I propose looking at this learning from the lens of language learning - a child constantly building and learning a wide variety of languages to make sense of the seemingly disparate and myriad input streaming in through their senses.

2 Building a Language

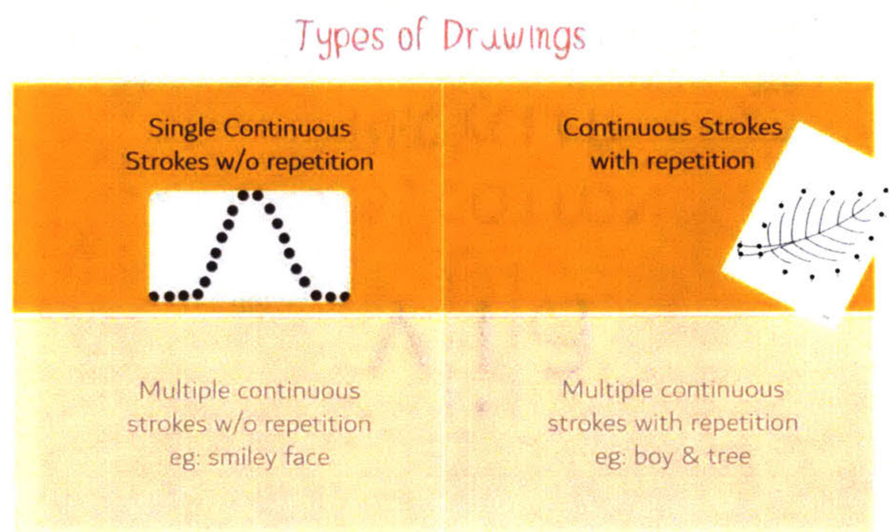
2.1 Internal representation vs external expression of a language

It is possible that the way we represent and express a language externally, is not its representation internally in our mind. If we think of the purpose of a language to be communication, then a language requires a tool or a medium for us to communicate with. For example, we use our mouth to communicate using a spoken language like English. Similarly, we use our hands to draw using our internal language of geometry or language of forms. The learnt structures lie within - the language is not in the hand. But if one was to only look at the sequence of actions as taken by the hand as it draws a shape, that sequence undoubtedly has a lot of structure in it. So, we can say the hand's actions are an outer representation of our cognitive 'language of geometry'.

By defining a language, we can represent an information source (in this case →the person drawing the shape) as a statistical process that generates the sequences to draw shapes or that correspond to the drawn shape. Now, one can create many possible sub-languages that can express (generate instances) like a human, thus it is advisable to choose primitives for such a sub-language that are both amenable to analysis and make common sense, as ultimately, we want it to be a close replica of an actual cognitive language that a human is using.

I present below two sub-languages for generating shapes and my reasoning of why one is better suited than the other. But let us first categorize the type of shape drawings one comes across in a typical sketchbook:

1. Single Continuous Strokes w/o repetition
2. Continuous Strokes with repetition
3. Multiple continuous strokes w/o repetition
4. Multiple continuous strokes with repetition



Before coming up with a language, it is helpful to think of a simplistic process of drawing shapes as follows: Our cognitive language of geometry guides the movements of our pen-holding hand to draw shapes on the sketchbook. And, since we do not know the structure of this language of geometry, we are trying to approximate it by creating possible sub-languages.

2.2 D.O.G: The Dots on Grid Language

Sub-Language Option 1

Sketchbook based 'Dots on Grid' or D.O.G language:

In my experiment I used a 9X13 grid - made of 117 cells or blocks. We could make each of these 117 blocks as a letter of the D.O.G language as shown below:

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76	77	78
79	80	81	82	83	84	85	86	87	88	89	90	91
92	93	94	95	96	97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112	113	114	115	116	117

Sample Words using D.O.G Alphabets:

"Hexagon" is [112, 111, 110, 109, 108, 94, 81, 67, 54, 41, 29, 16, 4, 5, 6, 7, 8, 22, 35, 49, 62, 75, 87, 100],

"Butterfly" is [84, 71, 58, 45, 31, 17, 29, 28, 41, 54, 68, 81, 94, 95, 96, 98, 99, 100, 87, 74, 62, 49, 36, 35, 21, 33]

A hexagon of the given size is uniquely represented by the above tile pattern (not the actual line sketch of hexagon but a dotted version of it as used in the experiment).

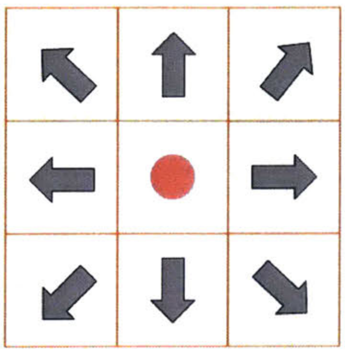
Although this language can represent any shape as drawn on the sketchbook, by specifying the location of ink on the sketchbook (one could theoretically use the specific {x, y} coordinates instead of approximate grid numbers, but that would lead to an impractically large alphabet set), but it is not an apt choice for a sub-language as it:

1. Fails Scramble Test: Even if the letters that make the Hexagon are scrambled, we would still get the same shape in the end, meaning it is read the same regardless of how it was written. This is contrary to how we draw shapes, where order of drawing a stroke is of importance.
2. Depends on grid size: The letters of this language are dependent on the grid size.
3. Fails Transform Test: Simple transforms on the shapes like moving them up, changes how the word is spelled even though it looks the same to a human observer.
4. Fails Scale Test: for same reasons as above

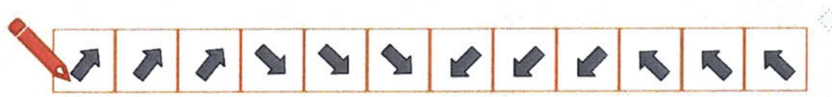
One benefit of using the above language is easy access to a lot of training data, as an existing shape image can easily be converted to this language, which could give us the probability distribution of the tiles in the language. But this would just be the monogram data as order is of no consequence in this language.

2.3 Sketch-O: Constructive actions as a language

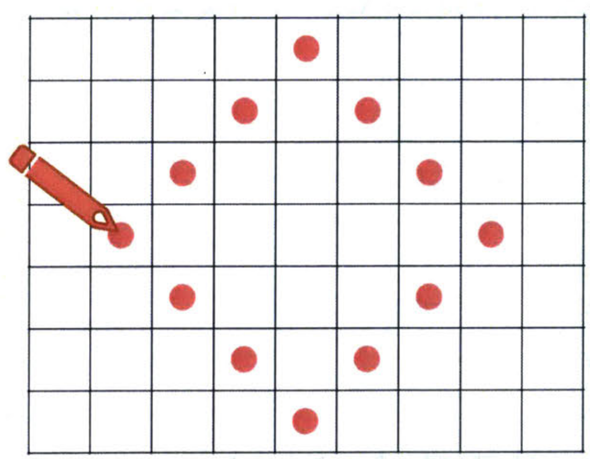
<u>Sub-language Option 2:</u> Sketch-O is a universal language of actions for drawing shapes, with the following alphabet:



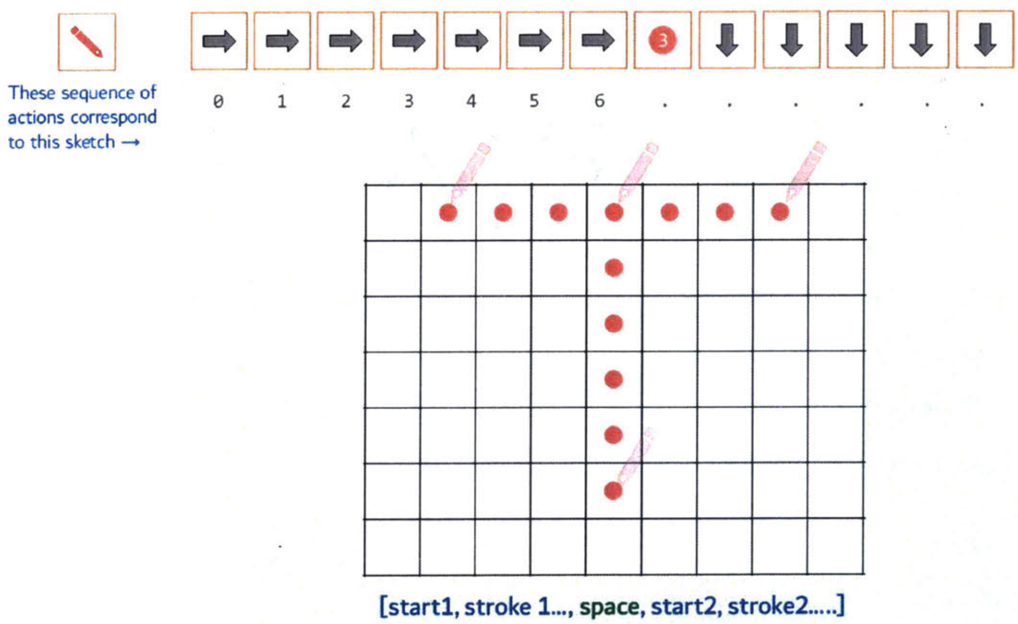
It is universal as regardless of culture, demographics etc. everyone draws shapes by holding a writing instrument and moving their hands. We can approximate a person's hand actions as they draw the shape as shown:



These sequence of actions correspond to the below sketch



The above is the case of a single continuous stroke without repetition. The above 8 arrow letter-set can generate any continuous stroke on our grid. To take care of shapes with multiple strokes, we add an added 'jump' symbol ('O') to our language, as shown below:



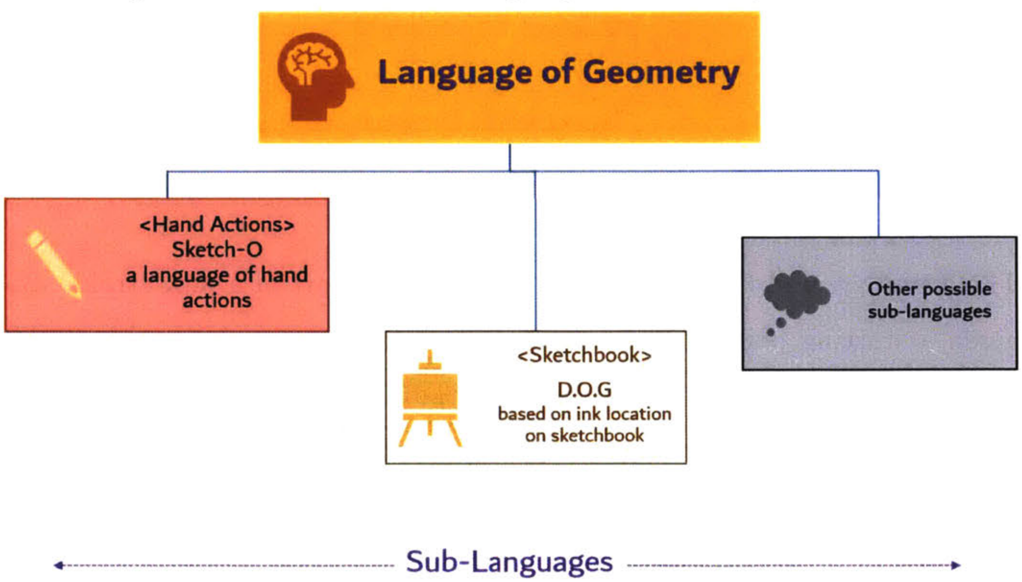
The Jump symbol ('O') can be thought of as a 'start of a new stroke'. So, given a starting point (cell 2 in the above grid), if one were making a 'T' shape, they would make a horizontal stroke as shown, and then press the Jump symbol and select a new starting point from amongst the existing array of points just drawn, and begin drawing the next stroke, going vertically downwards in the above case.

Sketch-O is a much better language than D.O.G:

1. Passes Scramble Test
2. Independent of Sketchbook size
3. Passes Transform Test
4. Passes Scale Test

The sequences generated by the language can be made compact by adding abstractions like Repeat (↗, n=5) would stand for ↗↗↗↗↗ which would translate to a line at a 45-degree angle on the sketchbook. Many such abstractions could be made like Rotate, Scale etc. typical of operations one comes across in vector graphic generation software programs.

Looking at one's own hand actions, rather than the sketchbook seems more in-sync with ones internal 'Language of Geometry'.



2.4 Role of cognitively rich & affectively diverse tools in helping children build languages

Children use their gift of constructing languages and constantly update their grammars to reduce their uncertainty about the world and make sense of the myriad inputs coming in from their senses. Thinking in this way, a child's brain can be thought of as a big language learning engine, and tools of all sorts play a significant role in helping them. For example, the mouth and the ears help to initialize, tune-up, and build their grammar engine for spoken languages. Similarly, children are not explicitly taught to play with LEGO blocks, but are just given as a material to play with. The blocks, the child's imagination, their hands, all act in tandem, like a tool as they go about building 2.5D/3D structures mimicking things around them or trying new experimental forms, thus helping solidify their visual language grammar engine. Similar is the role of tools like a sketchbook and a pen, clay, playdoh, and even digital tools like Scratch – a tool for children to tinker with code.

<u>A brief side note inspired from reading chapters from the book Mindstorms:</u>

I believe Seymour Papert's book *Mindstorms* [1] is word for word as relevant today as it was in 1980, with just an ever-expanding definition of what the word 'Computer' stands for — today, it being 'Computer + Sensors + AI/ML'. Building upon Papert's insight of using the affective to hone the cognitive, I wonder what form would have Turtle taken today, given the task of representing even richer computing models and paradigms?

As I read *Mindstorms*, one of the images that I got was of — kids living on the streets in urban cities, in close physical proximity with artifacts of modern science and technology (like cars, planes, and smartphones) but mentally, far divergent and alienated in their understanding of principles behind these. For me, the *'Poverty of Materials'* that Papert talked about is one of the important reasons behind what I think of as the real poverty, the *'Poverty of Models'* - models that we all assimilate and acquire rapidly in our early years. The richer the environment (filled with right, challenging, and fascinating tools & materials) that one grows in, the more sophisticated and diverse their toolkit of cognitive languages or models, which then act like compositional building blocks that lay the foundations of our future learning, development, and growth, ultimately affecting the very cultures and society that we grew up in.

3 Experiment Design Methodology & Software Tools

The question that I was curious about was → what kind of language is a child building to make sense of all the visual input streaming in through their eyes such as, the forms of thousands of objects, their shapes, color, patterns, etc. What might be the possible alphabet sets and the grammar they abstract in building the cognitive language of geometry.

3.1 Global search-based task vs analyzing instances

The problem with the usual approach of asking a person to create common instances from a language (drawing common shapes in this case) and analyzing those to study underlying aspects of the language, is:

Short Answer: Instances are heavily influenced by temporal and spatial context of the user and thus they are not necessarily representative of the underlying grammar or higher-level principles and knowledge of the language. Whereas for solving a global search-based task in the language, one is much more likely to use these higher-level strategies and abstract knowledge about the grammar of the language.

Long Answer: Our aim is to learn the grammar learnt by the child based on all the input data (visual input in terms of objects in the world and their shapes and form). Now let us compare these two tasks:

Task A: Create few instances from the language → draw 5-7 common shapes that you know of.

Task B: Find the shape that is hidden behind a set of blocks.

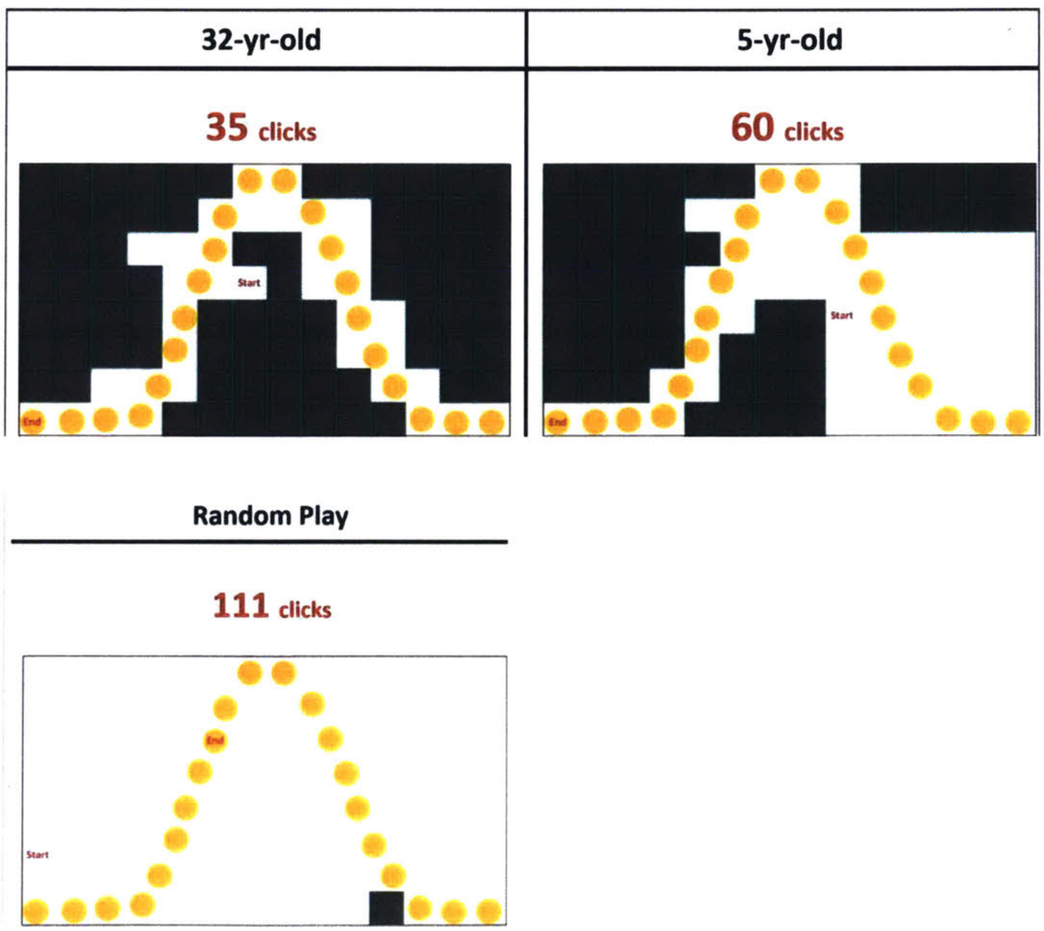
I argue that Task B is much more apt when it comes to finding higher level grammars and principles learnt by the learner of the language. Task A will be heavily dependent on the user's *spatial* and *temporal* context, i.e. the shapes they draw are highly likely to be influenced by objects in their surrounding or fresh in their memory, e.g. remembering the 'donut' they ate yesterday.
But for Task B - which is a task about guessing the underlying shape, when it could be any possible shape of a given length - the user's guesses will be informed by the underlying probability distributions (i.e. P(next dot|previous dots), that have been distilled from their input visual data over time.

3.2 General Experiment – 'Finding GoDot'

A 'Finding GoDot' experiment broadly consists of deducing the output of a generative process. So instead of asking a user to generate instances of a language for us (which are likely to suffer much more from spatial and temporal biases), we directly test their knowledge of learned inductive constraints for the language without introducing biases of any form.

And as I argue in this thesis – to look at a lot of human learning as a form of language learning, thus broadening our typical understanding of languages beyond the ones used for reading and writing, we can create versions of the above task for a wide *variety* of data and information sources. These can include constructive action sets of any kind (from drawing, to building LEGO structures, to dancing), to sounds (musical notes, song sequences, bird songs...) etc.

As a specific case, I implemented the above for studying aspects of the language of geometric concepts. The experiment as shared in section 1.4 consisted of asking a subject to uncover a hidden shape by removing the least number of blocks, one at a time. Below are results from a 5-year-old, a 32-year-old, and a random agent doing the same task of finding a hidden shape (~a gaussian in this case).



A test version of the experiment can be played at: https://manujdhariwal.github.io/Finding_GoDot_SM/

Figure 5 below shows the initial UI design for the experiment's WebApp.

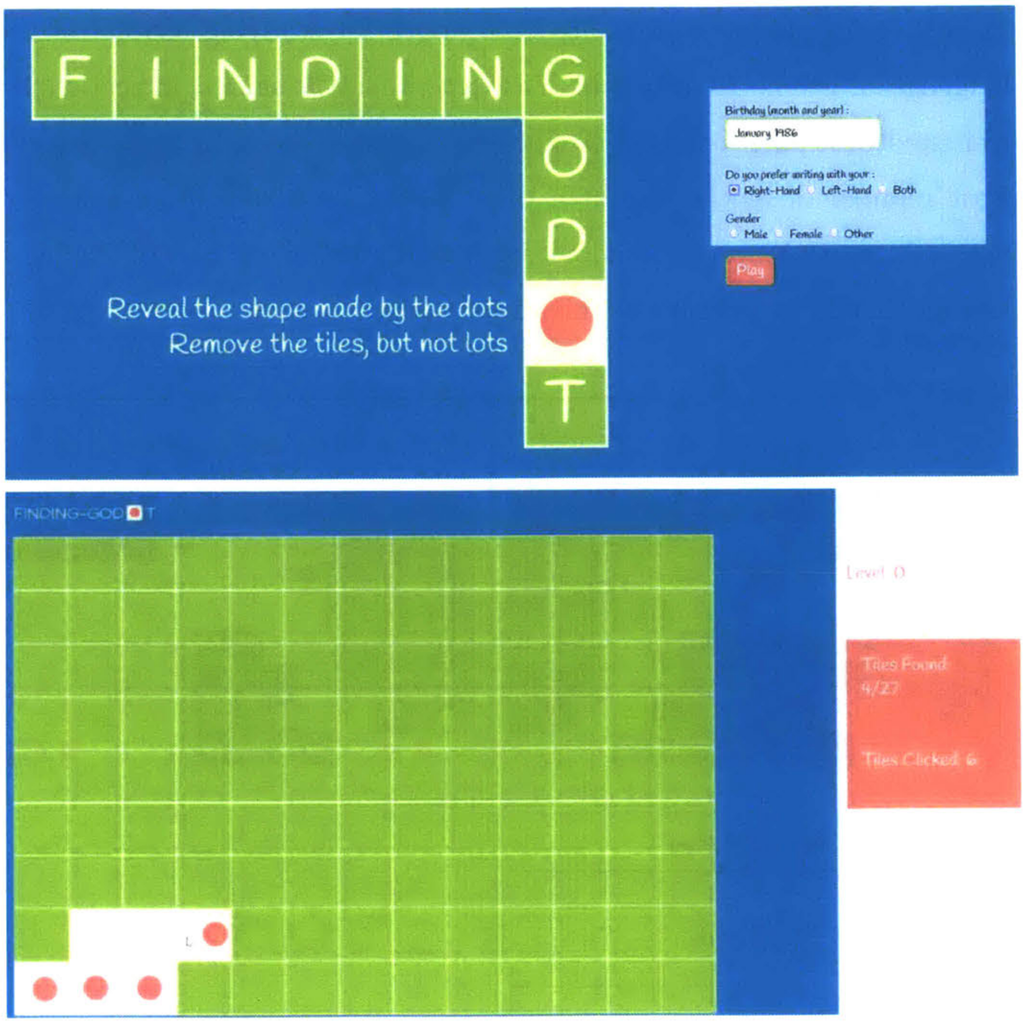
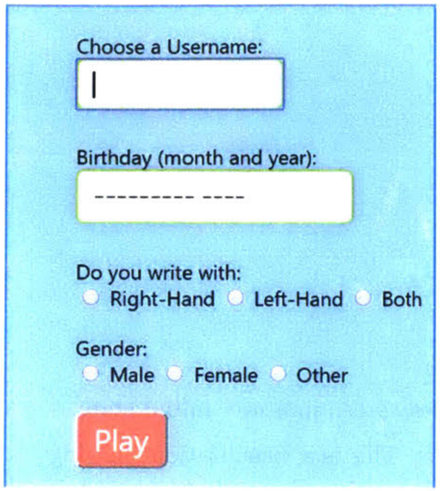


Figure 5 initial UI design



The users are asked to enter their birth month and year, so we can compare data across users on a more continuous scale.

They are also asked to enter their dominant hand information, as that might influence their search behavior, as a left-handed person is likely to draw certain shapes in a different stroke order than a right-handed person, and thus their expectation of the next hidden dot is likely to vary.

3.3 Observations from playtesting & design revisions

I tested the experiment's web app with both children (ages 2-7 years) and adults to test the design and verify if it served the goals behind the experiment. Below are some of my observations from the first playtesting sessions.

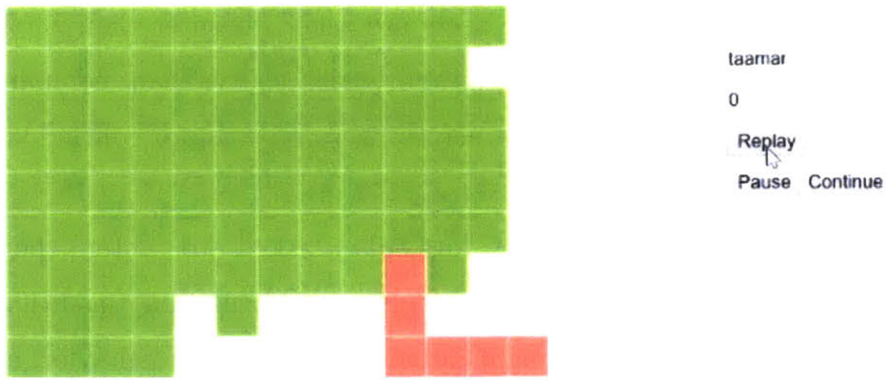


Figure 6 Game State: Pink tiles show correct tiles, Green indicates initial state of a tile and Cream tiles are empty or incorrect tiles. This is a visualization showing a 2 yr. old finding the hidden shape (~mountain)

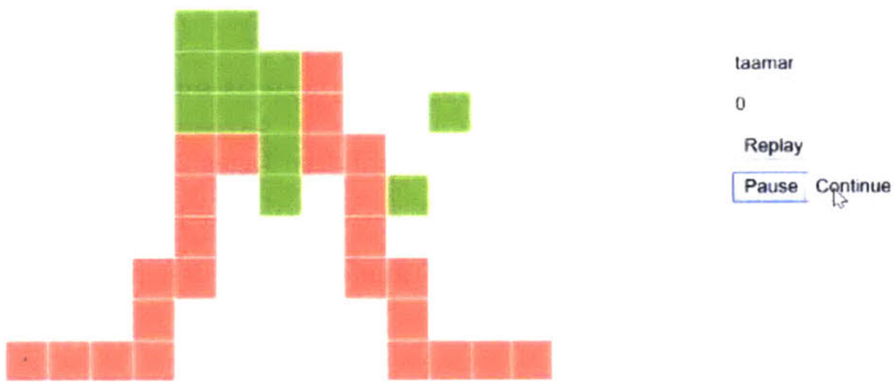


Figure 7 Endgame State: The 2 yr. old removes almost all the green tiles before finding the pink tiles.

Child Name: Gaga | Age: 2.3 yrs. | Accompanied with her mother
Observations: As it is hard to fully communicate to a 2-year-old what the objective/goal of the experiment/game is, they end up forming their own goals & reward functions out of the experiment.

Initially, Gaga (~2 yr. old) was unsure and took her time in clicking tiles to hide them. But as time progressed and she had removed quite a few tiles, she inferred the goal of the game as removing all the green tiles from the screen, and she was enjoying the sound that came when she removed a tile (whether correct or incorrect).

Her Inferred Simple Goal:

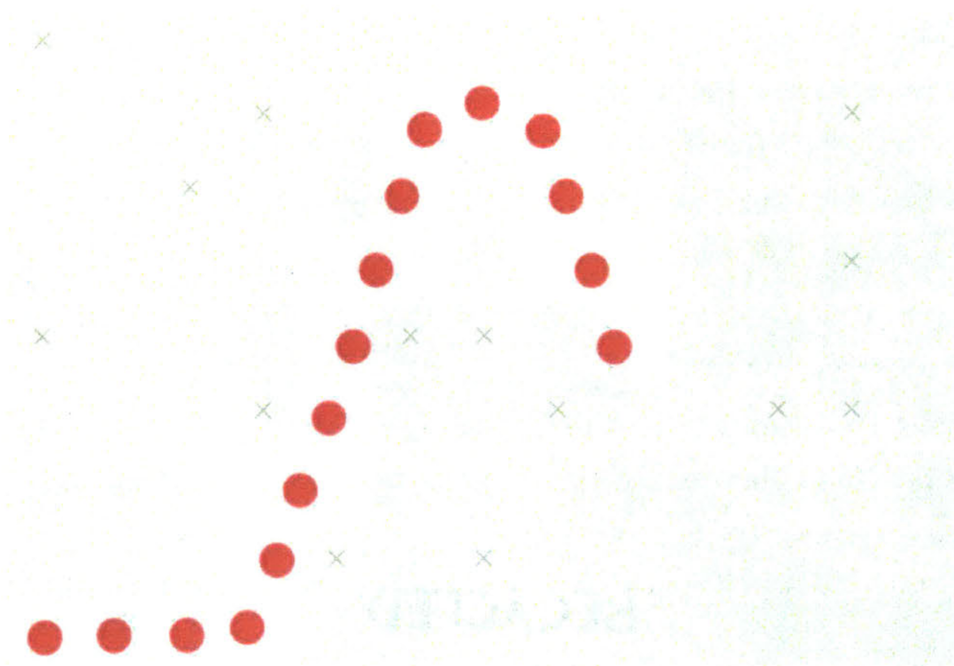
Remove all the green tiles off the screen!

Actual Goal: Uncover the pattern formed by the dots by removing the least number of tiles.

Her Reward Function: Sound of clicking the tile + Joy of clearing up green tiles

With other 2-3 yr. olds, I saw the same pattern of them liking the sound the game made when they clicked on an incorrect tile (tile which has no dots hidden behind it). Also, I was finding it hard to make sure if they understood that they needed to click the least number of tiles to uncover the hidden shape.

Based on the above, I changed the game UI as shown below, plus I changed the incorrect tile sound and added a subtle animation which both acted as negative feedback for them.

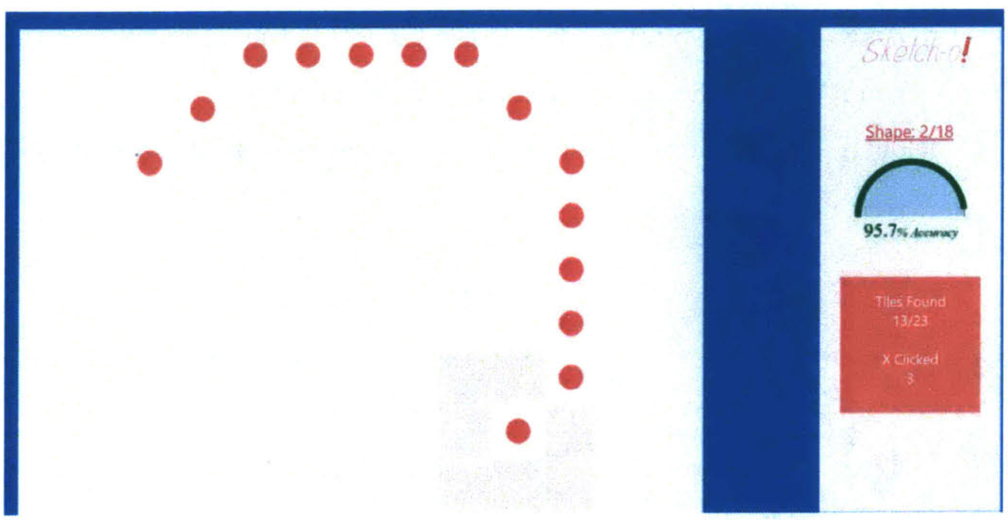


3.4 Setup: Prediction and entropy for the language of geometric concepts

As said before, it is a non-trivial problem to specify the building blocks of our cognitive language of geometry. But approximations can be made by proposing possible sub-languages that have a subset of the expressivity of the more complex cognitive counterpart.

Shannon in his paper "Prediction and entropy of printed English" from 1950, had proposed an interesting experimental method to calculate the bounds on entropy of the English language. Unlike his objective approach, the experimental method very cleverly taps into every English language speaker's enormous statistical knowledge about the structure of the language. Details on both his objective and experimental methods are shared in Chapter 4.

We can similarly tap into a person's statistical knowledge about the cognitive language of geometry and at least get rough bounds of its entropy and redundancy. The general experiment as proposed in this thesis, can be modified to a sequential version, where a user's search is constrained to find dots in the order in which they were drawn. The figure below shows this new setup.

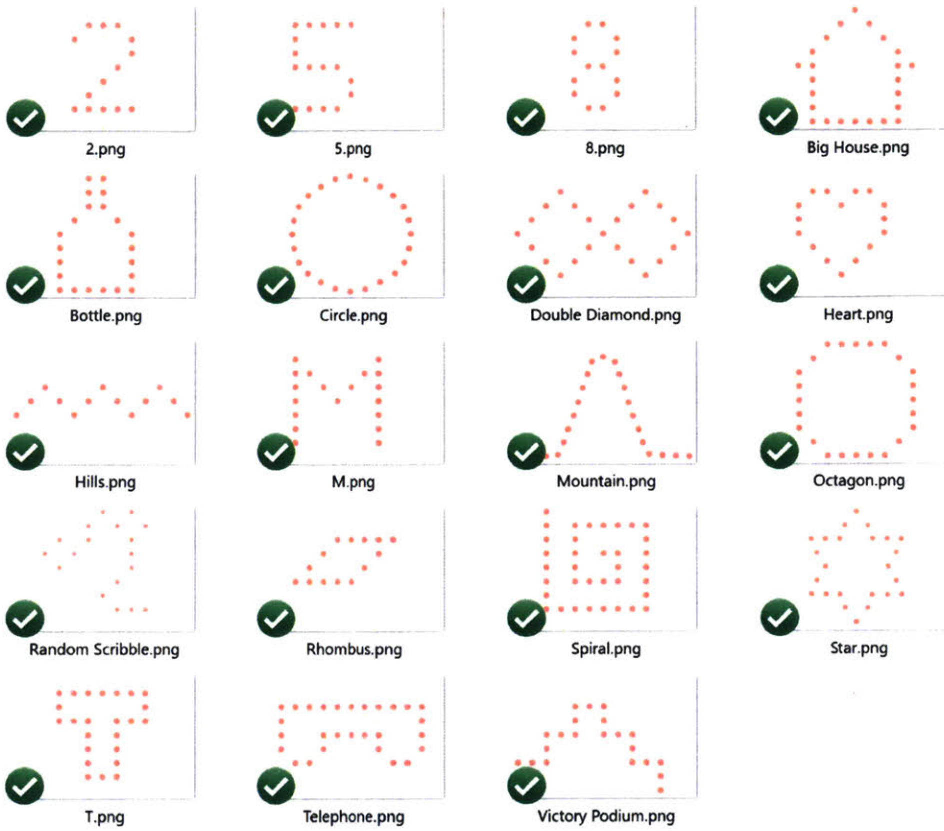


The entropy calculated using this experimental setup and Shannon's method will be a joint function of people's brains and the material (here shapes) they were shown. It is not an objective property of either people's brains or of the stimuli. But if we have a well-defined population of stimuli (shapes) than the entropy bounds calculated will also be well-defined and could stand as an approximation for people's language of geometry.

I used a set of 18 different shapes for the experiment, each of which were tested with over 60 participants (divided between mTurk and friends & family). The shapes were chosen based on the below three criteria:

1. Shapes with varied kinds of symmetries
2. Shapes that subjects are likely to have strong priors for, like numbers, letters 2,5,8, M, Z, R
3. And shapes with one or more repeating pattern/rule eg: spiral (to draw a square spiral we follow the repeating rule of increasing the count of dots by one and taking a 90-degree turn)

Below if a snapshot of the shapes used for the sequential experiment.



Based on the calculations (detailed out in the next chapter), shapes on an average have a high redundancy (roughly between 60%-80%).

For a detailed analysis of various kinds of symmetries (Mirror Symmetry; Rotational Symmetry & Translational Symmetry) in a shape and how to go about quantifying how symmetrical a shape is, please refer to [6], [7], [8] & Mach (1906/1959).

For a qualitative discussion on some informational aspects of Visual Perception, please refer to [5].

3.5 Extending Shannon's experiment to a wide variety of languages

One of Shannon's key insights was, about translating the English language into a reduced text form (details in Chapter 4), through his prediction experiment, and using that to calculate the bounds on the entropy and redundancy of English. Although these values can be directly calculated for a language like English with its ton of readily available language data, *I note, that the real value of Shannon's experiment is for languages for which there is no such readily available data or for which the only source for this kind of language statistics is our own cognitive machinery!*

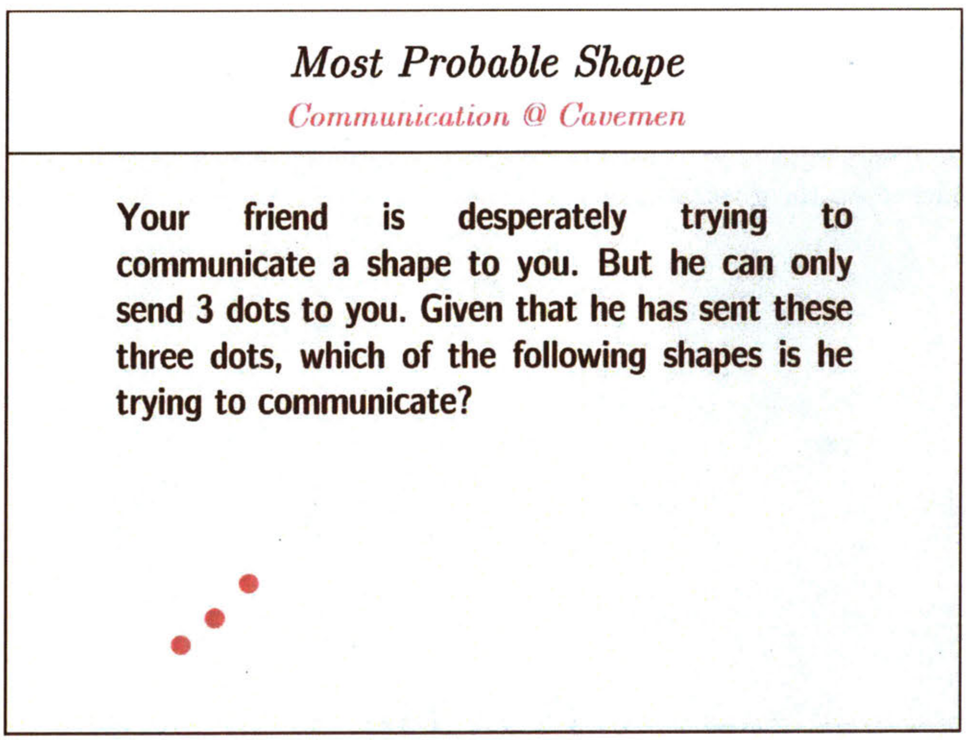
The first step to access this is to have a broader view of a lot of human learning as being a kind of language learning. Next, we can apply the 'guess the hidden instance' paradigm to that language's output. The above method is directly portable for languages whose outputs are inherently discrete like English text. But languages whose expressions have a continuous form (like drawings), must be cleverly discretized such that each discrete unit is self-contained and does not carry obvious information about other units. The sequential experiment implemented for the language of geometry is a good example of the same. Had the shape drawings not been discretized, the experiment would have fetched us no information, as the average information per dot (entropy) would have been zero!

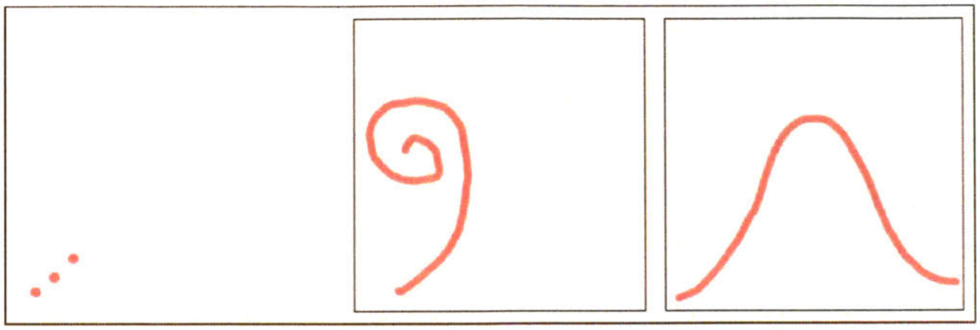
Using the above, one can generalize Shannon's method to calculate the entropy and redundancy bounds on many kinds of languages like language of music, or cases where one thinks of constructive actions as a language such as, LEGO blocks, Tetris, and others, even forms of dance and many more.

3.6 Communicating the Most Probable Shape

Another set of questions that can be answered using the Finding GoDot paradigm are of the form:

3.6.1 GoDot Cavemen Problem:





3.7 RePlay Tool: Visualizing user's data generation process

I created a simple tool that lets one watch the replay of a user's game session. Such dynamic visualization of the generation of a user's data gives one much better insights and leads to asking newer deeper questions from the underlying task.

This tool also visualizes the sequence of user's actions using the alphabet of Sketch-O language and compares them with the correct Sketch-O sequence for the given shape.

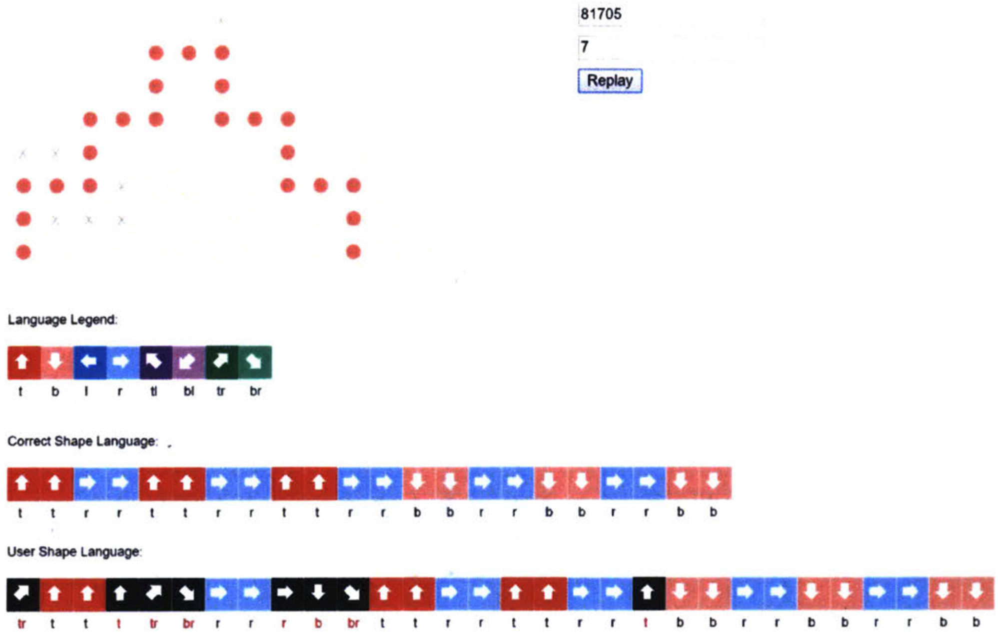


Figure 8 Sketch-O actions sequence data for a shape. Left-handed, 33 yr. old user.

Figure 9 Sketch-O actions sequence data on the same shape for a 2.5 yr. old

4 Calculating Entropy & Redundancy of a language

4.1 What exactly is 'Entropy' a measure of?

Now to use Shannon's method for calculating approximate complexity measures for our language of geometric concepts we designed one good possible sub-language Sketch-O. Using Sketch-O we could represent the information source, the person drawing the shape, as a statistical process that generates Sketch-O letter sequences to draw shapes.

The Entropy(H) then is:

$H = average\ information\ produced\ for\ each\ dot\ of\ the\ shape$

So, when we think of the person drawing a shape as an information source, who produces information at each step as he goes about drawing the shape (we can think in this manner for any generative process), then the entropy basically is a measure of average information that is produced for each dot.

Important Note & Clarifications:

I: It is *actually the average information produced for 'each letter' of our language of actions→ Sketch-O. The dot is merely an outer visible manifestation of our having taken the underlying action, from the 8 available actions that form the alphabet of our language.*

II: Here the word 'Information' does not stand as a measure of meaning or semantics being conveyed by the source. Please refer to section 4.3 for further clarifications on this.

H = *average information produced for each letter of our language shape*

H = *average information produced for each constructive action (from amongst the alphabet of our language Sketch-O) when drawing a shape.*

So, the same letter could give us different amounts of information depending on 'where' and in 'which' sequence it comes in at. This is what leads to structure in a language. And hence entropy is a good measure of the statistical structure in a language.

4.2 What is meant by a language being 'redundant'?

Now more the structure in a language, i.e. more the correlation between letters of a sequence (not just amongst adjacent ones but even long-range correlations), the more redundant that language is. Which is not always a bad thing! So, Redundancy is related to the extent to which it is possible to compress the language. Hence a random sequence of letters has zero redundancy, as each letter is independent of others.

4.3 Note on Kolmogorov Randomness

The central idea behind this measure of complexity is that a string of bits is random if and only if it is shorter than any computer program that can produce that string (Kolmogorov randomness)—this means that random strings are those that cannot be compressed.

Eg: The decimal digits of pi form an infinite sequence and never repeat in a cyclical fashion. From this point of view, a 3000-page encyclopedia has less information than 3000 pages of completely random letters, even though the encyclopedia is much more useful.

The information content or complexity of an object can be measured by the length of its shortest description. For instance, the string

"0101010101010101010101010101010101010101010101010101010101010101"

has the concise description "32 repetitions of '01'", while

"1100100001100001110111101110110011111010010000100101011110010110"

presumably has no simple description other than writing down the string itself.

4.4 Calculating Entropy & Redundancy

To calculate Entropy & Redundancy, we need lot of language statistics in forms of various probability distributions of letters, pairs of letters, words etc. of the language.

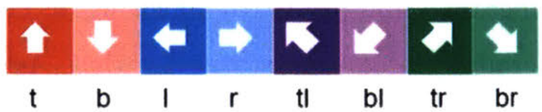
For details on calculating entropy from existing statistical data on the language, please refer [2] – Prediction and Entropy of Printed English by CE Shannon.

But as discussed in sections 3.4 & 3.5, Shannon devised his prediction experiment method to overcome this limitation, by devising a clever method to tap into the detailed language statistics that people accumulate (~in their brains) and refine, as their language skill increases over time.

The experiment translates a sequence written in Sketch-O (eg: a sequence that draws an Octagon) to the reduced text format. And by repeating the experiment with a good enough representative population of shapes, with many people, we can get frequencies for letters of the reduced text.
For example:
The figure below shows the data of a real user, trying to uncover the hidden shape (an Octagon here).

Now by using the following language legend for the eight Sketch-O actions:

Language Legend:



An Octagon, as shown above, is represented by the below sequence of actions:

And the figure below shows the sequence of actions as taken by the user as he uncovered the shape. The black arrows stand for – the errors in predicting the location of the next dot or the prediction errors made by the user, while guessing the next letter in the actual Octagon forming sequence above.

Side Note: Looking at this user's data, he has an extremely good understanding of shapes (~small no. of errors) – incidentally while talking with users after they had completed the game, this user turned out of be an experienced visual designer and painter!!

To summarize:

Sketch-O Sequence:

(A) 

Reduced Text Sequence:

(B) 4 2 1 2 1 1 2 1 1 1 1 1 2 1 1

From the data in the second line B, it is possible to set upper and lower bounds for the entropy of the language in (A). Line B can be thought of as a translation of line A. There is a theorem on stochastic processes that the redundancy of a translation of a language is identical with that of the original, if it is a reversible translating process going from the first to the second [4]. Consequently, an estimation of the redundancy of the line B gives an estimate of the redundancy of the original language. Line B is much easier to estimate than line A since the probabilities are more concentrated. The symbol 1 has an extremely high probability, and the symbols 2 to 8 have subsequently smaller probabilities.

Shannon had proved that if the probability of taking r guesses until the next letter in the correct sequence (eg: sequence A above) is guessed is P_r, then the entropy, H (in bits per letter) is:

$$\sum_r r(P_r - P_{r+1}) * \log_2 r \le H \le \sum_r P_r \log_2 \frac{1}{P_r} \qquad (1)$$

$where\ for\ SketchO\ alphabet\ of\ 8\ letters, \quad r : 1 \le r \le 8$

For details please refer to equation 17 in [2].

From the above equation (1), one can get the bounds on the Redundancy of the language as follows:

$$1 - \frac{H_{Upper}}{H_{Max}} \leq \; R \leq 1 \; - \frac{H_{Lower}}{H_{Max}} \qquad (2)$$

$, where:$
$H_{Max} = \log_2 8 \; = 3$
$and \; H_{Upper} \; and \; H_{Lower} \;\; are \; from \; equation \; (1)$

Using the experiment setup as discussed in Section 3.4, participants were required to guess the various shapes, dot by dot (in the likely order they would have been drawn while drawing the shape). There was a total of 18 shapes used in the experiment (details on criteria for including various shapes are shared in section 3.4). A total of 750 samples were collected - 250 from friends and family and 500 from mechanical Turk. The below data is using only the data from friends and family. This was done, because the lower bound of entropy as calculated by Shannon assumes of ideal prediction, and I noticed that friends and family had done the experiment with much more seriousness and thought than many of the mTurk users. (Although this should not have a significant impact on the end results either way).

Table I shows summary of data for 240 samples corresponding to all shapes of sizes (N=13 to 35). The column corresponds to the number of preceding dots known to the participants; the row is the number of the guess. The entry in column N at row S is the number of times a subject guessed the right dot (or letter) at the Sth guess when (N-1) dots (or letters) were known.

Smoothed frequencies of Reduced Text with N

	1	2	3	4	5	6	7	8	9	10	11	12
1	18.4	66.7	64.2	55.2	57.7	63.7	56.5	58.6	59	71.4	62.4	82.1
2	14.6	15.4	12	27.4	23.1	22.7	26.5	18.9	17.1	9.5	20.6	11.2
3	19.3	3.9	8.6	10.7	9	7.3	8.2	9.5	10.7	5.6	8.6	3.9
4	9.6	2.9	3.1	1.4	2.1	1.3	1.8	2.7	2.7	5.6	1.8	0.6
5	9.6	3.9	3.1	1.4	2.1	1.3	1.8	2.7	2.7	2.8	1.8	0.6
6	9.6	2.9	3.1	1.4	2.1	1.3	1.8	2.7	2.7	2.8	1.8	0.6
7	9.6	2.9	3.1	1.4	2.1	1.3	1.8	2.7	2.7	2.8	1.8	0.6
8	9.6	2.9	3.1	1.4	2.1	1.3	1.8	2.7	2.7	2.8	1.8	0.6

Figure 10 Table I

For example, the entry 18.9 in column 8, row 2, means that with preceding 7 dots known, the correct dot was obtained on the second guess ~nineteen times out of hundred. Some other points worth noting are:

- The first dot (1) above corresponds to the 2nd dot of the shape, as the starting dot (from where the pen would have started drawing the shape) is given to the user.

-Smaller values of frequencies have been uniformly smoothed, especially for values in the right lower part of the table. This is done to somewhat overcome the worst sampling fluctuations. The lower numbers in this table are the least reliable and these were averaged together in groups.

The upper and lower bounds given by Eq (1) above, were then calculated for each column giving the following results:

	1	2	3	4	5	6	7	8	9	10	11	12
Lower	2.66	0.99	1.07	1.1	1.1	0.9	1.1	1.14	1.15	0.96	0.97	0.44
Upper	2.93	1.75	1.85	1.75	1.83	1.58	1.79	1.92	1.92	1.7	1.7	0.99

Figure 11 Lower and Upper Bounds on Entropy in Bits per Letter

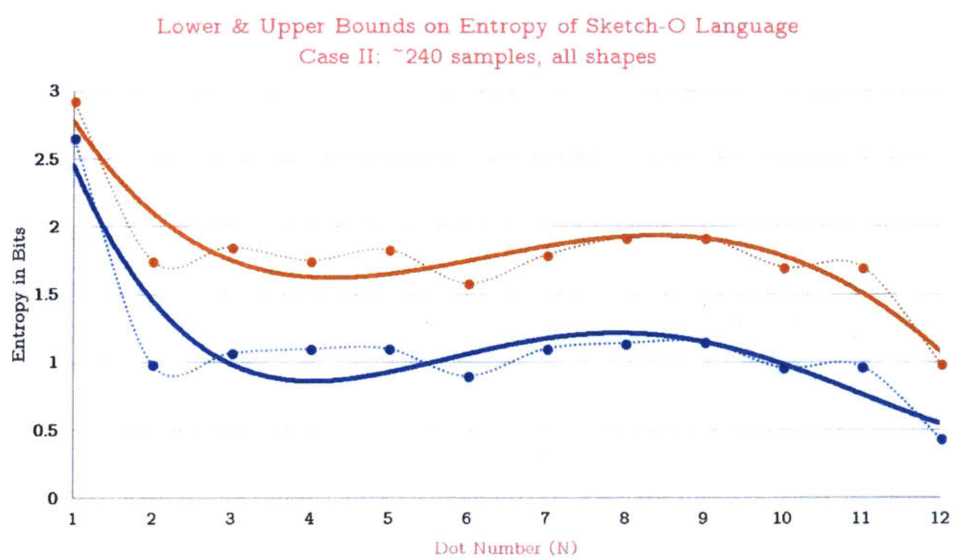


Figure 12 Upper and Lower experimental bounds for the entropy of 8-letter Sketch-O language for N ϵ [1,12]

And the upper and lower bounds on the Redundancy given by Eq (2) above, came out to be as:

Upper	11.24	66.88	64.44	63.18	63.22	**70**	63.37	**61.83**	61.8	68.03	67.51	**85.22**
Lower	2.3	41.74	38.21	41.56	38.89	**47.41**	40.39	**36**	35.88	43.35	43.39	**67.06**

In general we can say that the redundancy for Sketch-O language or for a language of shapes (single stroke shapes) has an upper bound between 60%-80% and a lower bound between 40%-60%.

Below is a snapshot of Non-smoothed data for shapes with length (21,22 & 23) amounting to ~100 samples.

Frequency of Reduced Text Symbols with N (number of dots)

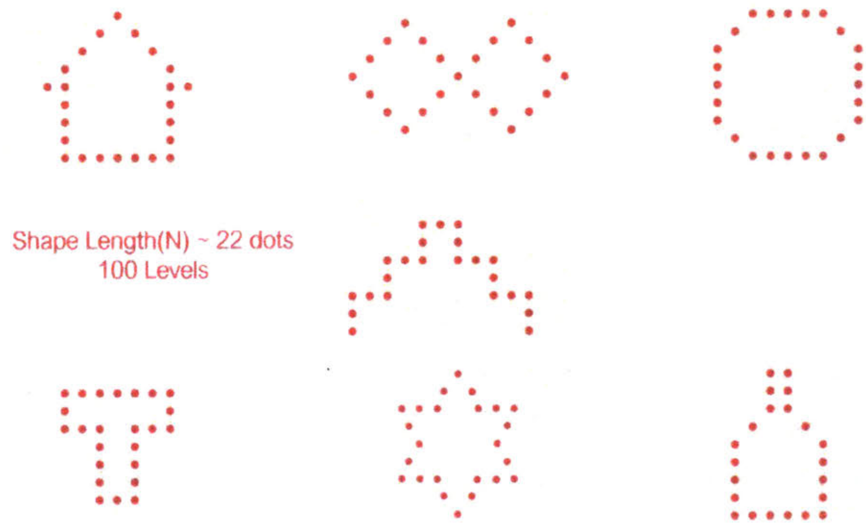
	1	3	5	7	8	10	12	14	16	18	20
1	19.2	39.4	47.9	60.7	75.6	78.8	88.3	80.9	91.5	90.5	88.3
2	17.1	13.9	17.1	29.8	17.1	9.6	4.3	17.1	6.4	7.5	9.6
3	17.1	18.1	17.1	3.2	3.2	4.3	4.3	0	1.1	1.1	2.2
4	8.6	9.6	7.5	6.4	2.2	5.4	2.2	2.2	1.1	1.1	0
5	9.6	11.8	7.5	0	0	1.1	0	0	0	0	0
6	9.6	4.3	1.1	0	2.2	1.1	1.1	0	0	0	0
7	9.6	3.2	2.2	0	0	0	0	0	0	0	0
8	9.6	0	0	0	0	0	0	0	0	0	0

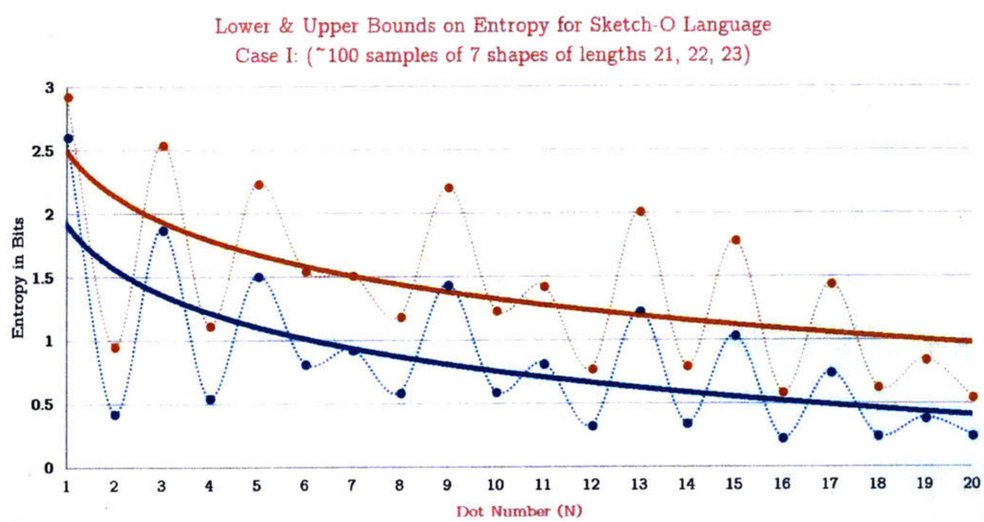
The upper and lower bounds given by Eq (1) above, for this case of ~100 samples of similar sized shapes (shape length = [21,22,23]) came out to be as:

	1	2	3	4	5	6	7	8	9	10
Lower	2.61	0.43	1.88	0.55	1.51	0.82	0.93	0.59	1.44	0.6
Upper	2.93	0.96	2.55	1.12	2.24	1.55	1.52	1.19	2.21	1.24

	11	12	13	14	15	16	17	18	19	20
Lower	0.82	0.33	1.23	0.35	1.04	0.23	0.75	0.25	0.39	0.25
Upper	1.43	0.78	2.02	0.8	1.79	0.59	1.45	0.63	0.85	0.55

Figure 13 Lower and Upper Bounds on Entropy in Bits per Letter for 100 samples of shapes with sizes 21, 22 & 23.





Looking at the graphs for experimental bounds on the Entropy, we see:

- The first two dots give a lot of information about a shape drawing, as we can connect the dots mentally and get an idea about the direction of the drawing stroke.
- The graph for subsequent dots rises and falls, as knowing one dot, we have a fairly good idea about the next. But then after the next

dot, the shape might change its contour (these results are an aggregate of 17 different shapes in Case I (with ~250 samples) and 7 in the above case with ~100 samples)

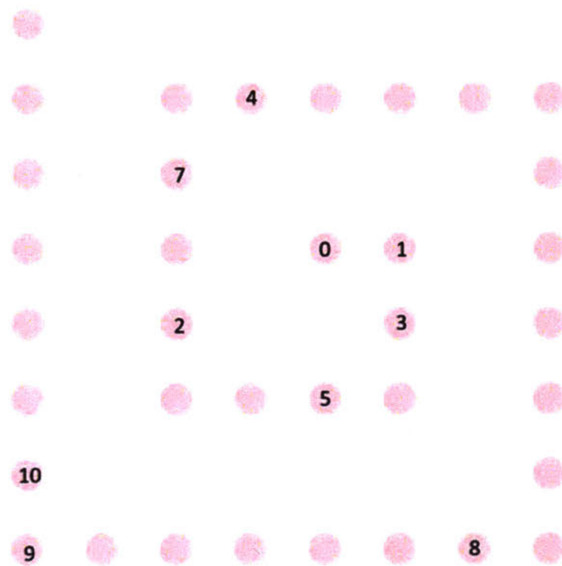
4.5 Most Information Rich Dots of a shape

By calculating the average information gained per dot (or Entropy) for individual shapes, we can get a ranking of the most informative dots of a shape. This gives us the most efficient way to communicate maximum information about the underlying shape with the least number of dots. Below I share the most information rich dots for two shapes: a) Square Spiral b) Heart:

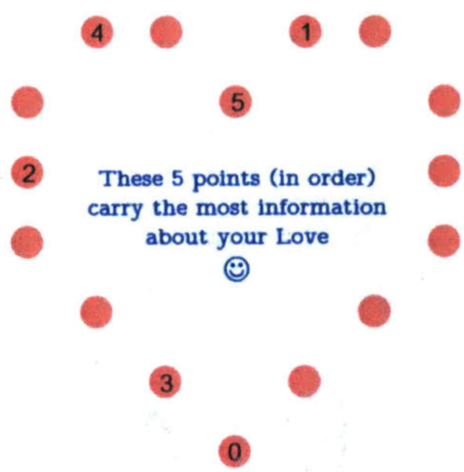
a) Square Spiral

Avg. Inf per Dot	**2.7**	**2.4**	**2.3**	**2.3**	**2.1**	**1.6**	**1.3**	**1.1**	**0.9**	**0.7**
Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**

Average Redundancy: 79.6%



b) Heart



Bits & Pieces

Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Entropy Lower	2.97	2.29	2.23	1.83	1.56	1.33	1	0.5	0.17	0.17	0.06	0	0	0	0

This gives us the first steps to answer the general question in section 3.6 (GoDot Cavemen Problem).

4.6 Mental Simulation: getting more Information than is

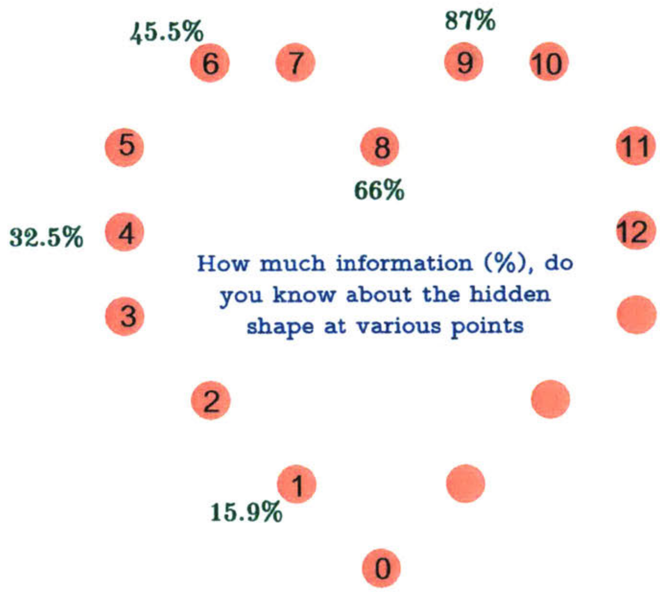


Figure 14 By dot 9 you are highly likely to guess that the hidden shape is a Heart!

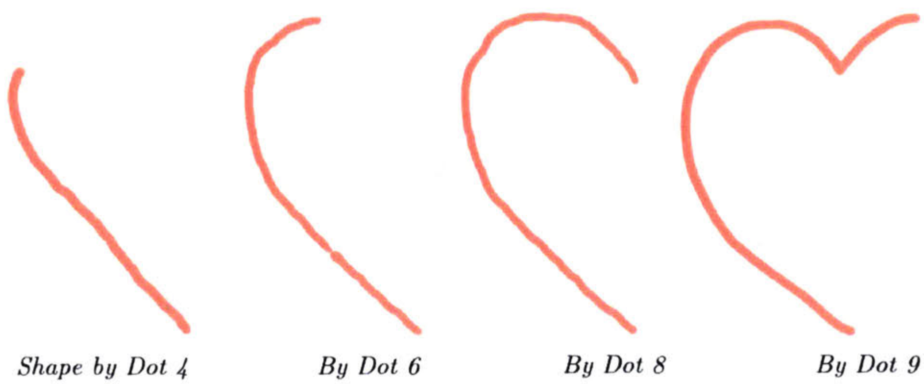


Figure 15 Increasing likelihood of the underlying shape being a Heart at various stages.

Dot Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Amount of Information (%) known about shape	15.9%	15.9%	16.3%	32.5%	32.5%	45.5%	54.9%	66%	87%	88.2%	88.2%	91.8%	98.8%	100%	100%

4.6.1 Three Cases: Evidence of Mental Simulation

By plotting the running total of the 'average information per dot' with the number of dots revealed (currently visible), we get evidence that people use a forward model of possible underlying shapes to simulate incomplete shapes beyond what is visible.

Let us see three different cases of the above phenomenon by making plots for three individual shapes:

Case I: Square Spiral

Case II: Heart

Case III: Random Scribble (a non-sensical placement of dots)

Case I: Square Spiral

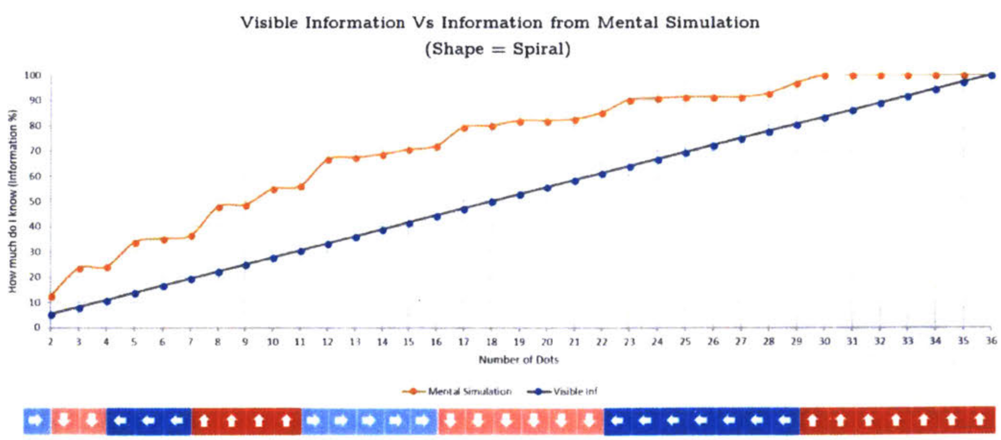
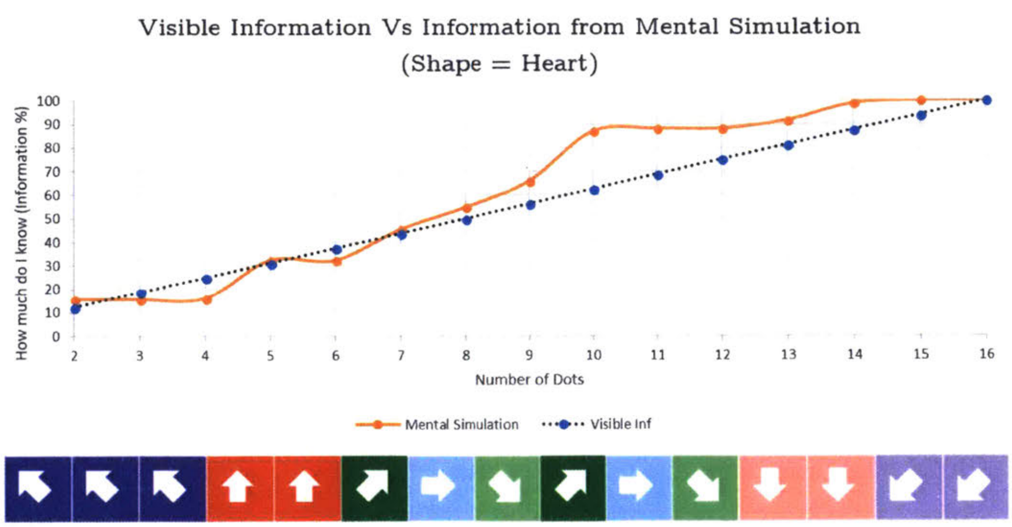


Figure 16 Plot showing Information from Mental Simulation, with Sketch-O sequence of shape below. By comparing the both we can see at what points of the sequence there is an upsurge in information from mental simulation!

A square spiral is a regular shape with a repeating pattern of dots increasing by a count of 1 at every turn. In the above plot we can see at what points in the Sketch-O sequence of the shape there is an upsurge in the information one gets from Mental Simulation.

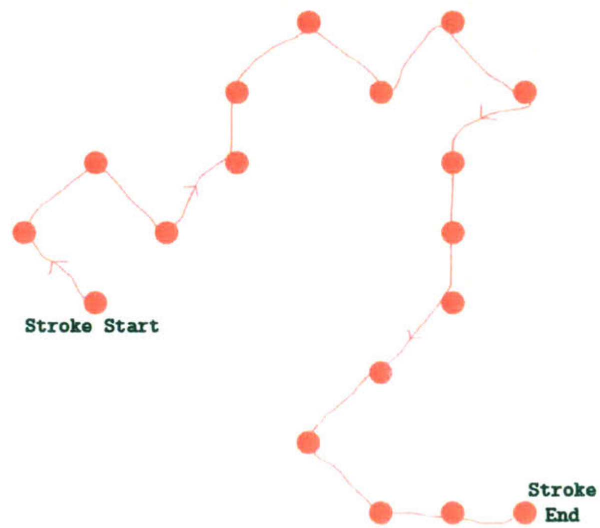
Case II: Heart

The Heart is an interesting case, where in the beginning of the shape, we have lesser information about the shape than is visible, this is a form of mis-information due to many contours the shape can converge to at that point. After dot 7 the user reaches the tipping point where he starts simulating the underlying shape as a Heart!

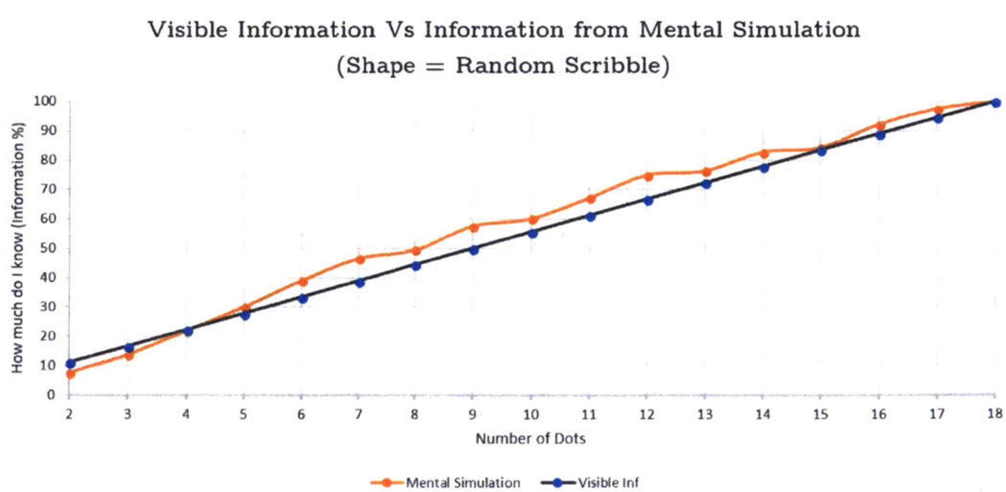


Case III: Random Scribble

The Random Scribble is a non-sensical dot pattern. It was generated by me, so it cannot be called a fully random shape. By looking at the

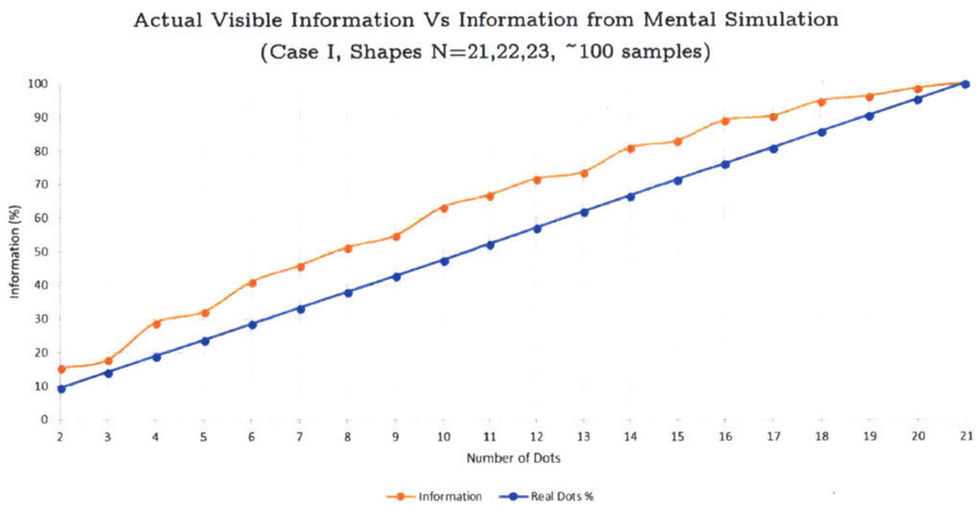


plot for the 'Random Scribble' shape we can easily see that a user is not able to simulate and gain a lot of information than what is being revealed to him.



For a completely random scribble, points for both the parameters would have completely overlapped.

The same plot when made for aggregate data from 100 samples of 7 shapes (with shape lengths ϵ [21, 22, 23]) is as follows:



5 Next Steps

5.1 Multi Stroke Shape Drawings

The immediate next step would be to extend this study for multi stroke shape drawings, which include most letters, numbers, and common objects. Single stroke shapes form the compositional blocks that form multi stroke shapes. In section 2.3 the jump alphabet had already been defined as part of the Sketch-O language. With the Jump symbol, in addition to the 8 arrow symbols, Sketch-O can generate any multi stroke shape on our sketchbook grid. Next, we can calculate bounds on the entropy and redundancy of multi stroke shapes by using a similar experimental procedure as described in this thesis. These bounds will be much better at approximating the entropy bounds for the cognitive language of geometric concepts, then the ones calculated for single stroke shapes.

5.2 Modeling the task

5.2.1 Bayesian Program Learning framework

A good first step to start modeling the task of generating shape drawings, would be to take cues from the work on - Concept learning as motor program induction [9], [10]. It is closely aligned with the approach taken in this thesis of modeling constructive actions as a language to generate shapes.

5.2.2 Training Dot RNN + refining using RL

Since this thesis thinks of shapes as coming from a language (Sketch-O), using RNN's (recurrent neural network) seems like a good approach for training a dot-RNN that can learn rules of this language. For the training data, I propose using the Google Draw dataset to act as the shape language corpus. Then this RNN could be further tuned using RL (reinforcement learning) as described in the RL Tuner Model from [11]. Reinforcement Learning will help learn domain specific constraints (eg: searching for the next dot near a found dot, as shapes are continuous- this is for the modeling task as described in the general version of the experiment) and the RNN will reflect the information learned from the data.

6 Conclusion

In this thesis I develop a rich, novel experimental paradigm for studying various aspects of the cognitive language of geometric concepts. I propose looking at constructive actions as a language and create a sub-language Sketch-O for generating shape drawings. Then I use a sequential modification of the broader experiment to calculate the bounds on entropy and redundancy of this language. The experimental setup thus used generalizes Shannon's prediction experiment for a wide variety of languages, beyond only text-based. The approximate entropy bounds for single stroke shape drawings, lie between 0.4 bits/letter to 0.8 bits/letter, and our further reduced with longer shape lengths. I then compute entropy (average information per letter) values for individual shapes and use them to show evidence of subjects using a rich forward model to mentally simulate incomplete shapes, thus gaining information about the underlying shape more than is physically visible. I further show evidence by testing subjects with a non-sensical shape (~a random scribble) and use its data to show that unlike regular everyday shapes, subjects fail to mentally simulate the random shape and almost no information beyond what is visible.

References

[1] Papert, Seymour. *Mindstorms: Children, computers, and powerful ideas.* Basic Books, Inc., 1980.

[2] Shannon, Claude E. "Prediction and entropy of printed English." *Bell Labs Technical Journal* 30.1 (1951): 50-64.

[3] Shannon, Claude E., Warren Weaver, and Arthur W. Burks. "The mathematical theory of communication." (1951).

[4] C. E. Shannon, "The redundancy of English," in Trans. 7th Conf. Cybern., Mar. 1950, pp. 123–158.

[5] Attneave, Fred. "Some informational aspects of visual perception." *Psychological review* 61.3 (1954): 183.

[6] Zabrodsky, Hagit. *Symmetry: A Review.* Leibniz Center for Research in Computer Science, Department of Computer Science, Hebrew University of Jerusalem, 1990.

[7] Wagemans, Johan. "Characteristics and models of human symmetry detection." *Trends in cognitive sciences* 1.9 (1997): 346-352.

[8] Koffka, Kurt. "Principles of Gestalt Psychology, International Library of Psychology, Philosophy and Scientific Method." (1935).

[9] Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2012). Concept learning as motor program induction: A large-scale empirical study. In Proceedings of the 34th Annual Conference of the Cognitive Science Society.

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

[11] Jaques, Natasha, et al. "Tuning recurrent neural networks with reinforcement learning." (2017).