The neural bases of reading computer programs

Talk based on our work - https://elifesciences.org/articles/58906

Shashank
CSAIL, MIT

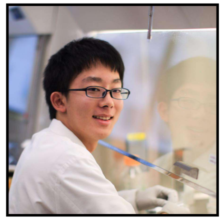
shash@mit.edu @ShashankSrikant

ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

MIT

Anna Ivanova



Yotaro Sueoka

Hope Kean

Ev Fedorenko

Una-May O'Reilly

Tufts

Riva Dhamala

Marina Bers

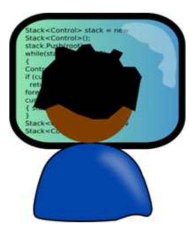
Some slides adapted from Nancy Kanwisher's course on The Human Brain (9.17, 2019)

ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

Big picture

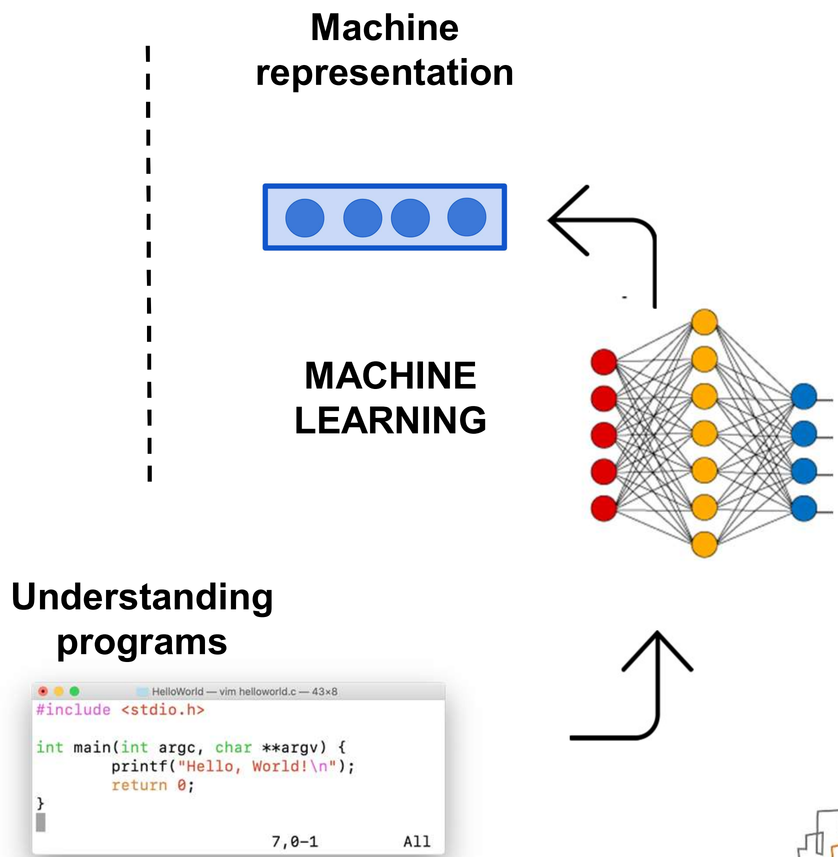
Understanding programs



```
HelloWorld — vim helloworld.c — 43×8
#include <stdio.h>

int main(int argc, char **argv) {
        printf("Hello, World!\n");
        return 0;
}
                              7,0-1         All
```

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL



ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

GENERATING ADVERSARIAL COMPUTER PROGRAMS USING OPTIMIZED OBFUSCATIONS

Shashank Srikant[1] Sijia Liu[2] Tamara Mitrovska[1] Shiyu Chang[2]
Quanfu Fan[2] Gaoyuan Zhang[2] Una-May O'Reilly[1]

[1]CSAIL, MIT [2]MIT-IBM Watson AI Lab
shash@mit.edu, Sijia.Liu@ibm.com, unamay@csail.mit.edu

ICLR, 2021

Dependency-Based Neural Representations for Classifying Lines of Programs

Shashank Srikant, shash@mit.edu, CSAIL, MIT
Nicolas Lesimple, nicolas.lesimple@alumni.epfl.ch, EPFL
Una-May O'Reilly, unamay@csail.mit.edu, CSAIL, MIT

arXiv, 2020

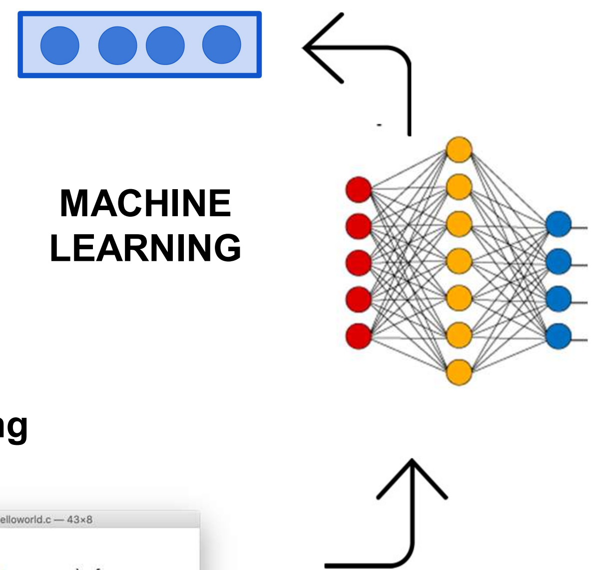
A System to Grade Computer Programming Skills using Machine Learning

Shashank Srikant, Aspiring Minds, shashank.srikant@aspiringminds.in
Varun Aggarwal, Aspiring Minds, varun@aspiringminds.in

KDD 2014, 2016

ALFA
ANYSCALE LEARNING FOR ALL

Machine representation



MACHINE LEARNING

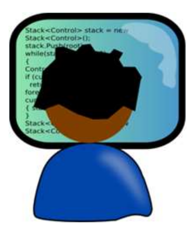
Understanding programs

```
HelloWorld — vim helloworld.c — 43×8
#include <stdio.h>

int main(int argc, char **argv) {
        printf("Hello, World!\n");
        return 0;
}
                              7,0-1          All
```

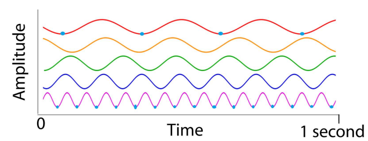
MIT CSAIL

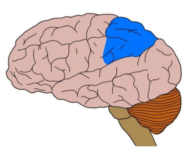
Big picture

Understanding programs



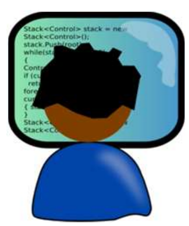
Neural representation





COGNITIVE NEUROSCIENCE

Understanding programs



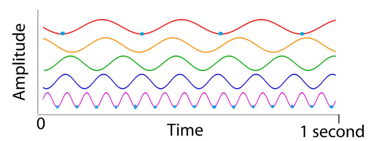
HelloWorld — vim helloworld.c — 43×8

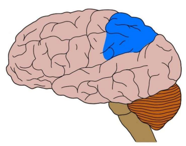
```
#include <stdio.h>

int main(int argc, char **argv) {
        printf("Hello, World!\n");
        return 0;
}
                        7,0-1          All
```

ALFA
ANYSCALE LEARNING FOR ALL

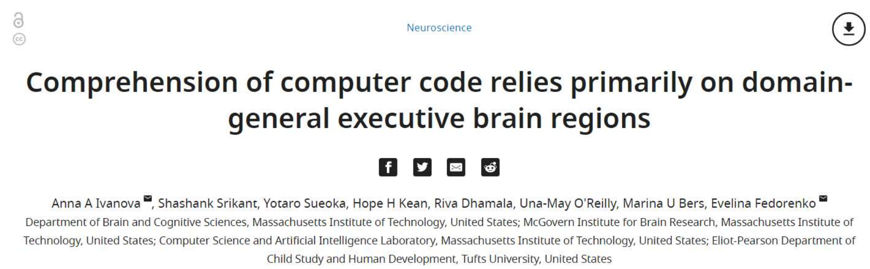
MIT CSAIL

Neural representation



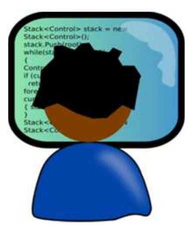


COGNITIVE NEUROSCIENCE



eLife, 2020

Understanding programs

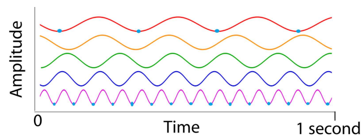


ALFA
ANYSCALE LEARNING FOR ALL

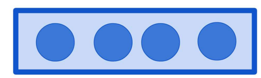
MIT CSAIL

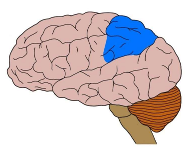
Big picture

Neural representation



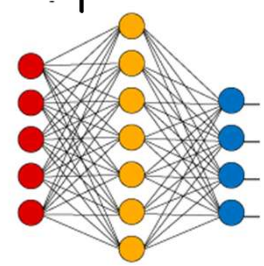
Machine representation



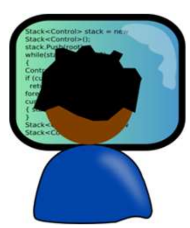


COGNITIVE & NEUROSCIENCE

MACHINE LEARNING



Understanding programs



```
#include <stdio.h>

int main(int argc, char **argv) {
        printf("Hello, World!\n");
        return 0;
}
                                   7,0-1          All
```

ANYSCALE LEARNING FOR ALL

MIT CSAIL

Investigating the human brain

Investigating the human brain

- Theory — Hansen et al. 2012
- Behavioral studies — Casalnuovo et al. 2020; Crichton et al. 2020
- Eye-tracking — Turner et al. 2014; Sharafi et al. 2015; McChesney et al. 2019; Pietek et al. 2020
- Neuroimaging
  - EEG — Busjahn et al. 2014
  - fMRI — Siegmund et al. 2014; Floyd et al. 2017
  - fNIRS — Ikutani et al. 2014
- Patient studies — ?

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Understanding the human brain

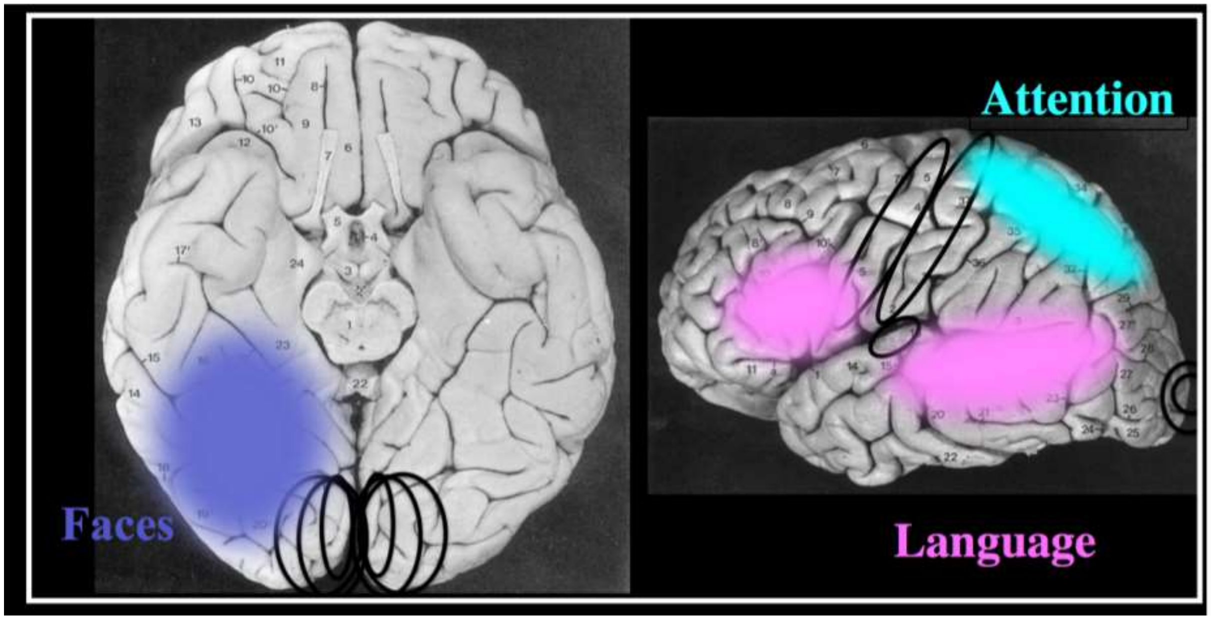
Understanding the human brain

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language

Understanding the human brain

Early 90s



ALFA ANYSCALE LEARNING FOR ALL

Slide adapted from Nancy Kanwisher's course on The Human Brain (9.17, 2019)

MIT CSAIL

Understanding the human brain

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language

Understanding the human brain

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language

Faces

Color

Places

Words/letters

Bodies

Motion

Shape

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Understanding the human brain

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language

Responds to both comprehension and production

Across modalities (speech, written, ASL)

Responds to typologically diverse languages

Causally important for language

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Understanding the human brain

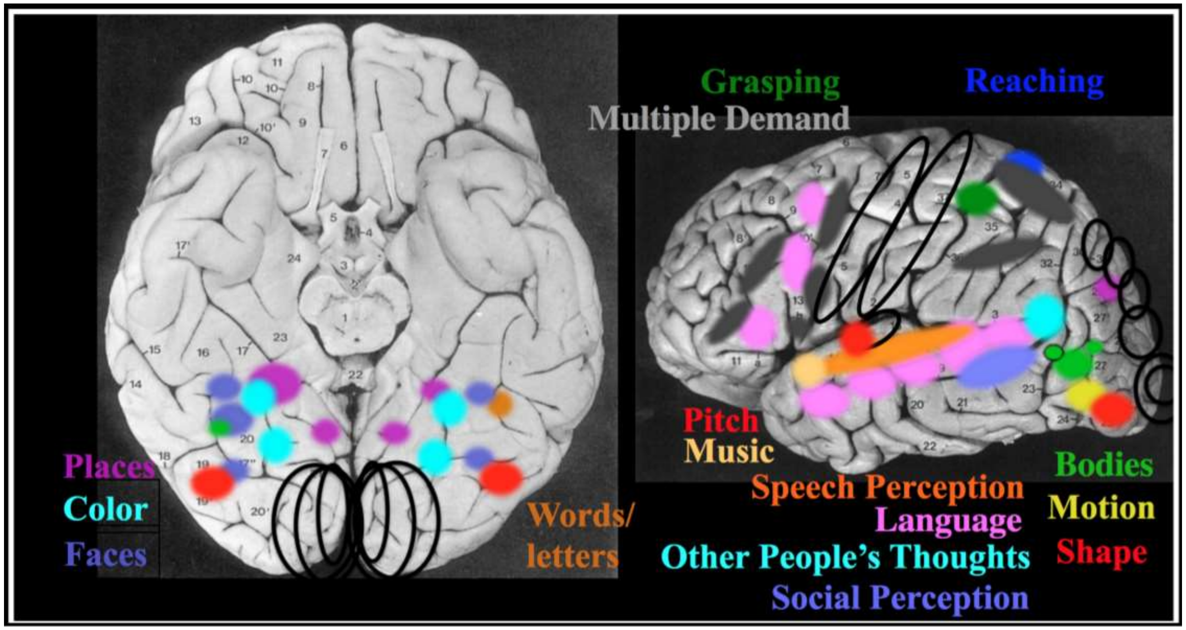
Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Broadly recruited in math, logic, reasoning, learning like tasks

Understanding the human brain

Current understanding



ALFA ANYSCALE LEARNING FOR ALL

Slide adapted from Nancy Kanwisher's course on The Human Brain (9.17, 2019)

MIT CSAIL

RESEARCH ARTICLE | COGNITIVE NEUROSCIENCE

Numerical cognition in honeybees enables addition and subtraction

Scarlett R. Howard[1], Aurore Avarguès-Weber[2], Jair E. Garcia[1], Andrew D. Greentree[3] and Adrian G. Dyer[1,4,*]
+ See all authors and affiliations

Science Advances 06 Feb 2019:
Vol. 5, no. 2, eaav0961
DOI: 10.1126/sciadv.aav0961

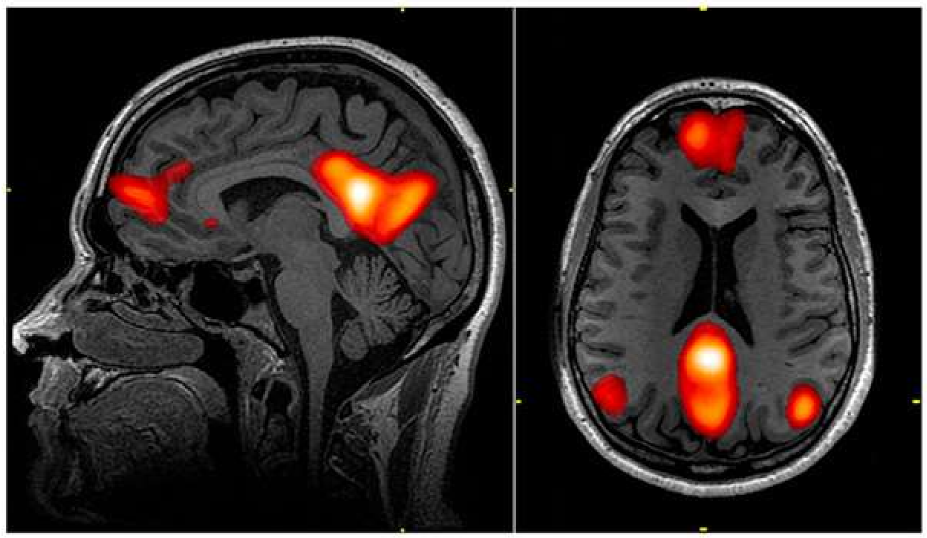
fMRI

fMRI

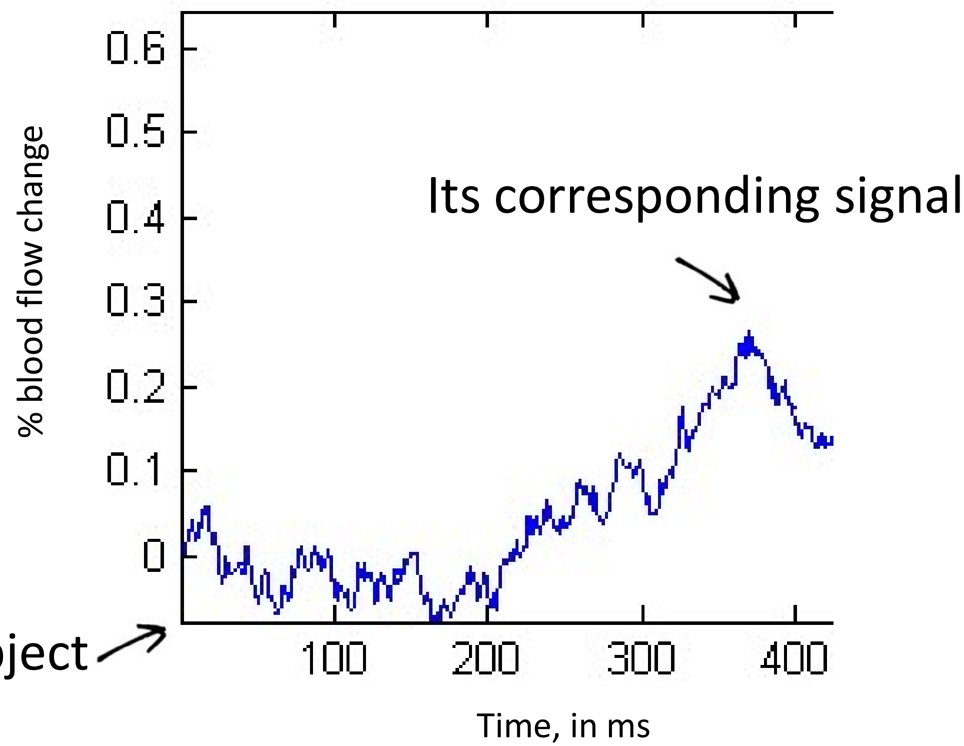
State of the art to investigate which areas of the brain involved in an action

fMRI

Measures blood flow changes in a region of interest (ROI)



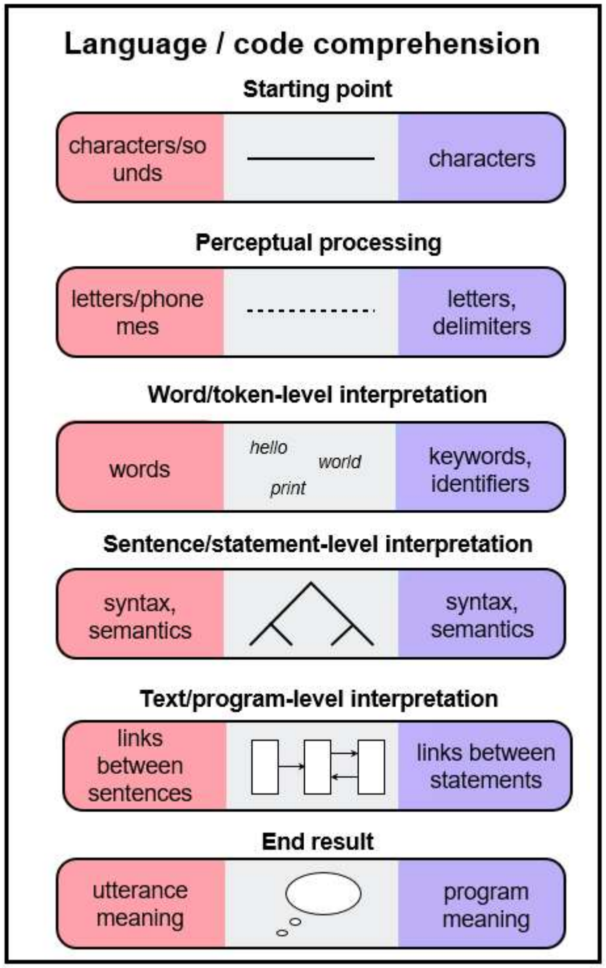
fMRI



Absolute measurements meaningless. Comparative analysis.

Programming

Programming

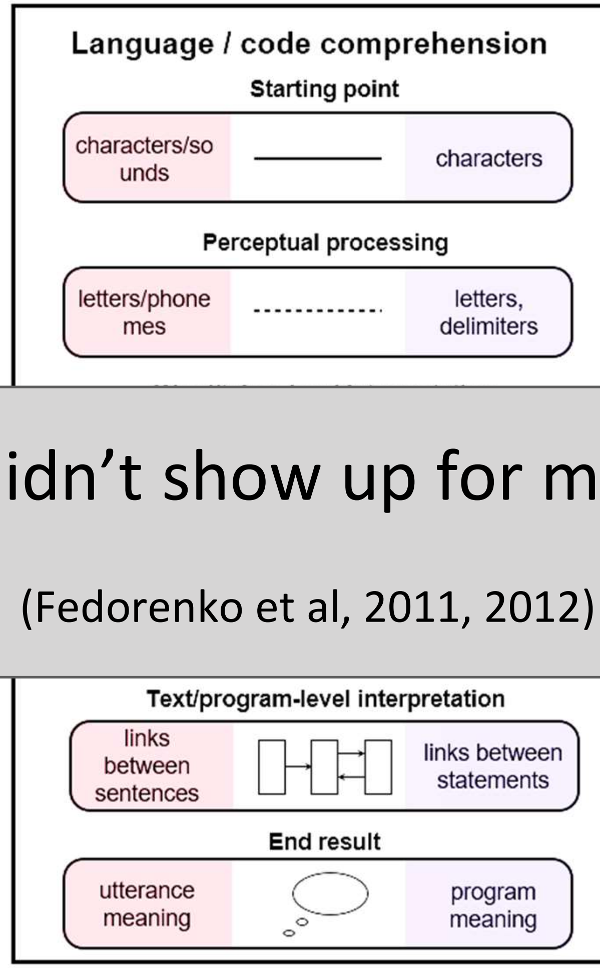


Fedorenko, Ivanova et al, 2019

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Programming



The parallel didn't show up for music or math.

(Fedorenko et al, 2011, 2012)

Fedorenko, Ivanova et al, 2019

Experiment Design

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Understanding code?

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Understanding code

1. Vision system activated

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Understanding code

1. Vision system activated
2. Recognize characters, tokens to form statements and blocks.

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Understanding code

1. Vision system activated
2. Recognize characters, tokens to form statements and blocks.
3. Understand what the code does

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Understanding code

1. Vision system activated
2. Recognize characters, tokens to form statements and blocks.
3. Understand what the code does
4. Mentally trace it/debug it and calculate output.

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Code comprehension?

1. Vision system activated
2. Recognize characters, tokens to form statements and blocks.
3. Understand what the code does
4. Mentally trace it/debug it and calculate output.

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Code comprehension

1. Vision system activated
2. Recognize characters, tokens to form statements and blocks.
3. Understand what the code does
4. Mentally trace it/debug it and calculate output.

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

Code simulation

1. Vision system activated
2. Recognize characters, tokens to form statements and blocks.
3. Understand what the code does
4. Mentally trace it/debug it and calculate output.

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Where to look?

Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system

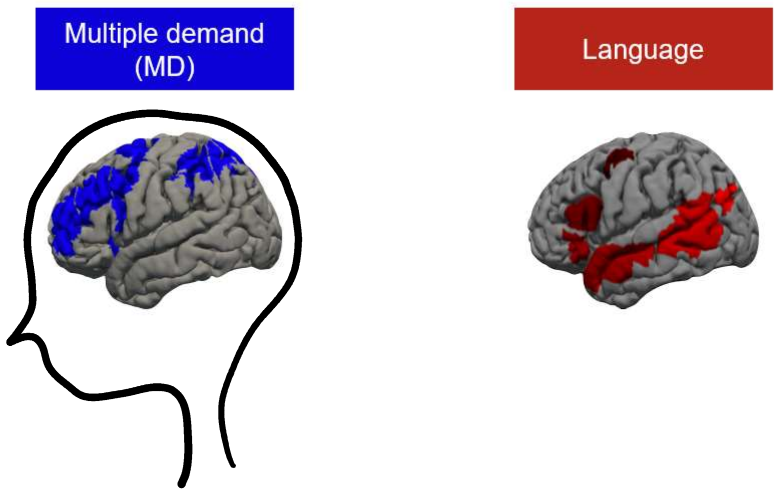
Code reading

1. Vision system activated
2. Recognize characters, tokens to form statements and blocks.
3. Understand what the code does
4. Mentally trace it/debug it and calculate output.

Where to look?

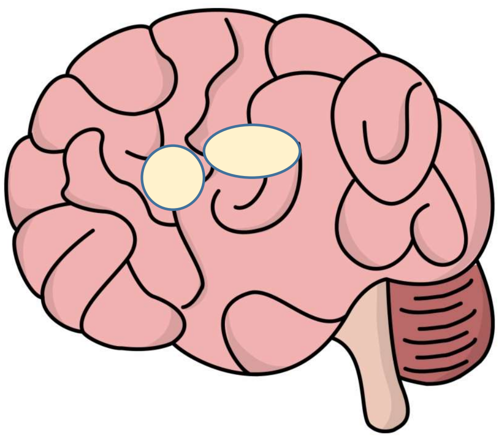
Broad functions

- Vision
- Audio
- Motor control and dexterity
- Emotions
- Language
- Multiple Demand system



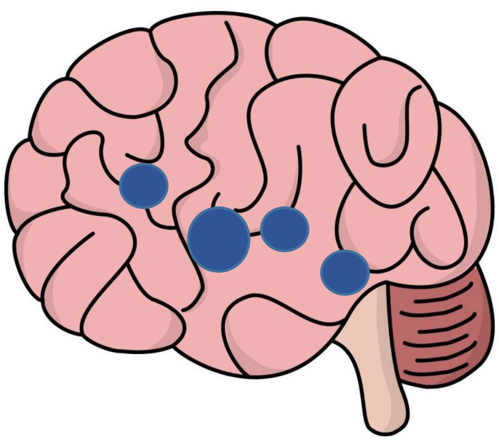
General strategy

1. Use localizers to pin down the regions of interest (ROIs)



Localizers for MD, Language system

● Language localized regions



Sentence Reading

NOBODY COULD HAVE PREDICTED THE EARTHQUAKE

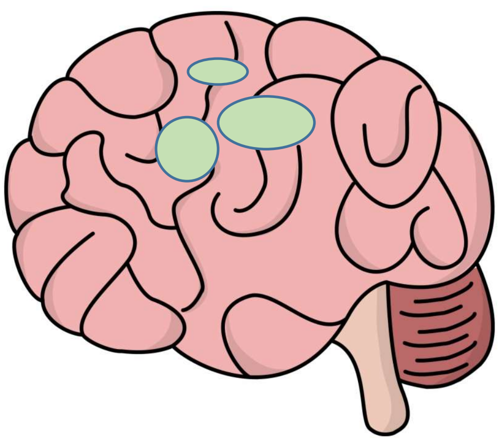
Non-word Reading

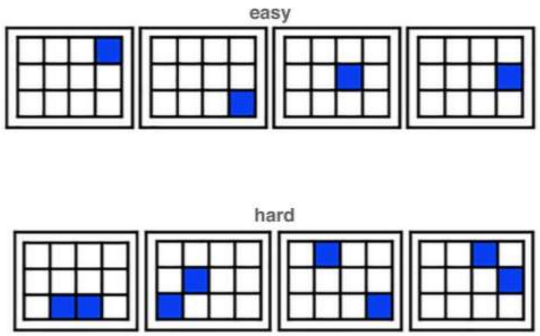
BIZBY ACWORILLY BUSHU SNOOKI BILIBOP KUKEE

MIT CSAIL

Localizers for MD, Language system

MD localized regions

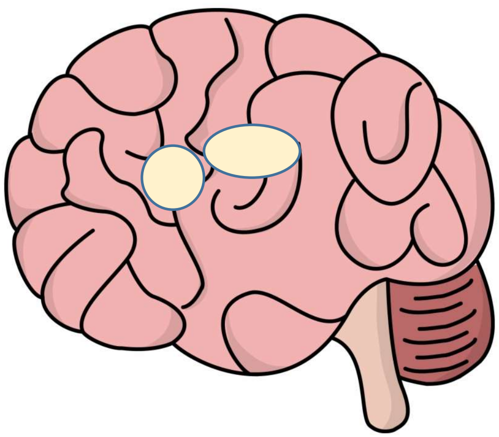


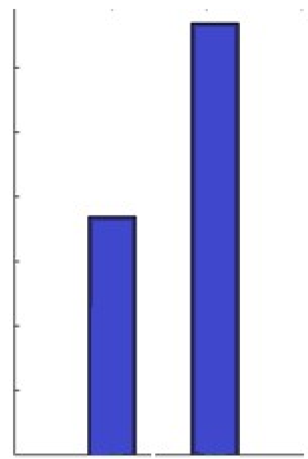


MIT CSAIL

General strategy

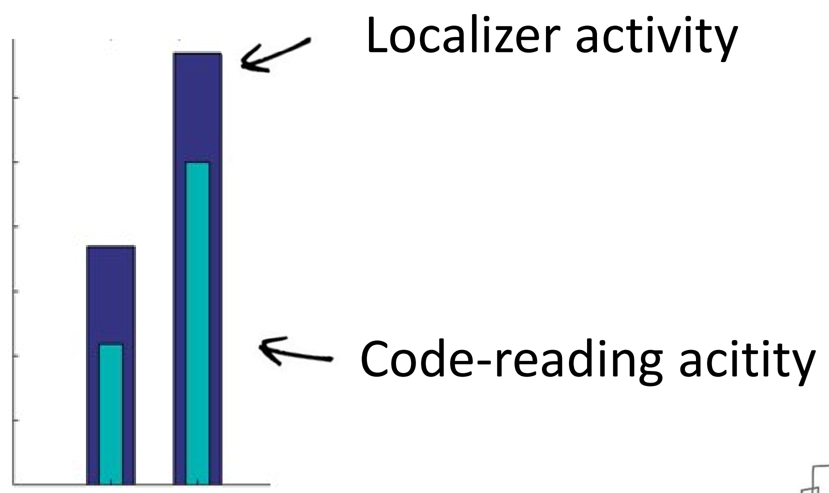
1. Use localizers to pin down the regions of interest (ROIs)





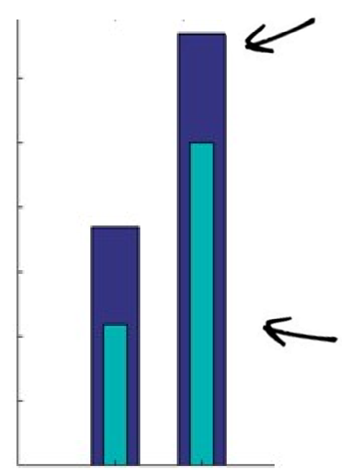
General strategy

2. In those ROIs, measure code-reading activity



General strategy

3. Infer whether that ROI processes code reading



What to look for?

What code-reading activity should we measure?

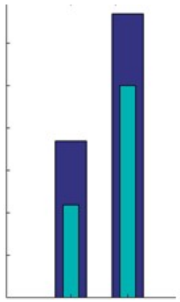
What are the experiment conditions which will help measure such activity?

Condition 1

Code versus Non-code

Condition 1

Code versus Non-code



```
big_num, small_num = 64, 16

if big_num % small_num == 0:
   print(1)
else:
   print(0)
```

Sentence Reading

NOBODY COULD HAVE PREDICTED THE EARTHQUAKE

Non-word Reading

BIZBY ACWORILLY BUSHU SNOOKI BILIBOP KUKEE

ANYSCALE LEARNING FOR ALL

MIT CSAIL

Condition 2

```
big_num, small_num = 64, 16

if big_num % small_num == 0:
   print(1)
else:
   print(0)
```

Code understanding

1. Vision system activated
2. Recognize characters, tokens to form statements and blocks.
3. Understand what the code does
4. Mentally trace it/debug it and calculate output.

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Condition 2

Disambiguate *code comprehension* and *code simulation*

Condition 2

code	sent

big_num, small_num = 64, 16
 	
if big_num % small_num == 0:
   print(1)
else:
   print(0)
	You are given two numbers 64 and 16. If the remainder when the first number is divided by the second number is 0, you perform one good deed. Otherwise, you perform no good deeds. How many good deeds will you perform?

filename = "alphabet.java" modified = filename.split(".") print(modified[-1])	A file is named "alphabet.java". You split the name at the dot character. What is the last part of the resulting split?

Condition 2

code comprehension

+

code simulation

```
filename = "alphabet.java"
modified = filename.split(".")

print(modified[-1])
```

code

sentence comprehension

+

code simulation

A file is named "alphabet.java". You split the name at the dot character. What is the last part of the resulting split?

sent

Condition 2

[*code comprehension* + *code simulation*] − [*sentence comprehension* + *code simulation*]

```
filename = "alphabet.java"
modified = filename.split(".")

print(modified[-1])
```

A file is named "alphabet.java". You split the name at the dot character. What is the last part of the resulting split?

code > **sent**

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Condition 3

Effect of meaningful variable names

English identifiers

```
height = 5
weight = 100
bmi = weight / (height*height)
print(bmi)
```

Japanese identifiers

```
sincho = 5
taiju = 100
keisu = taiju / (sincho*sincho)
print(keisu)
```

Condition 4

Different control and data-operations

Control – for, if, sequential

Data – math, string

```
string1 = "onion"
new_string = string1[-2:]
multiplied = new_string*4
print(multiplied)
```

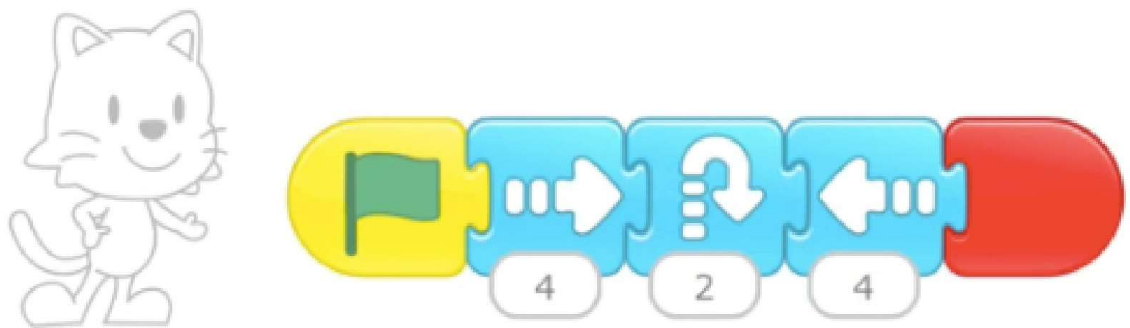
```
big_num, small_num = 64, 16
if big_num % small_num == 0:
        print(1)
else:
        print(0)
```

```
nums = [10, 2, 30]
prod = 1
for n in nums:
    prod = prod*n
print(prod)
```

Condition 5

Scratch Junior

code



sent

Kitten walks right, jumps, and then walks left

Setup

- Two separate experiments – Python and Scratch Junior
- Python – 24 participants
- Scratch Junior – 19 participants
- No participant saw multiple versions of the same question

Is the Language system processing code reading?

Sentence Reading

NOBODY COULD HAVE PREDICTED THE EARTHQUAKE

Non-word Reading

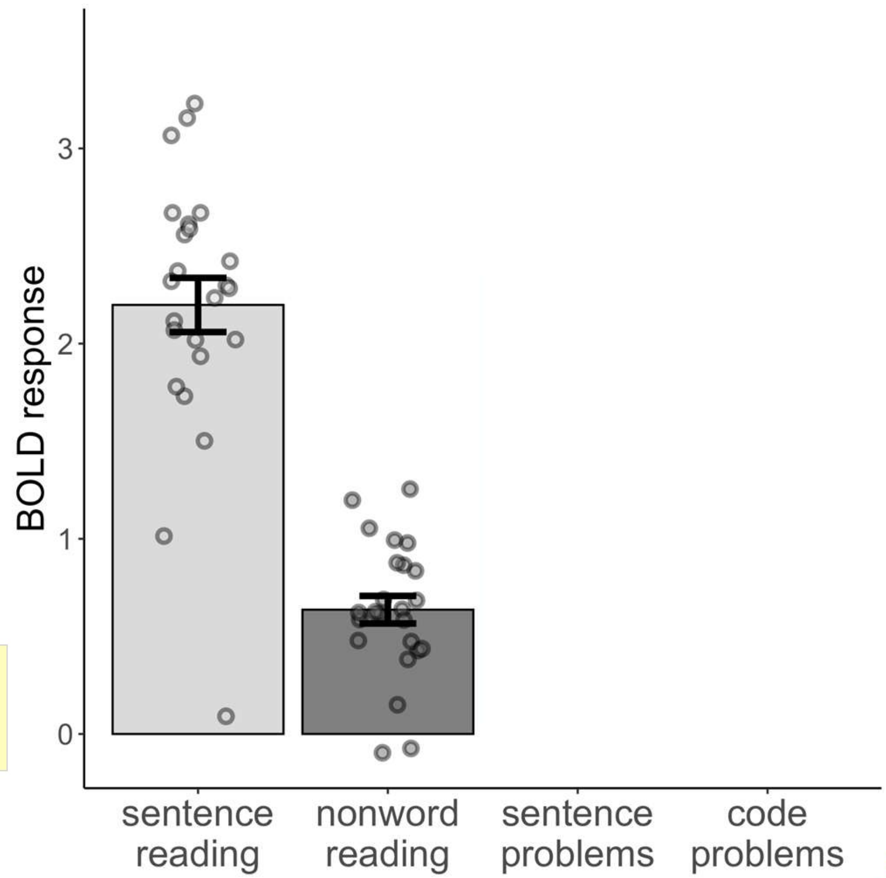
BIZBY ACWORILLY BUSHU SNOOKI BILIBOP KUKEE

```
filename = "alphabet.java"
modified = filename.split(".")

print(modified[-1])
```

code

A file is named "alphabet.java". You split the name at the dot character. What is the last part of the resulting split?

sent

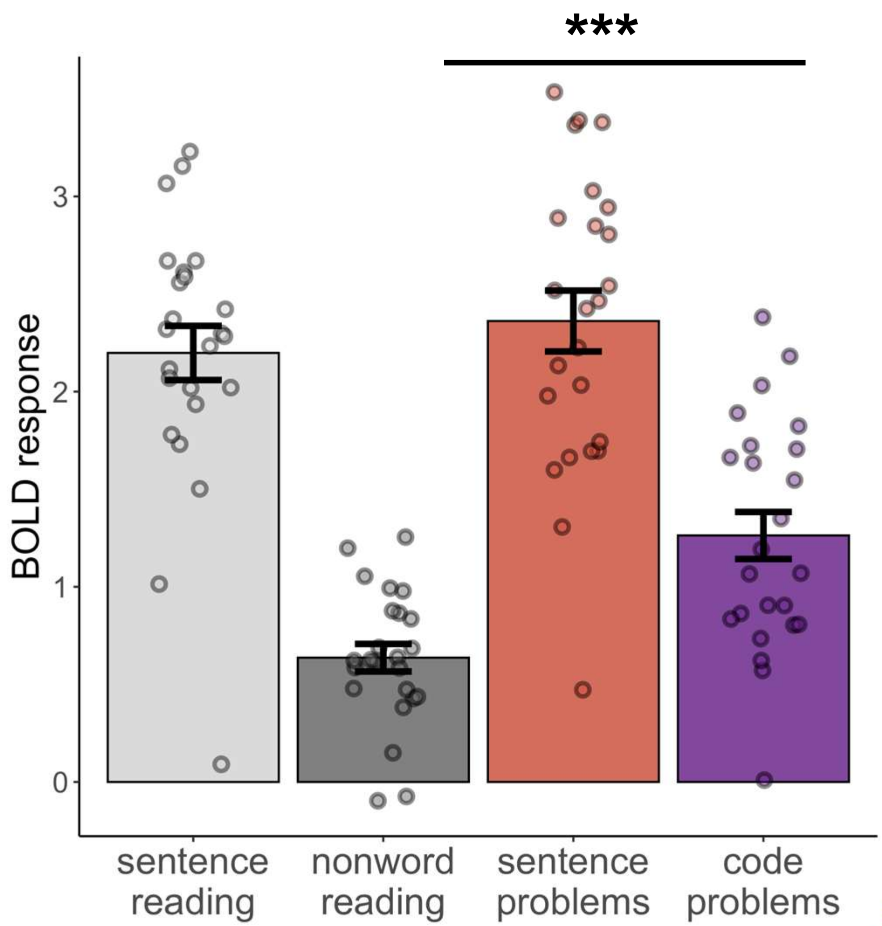


ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

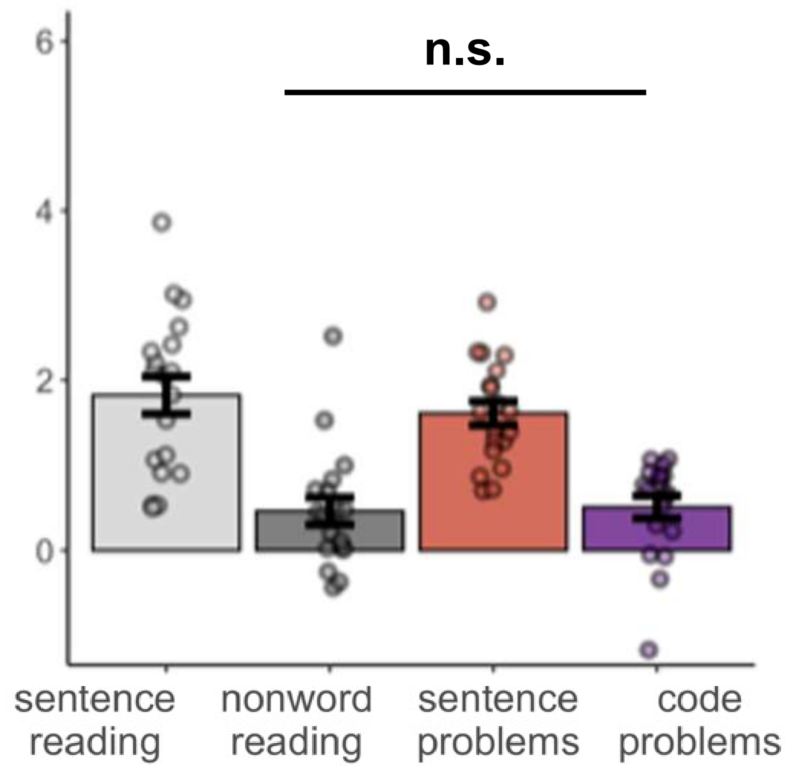
Is the Language system processing code reading?

Python



Is the Language system processing code reading?

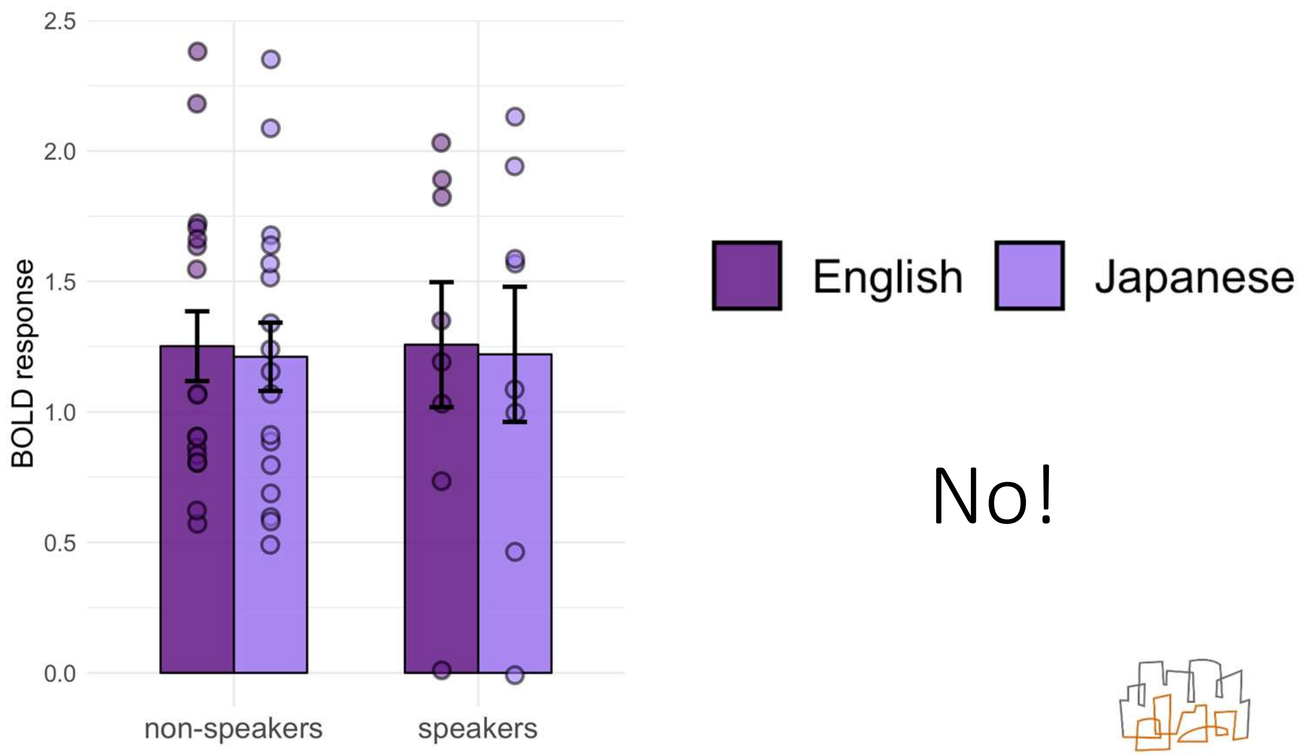
Scratch Junior



Is the Language system processing code reading?

No!

Are meaningful variable names driving the activity we see in Python?



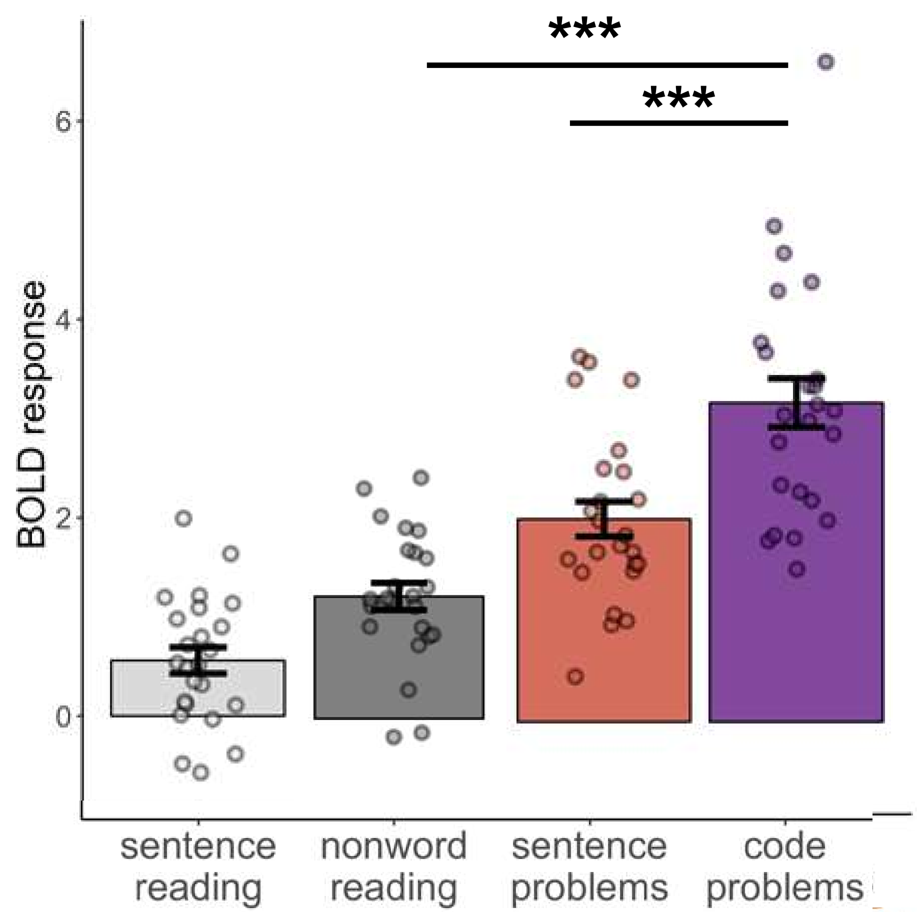
No!

ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

Is the MD system processing code reading?

Python

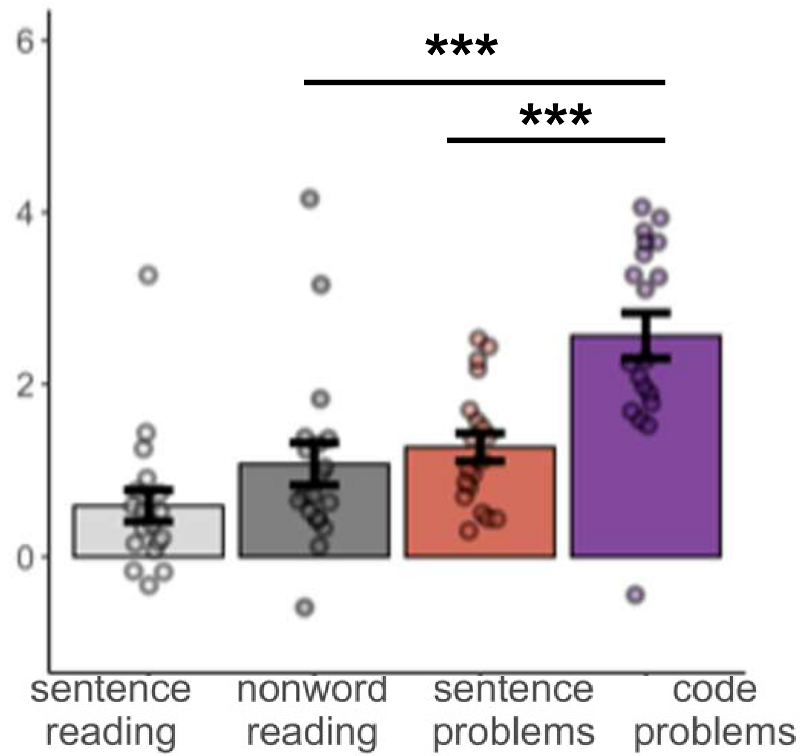


ALFA
ANYSCALE LEARNING FOR ALL

CSAIL

Is the MD system processing code reading?

Scratch Junior



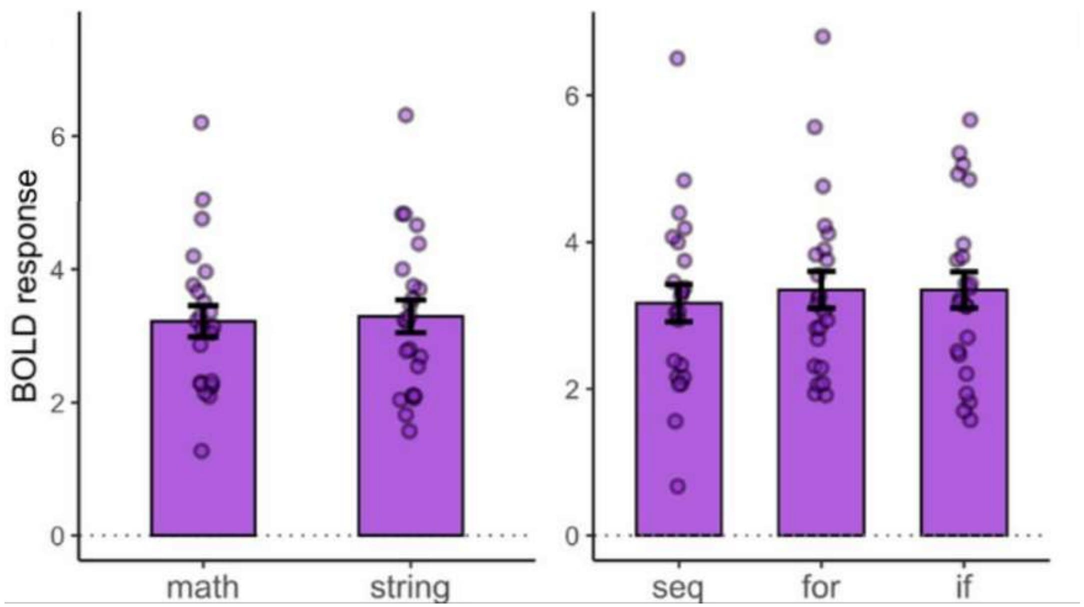
ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

Is the MD system processing code reading?

Yes!

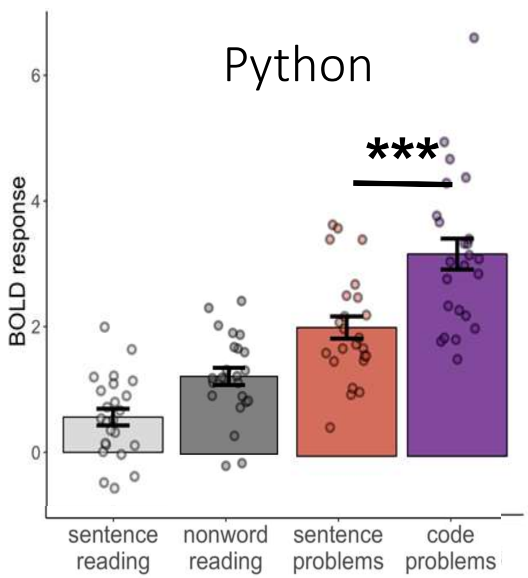
Does it generalize across control and data operations?

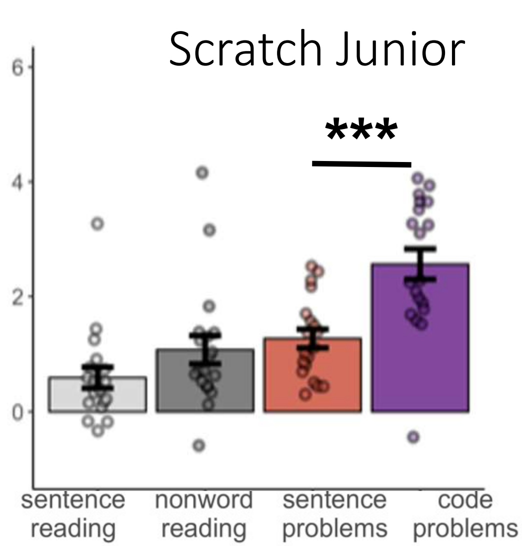


ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

Is the MD system processing *code comprehension*?





code problems = code comprehension + code simulation

sentence problems = sentence processing + code simulation

Yes!

ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

Is the MD system processing *code comprehension*?

The entire MD network is engaged

Not just regions in the network known to be involved in math and logic

Summary

	MD system	Language system
PYTHON	✓	✓
SCRATCH JR	✓	✗
	Strong, generalizable responses to code	**Moderate, task/language dependent (?) responses to code**

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

Another group comes to the same conclusion

New Results

Comment on this paper

Computer code comprehension shares neural resources with formal logical inference in the fronto-parietal network

Y. Liu, J. Kim, C. Wilson, M. Bedny

doi: https://doi.org/10.1101/2020.05.24.096180

This article is a preprint and has not been certified by peer review [what does this mean?].

Abstract Full Text Info/History Metrics Preview PDF

Previous

Posted June 01, 2020.

Download PDF

Supplementary Material

XML

Revision Summary

ALFA ANYSCALE LEARNING FOR ALL

MIT CSAIL

That’s great.

How do I use this information to improve my code reading skills?

PNAS

Improving fluid intelligence with training on working memory

Susanne M. Jaeggi*†‡, Martin Buschkuehl*†‡, John Jonides*, and Walter J. Perrig†

*Department of Psychology, University of Michigan, East Hall, 530 Church Street, Ann Arbor, MI 48109-1043; and †Department of Psychology, University of Bern, Muesmattstrasse 45, 3012 Bern, Switzerland

Edited by Edward E. Smith, Columbia University, New York, NY, and approved March 18, 2008 (received for review February 7, 2008)

Fluid intelligence (*Gf*) refers to the ability to reason and to solve new problems independently of previously acquired knowledge. *Gf* is critical for a wide variety of cognitive tasks, and it is considered one of the most important factors in learning. Moreover, *Gf* is closely related to professional and educational success, especially in complex and demanding environments. Although performance on tests of *Gf* can be improved through direct practice on the tests themselves, there is no evidence that training on any other regimen yields increased *Gf* in adults. Furthermore, there is a long history of research into cognitive training showing that, although performance on trained tasks can increase dramatically, transfer of this learning to other tasks remains poor. Here, we present evidence for transfer from training on a demanding working memory task to measures of *Gf*. This transfer results even though the trained task is entirely different from the intelligence test itself. Furthermore, we demonstrate that the extent of gain in intelligence critically depends on the amount of training: the more training, the more improvement in *Gf*. That is, the training effect is dosage-dependent. Thus, in contrast to many previous studies, we conclude that it is possible to improve *Gf* without practicing the testing tasks themselves, opening a wide range of applications.

dramatically, transfer of this learning to other tasks or domains remains shockingly rare (18–21).

Despite the many failures to find transfer in any domain, the sheer importance of identifying tasks that can lead to improvement in other tasks recommends continued investigation of transfer effects. With respect to *Gf*, the issue is whether one can identify a task that shares many of the features and processes of *Gf* tasks, but that is still different enough from the *Gf* tasks themselves to avoid mere practice effects. A recently proposed hypothesis by Halford *et al.* (22) might serve as a useful framework for the design of a transfer study in which one would like to improve *Gf* by means of a working memory task. Their claim is that working memory and intelligence share a common capacity constraint. This capacity constraint can be expressed either by the number of items that can be held in working memory or by the number of interrelationships among elements in a reasoning task. The reason for a common capacity limitation is assumed to lie in the common demand for attention when temporary binding processes are taking place to form representations in reasoning tasks (22). Other authors came to a related conclusion, stating that *Gf* and working memory are primarily

PNAS

Short- and long-term benefits of cognitive training

Susanne M. Jaeggi[1,2], Martin Buschkuehl[1,2], John Jonides, and Priti Shah

Department of Psychology, University of Michigan, Ann Arbor, MI 48109-1043

Edited by Dale Purves, Duke University Medical Center, Durham, NC, and approved May 17, 2011 (received for review March 1, 2011)

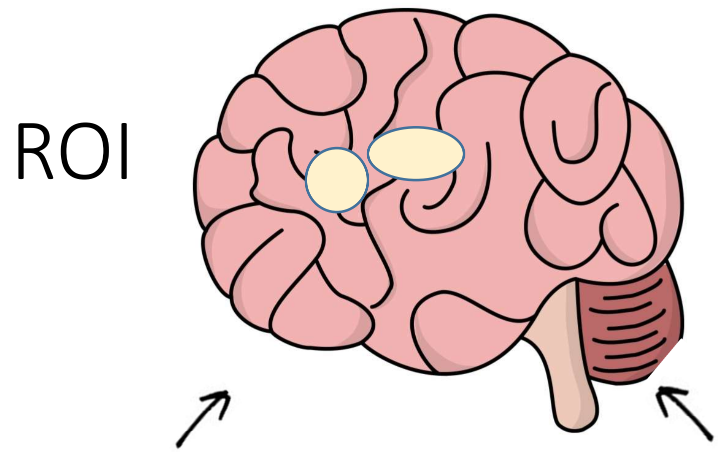
Does cognitive training work? There are numerous commercial training interventions claiming to improve general mental capacity; however, the scientific evidence for such claims is sparse. Nevertheless, there is accumulating evidence that certain cognitive interventions are effective. Here we provide evidence for the effectiveness of cognitive (often called "brain") training. However, we demonstrate that there are important individual differences that determine training and transfer. We trained elementary and middle school children by means of a videogame-like working memory task. We found that only children who considerably improved on the training task showed a performance increase on untrained fluid intelligence tasks. This improvement was larger than the improvement of a control group who trained on a knowledge-based task that did not engage working memory; further, this differential pattern remained intact even after a 3-mo hiatus from training. We conclude that cognitive training can be effective and long-lasting, but that there are limiting factors that must be considered to evaluate the effects of this training, one of which is individual differences in training performance. We propose that future research should not investigate whether cognitive training works, but rather should determine what training regimens and what training conditions result in the best transfer effects, investigate the underlying neural and cognitive mechanisms, and finally, investigate for whom cognitive training is most useful.

n-back training | training efficacy | long-term effects | motivation

Given the importance of WM capacity for scholastic achievement (14), even beyond its relationship to Gf (12, 15, 16), improving children's WM is of particular relevance. Although there is some promising recent research demonstrating that transfer of cognitive training is an obtainable goal (4–6), there is minimal evidence for training and transfer in typically developing school-aged children. Furthermore, whether there are long-term transfer effects is largely unknown. Our goal in this study was to adapt WM training interventions that have been found effective for adults (17, 18) to train children's WM skills with the aim of also improving their general cognitive abilities. We trained 62 children over a period of 1 mo (see Table 1 and *Materials and Methods* for demographic information). Participants in the experimental group trained on an adaptive spatial n-back task in which a series of stimuli was presented at different locations on the computer screen one at a time. The task of the participants was to decide whether a stimulus appeared at the same location as the one presented n items back in the sequence (Fig. 1) (17, 18). Participants in the active control group trained on a task that required answering general knowledge and vocabulary questions, thereby practicing skills related to Gc (Fig. S1) (7). Both training tasks were designed to be engaging by incorporating video game-like features and artistic graphics (19–22) (Fig. 2 and *Materials and Methods*). Before and after training, as well as 3 mo after completion of training, participants' performance was assessed on two different matrix reasoning tasks (23, 24) as a proxy for Gf. Because

ALFA
ANYSCALE LEARNING FOR ALL

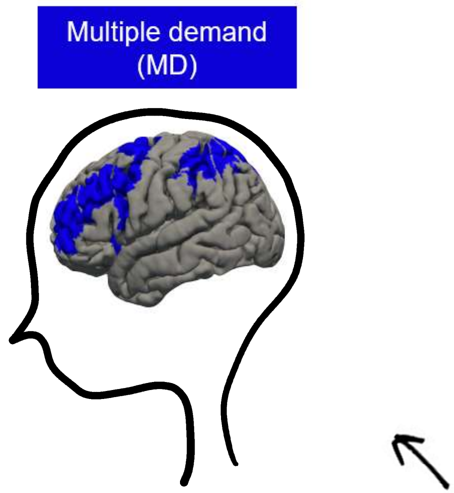
MIT CSAIL



Cognitive function 1

Cognitive function 2

ROI



Working memory

Code comprehension

fMRI is not the end-game though

Activity in the region $\Rightarrow$ Specialization

Activity in the region $\overset{?}{\Leftarrow}$ Specialization

Well then …

- PL is not processed as a language by our brains
- Results from such neuroimaging research need to be applied carefully
- Lots of interesting behavioral and patient studies waiting to be explored!

Thanks