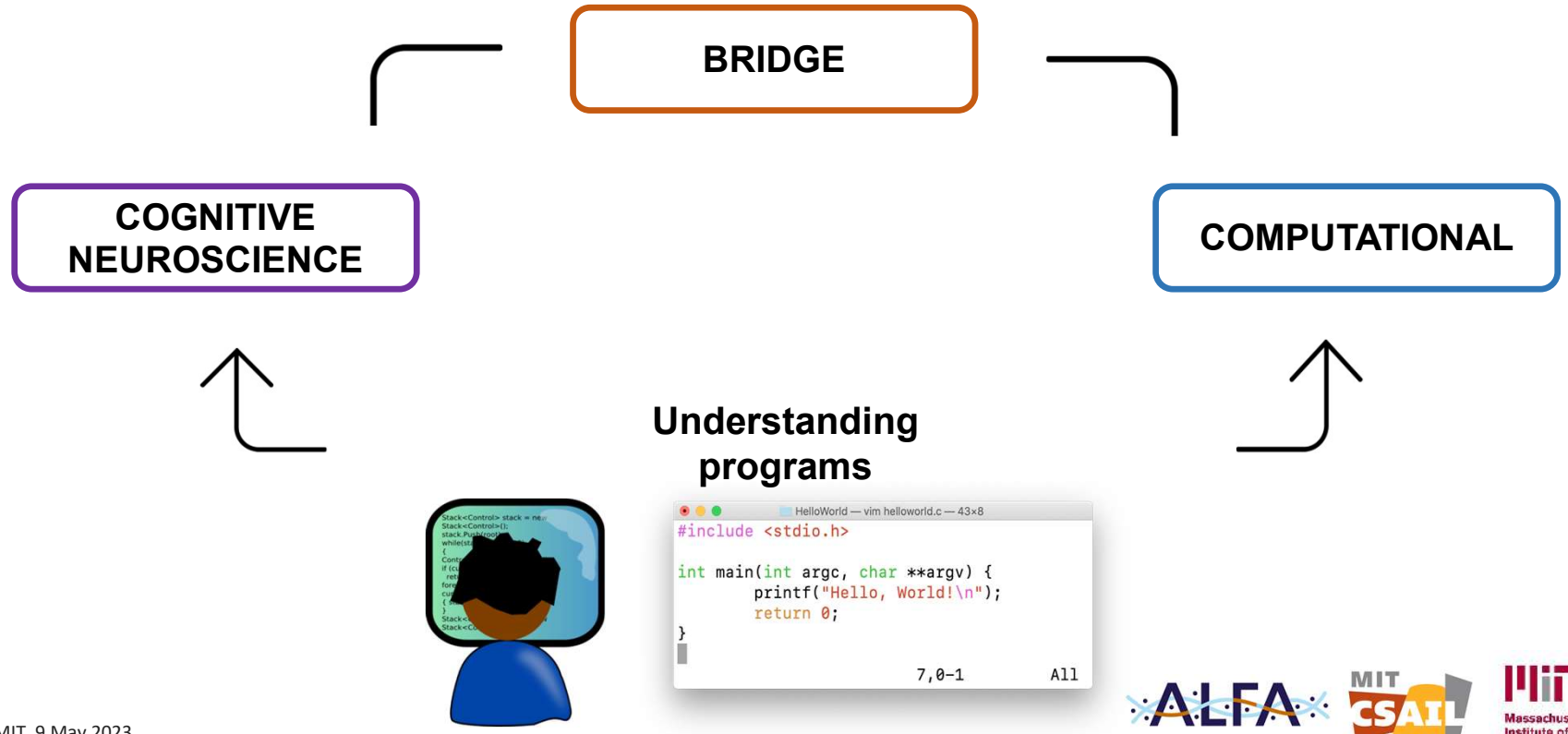
Understanding Computer Programs: Computational and Cognitive Perspectives

Shashank Srikant

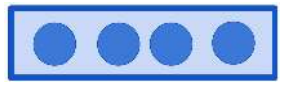
Ph.D. Thesis Defense

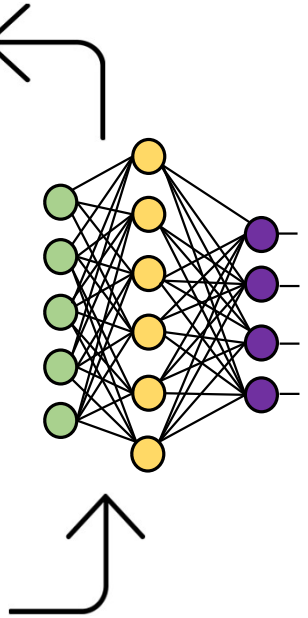
9 May 2023

Understanding Computer Programs: Computational and Cognitive Perspectives



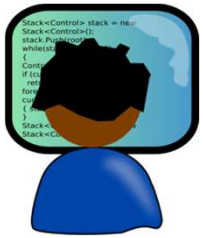
Model representation





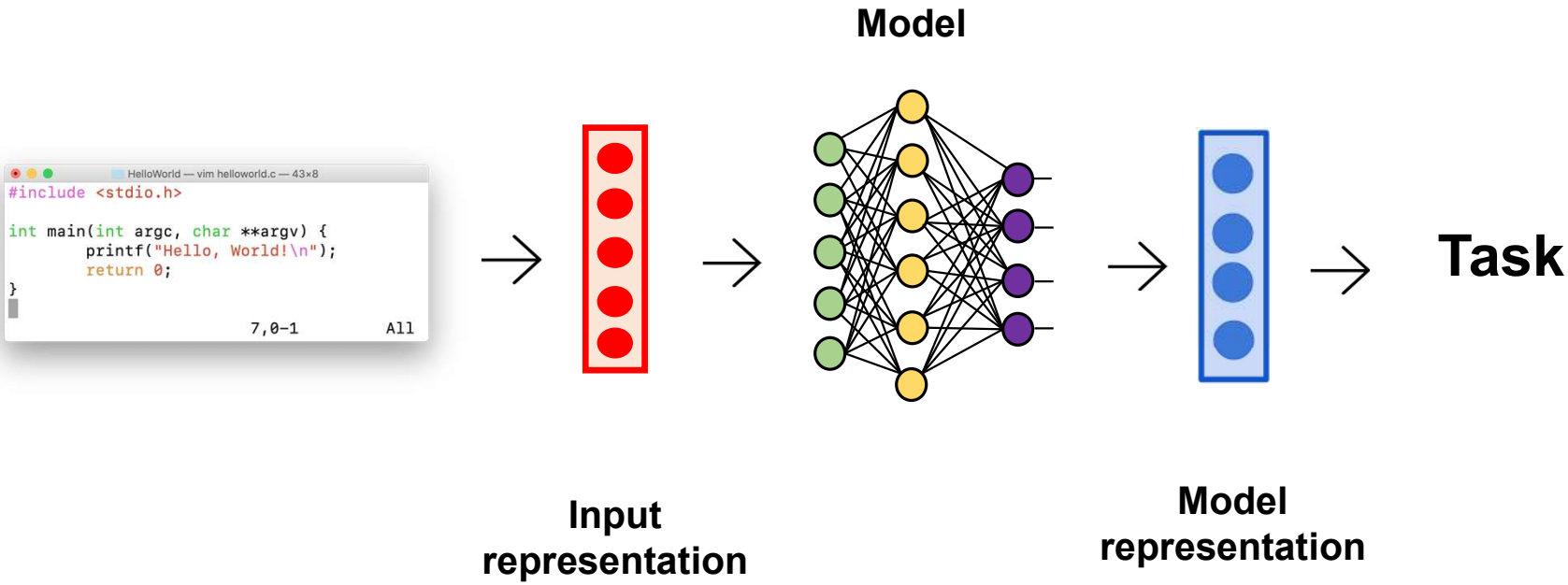
MACHINE LEARNING

Understanding programs



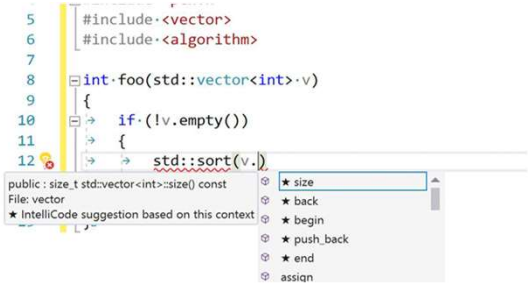
COMPUTATIONAL

Code models


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
Model
Input representation
Model representation
Task

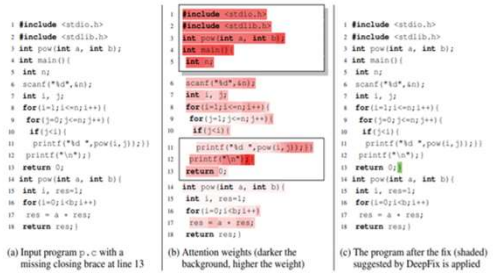
Code models

Code autocompletion and summarization



Improving variable names

Assignment grading and error fixing

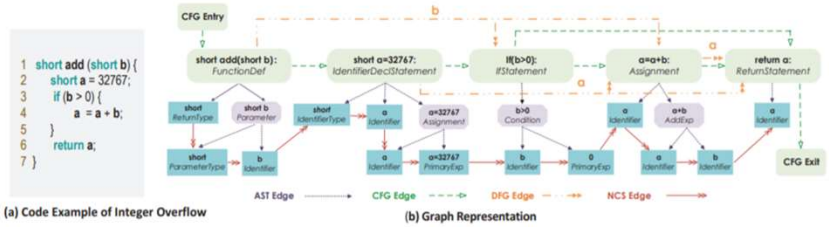


(a) Input program p.c with a missing closing brace at line 13

(b) Attention weights (darker the background, higher the weight)

(c) The program after the fix (shaded) suggested by DeepFix is applied

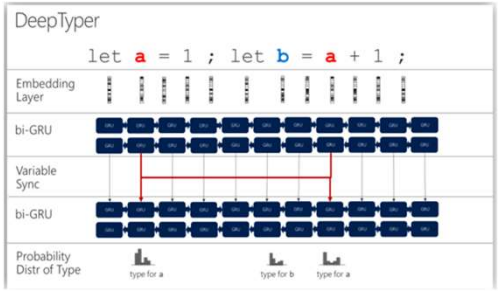
Vulnerability detection



(a) Code Example of Integer Overflow

(b) Graph Representation

Predicting types



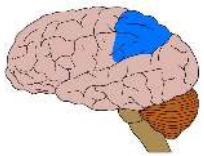
Raychev, V., Vechev, M., & Krause, A. (2015). **Predicting program properties from" big code"**. *ACM SIGPLAN Notices*.

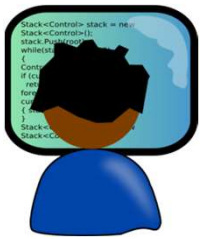
Zhou, Y., Liu, S., Siow, J., Du, X., & Liu, Y. **Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks**. NeurIPS 2019

Hellendoorn, V. J., Bird, C., Barr, E. T., & Allamanis, M. **Deep learning type inference**. *FSE. 2018.*

Gupta, Rahul, et al. "Deepfix: Fixing common c language errors by deep learning." *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

Allamanis, M., Brockschmidt, M., & Khademi, M. (2017). **Learning to represent programs with graphs**. *arXiv preprint arXiv:1711.00740.*

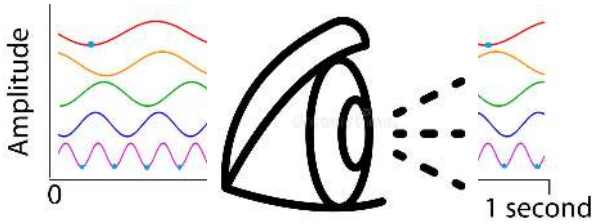




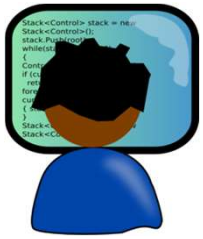
Neural activation

Pupil dilation

Gaze signals

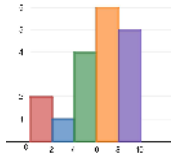


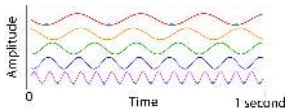
Understanding programs

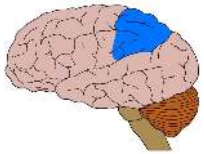


COGNITIVE NEUROSCIENCE

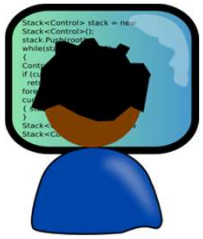
Behavioral, Neural representation





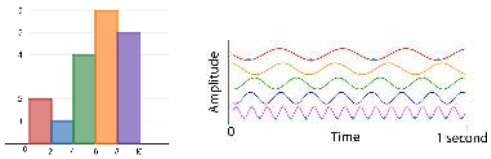


Understanding programs



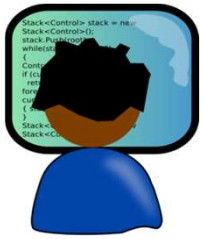
COGNITIVE NEUROSCIENCE

Behavioral, Neural representation



Model representation

Understanding programs



COGNITIVE NEUROSCIENCE

COMPUTATIONAL

Overview

COMPUTATIONAL

1. What is a test of a code model's basic understanding of code?

2. Can code models learn concurrent programs?

BRIDGE

5. What is "important" to programmers when reading code?

6. Can program/stimuli generation be optimized for cognitive behavior?

COGNITIVE NEUROSCIENCE

3. Which parts of our brains are involved in code comprehension?

4. Are program concepts encoded in the brain?

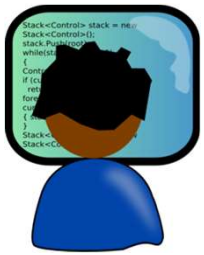
BRIDGE

COGNITIVE NEUROSCIENCE

Computational perspective

COMPUTATIONAL

Understanding programs



Question 1

What is a test of a code model's basic understanding of code?

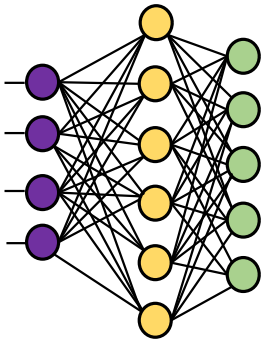
Humans

```
def remove_extras ( lst ) :
    new_list = [ ]
    for item in lst:
        if item not in new_list:
            new_list . append( item )
    return new_list
```

Humans

```
def remove_extras (ABC ) :
    new_list = [ ]
    for item in ABC :
        if item not in new_list:
            new_list . append( item )
    return new_list
```

Code models?

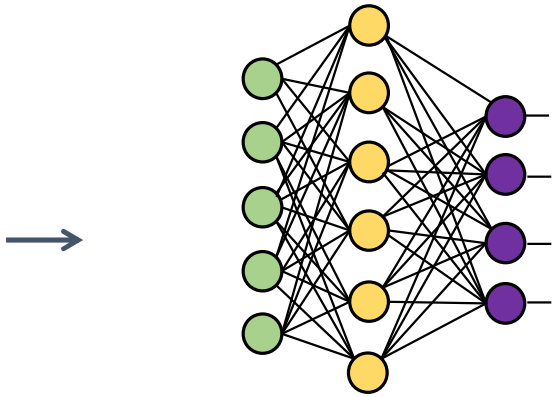


?

```
def remove_extras (ABC ) :
    new_list = [ ]
    for item in ABC :
        if item not in new_list:
            new_list . append( item )
    return new_list
```

A test for code models

```
def remove_extras ( ? ) :
    ? = [ ]
    for ? in ?:
        if ? not in ?:
            ?. append( ? )
    return ?
```



$$P_{perturbed} = \mathbf{z} \,.\, \mathbf{u} + (\mathbf{1} - \mathbf{z}) \,.\, P$$

where $\Sigma\, \mathbf{z} \leq k, \quad \mathbf{z} \in [0, 1]^n$

$\Sigma\, \mathbf{u}_i = 1, \quad \mathbf{u}_i \in [0, 1]^{|V|}$

Where to modify?

What to modify to?

GENERATING ADVERSARIAL COMPUTER PROGRAMS USING OPTIMIZED OBFUSCATIONS

Shashank Srikant[1] **Sijia Liu**[2,3] **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 [3]Michigan State University
shash@mit.edu, liusiji5@msu.edu, unamay@csail.mit.edu

ICLR 2021

Result

A code summarizer model *misunderstood* 30% of programs in which just one variable name was changed

Contribution

An optimizable code modifier to test the basics of code model understanding

Question 2

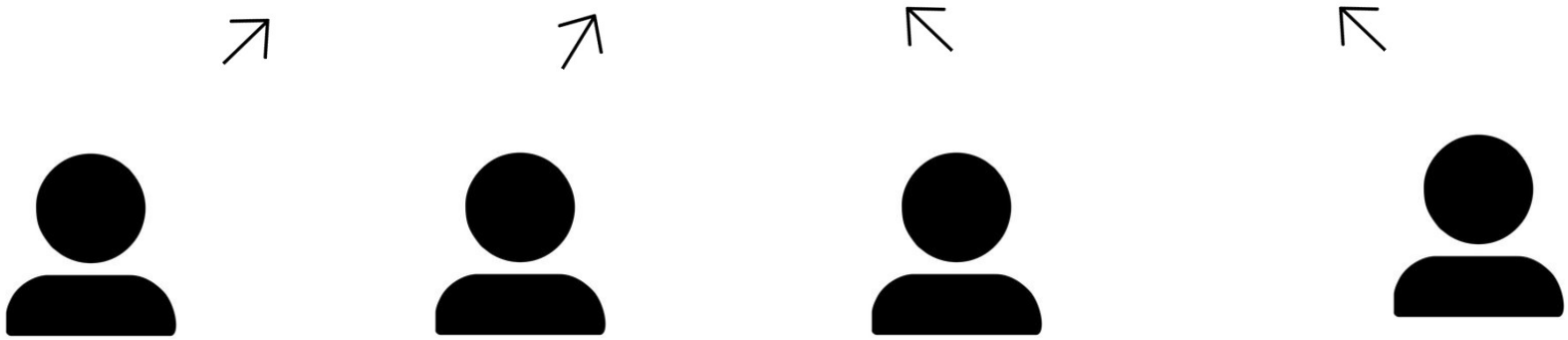
Can code models learn concurrent programs?


HOTEL

Time	Execution trace	
1s	read x	thread 1
2s	write x	thread 1
3s	read y	thread 2
4s	write y	thread 2

```
def book ( n ) :
    total = read ( )
    new = total - n
     write (new)
```

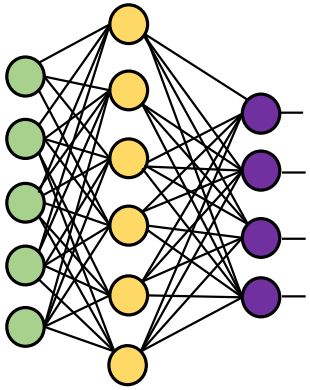
Data races

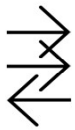


No annotated datasets!

```
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
```

Time	Execution trace	
1s	read x	thread 1
2s	write x	thread 1
3s	read y	thread 2
4s	write y	thread 2





Set of annotated execution traces

Idea

Generate this larger set

Set of annotated execution traces

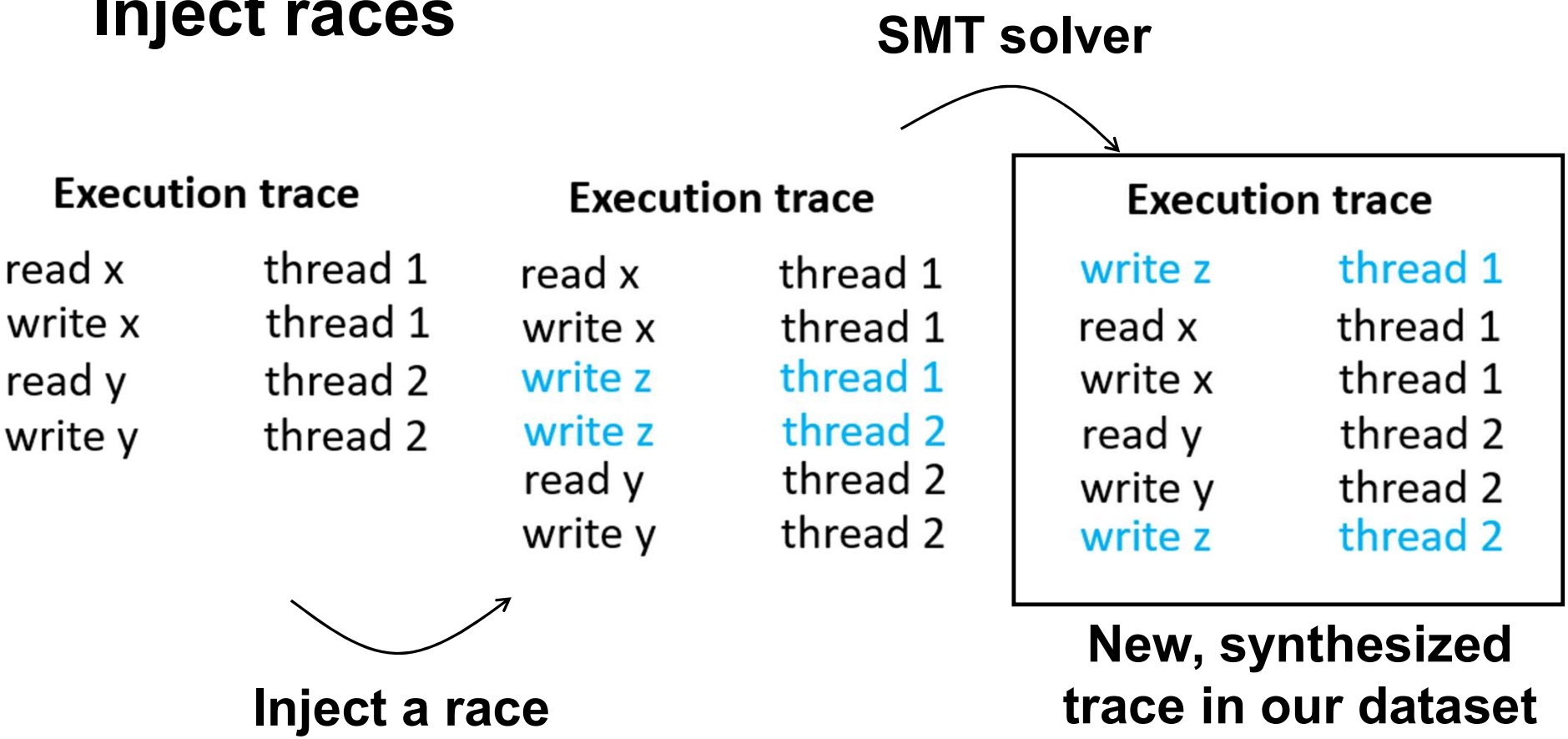
Idea

Generate this larger set

Source bugs from concurrent programs in production

Use established data race detectors to create a true-positive dataset

Inject races



RaceInjector: Injecting Races To Evaluate And Learn Dynamic Race Detection Algorithms

Michael Wang[1], Shashank Srikant[1,2], Malavika Samak[1], Una-May O'Reilly[1,2]
{mi27950, shash}@mit.edu {malavika, unamay}@csail.mit.edu
[1]CSAIL, MIT [2]MIT-IBM Watson AI Lab

SOAP 2023, PLDI

Result

Generated dataset contains examples which data race detection algorithms from the last 4 decades fail to detect

Contribution

First steps at modeling concurrent programs and data races

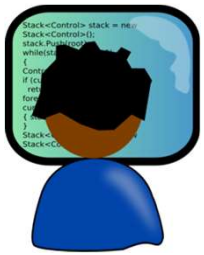
BRIDGE

Cog Neuro perspective

COGNITIVE NEUROSCIENCE

COMPUTATIONAL

Understanding programs

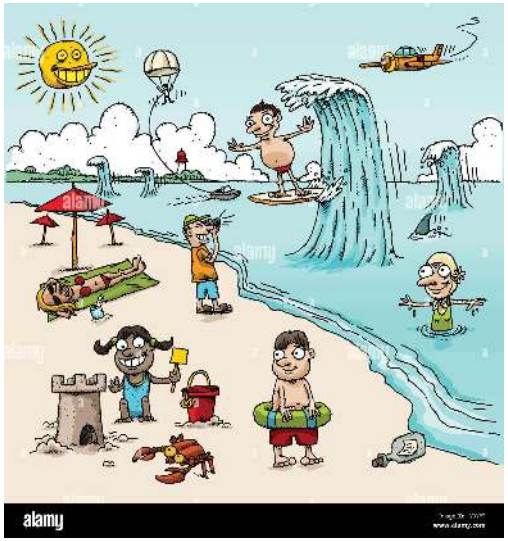


```
HelloWorld — vim helloworld.c — 43×8
#include <stdio.h>

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

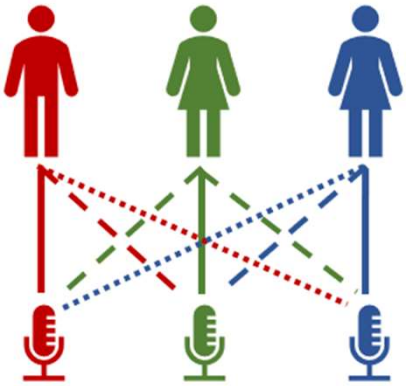
Question 3

Which parts of our brains are involved in code comprehension?

Human intelligence tasks



Describe the scene



Speaker identification

This one is better than the old one.

This one is better than nothing.

Sentiment analysis

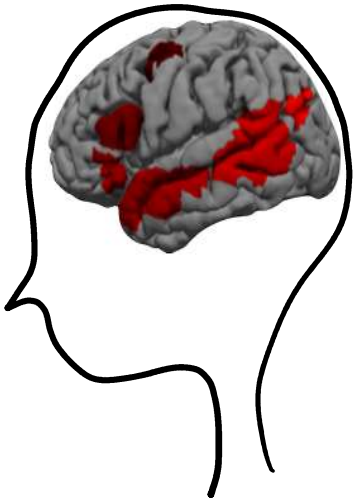
Human intelligence task?

```
def remove_extras ( lst ) :
    new_list = [ ]
    for item in lst:
        if item not in new_list:
            new_list . append( item )
    return new_list
```

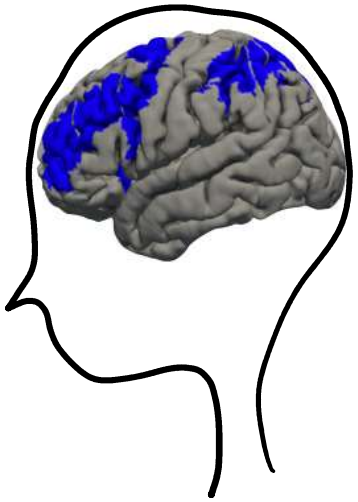
Responds strongly to language

Across different modalities – reading, writing, sign language

Language system



Multiple Demand system



Working memory

Logic

Math

General problem solving

```
group_data = [1, 5, 3]
num = 3

for value in group_data:
    product = num*value

print(product)
```

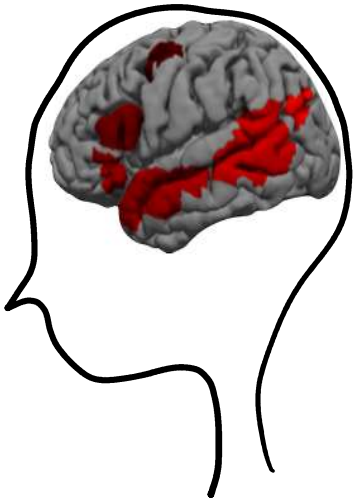
You have observations from three experiments: 1, 5, and 3.
You define a value 'product'.
Starting from the first observation, the value will get updated as the product of the observation and three.
After going through the three observations, what will the value be?

Result

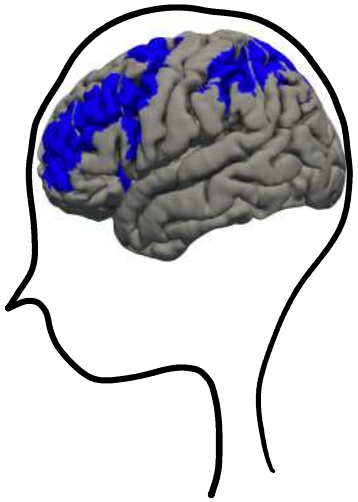
Responds strongly to language

Across different modalities – reading, writing, sign language

Language system



Multiple Demand system



Working memory

Logic

Math

General problem solving

Neuroscience

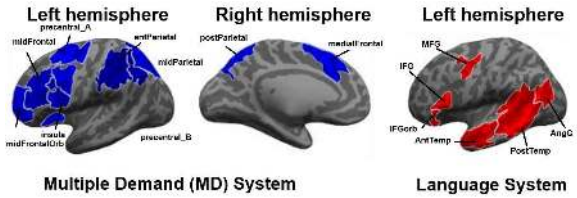
Comprehension of computer code relies primarily on domain-general executive brain regions

Anna A Ivanova, Shashank Srikant, Yotaro Sueoka, Hope H Kean, Riva Dhamala, Una-May O'Reilly, Marina U Bers, Evelina Fedorenko
Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, United States; McGovern Institute for Brain Research, Massachusetts Institute of Technology, United States; Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, United States; Eliot-Pearson Department of Child Study and Human Development, Tufts University, United States

eLife, 2020

Contribution

Experiment design to separate out responses from language sensitive brain regions vs. others

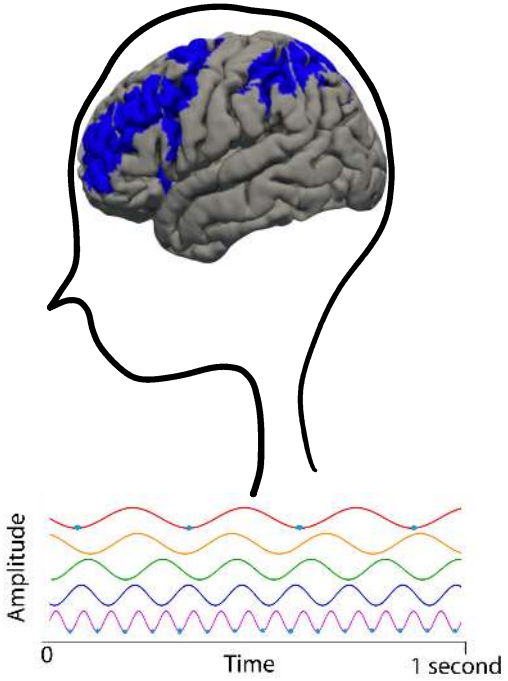


Question 4

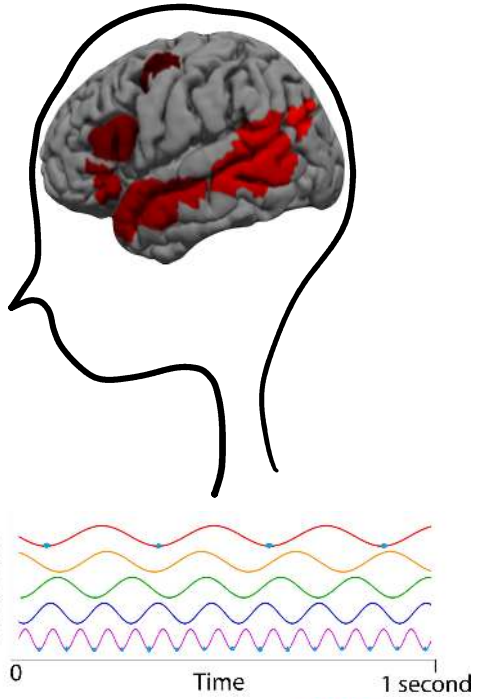
Are program concepts encoded in the brain?

```
group_data = [1, 5, 3]
num = 3

for value in group_data:
    product = num*value

print(product)
```

Multiple Demand system



Language system



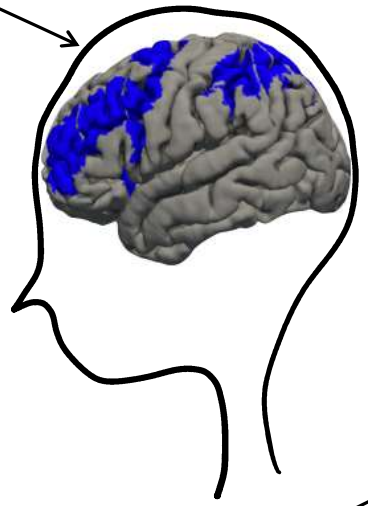
loops

branching

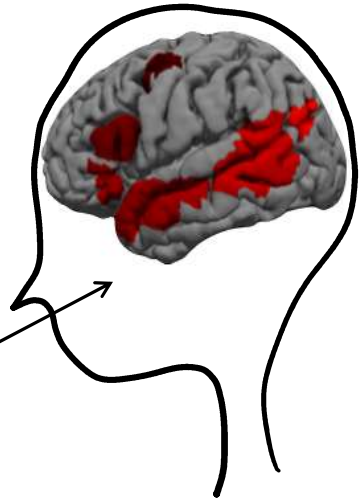
string-related operations

syntax, syntax tree

Multiple Demand system

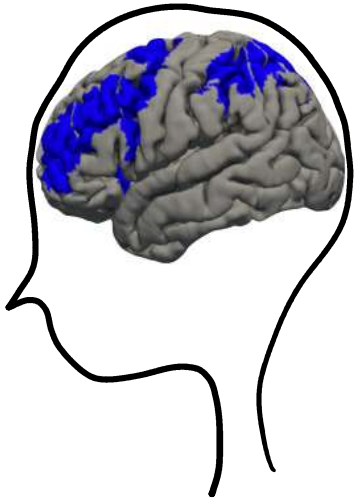


Language system

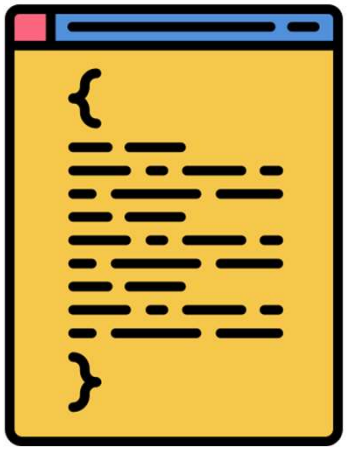


General problem solving

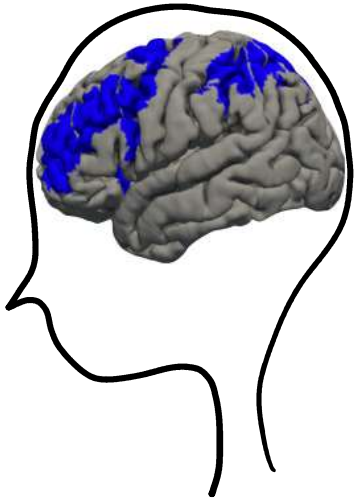
Multiple Demand system



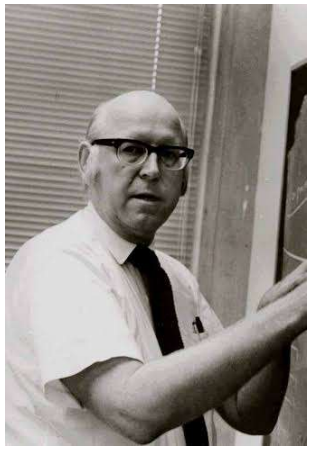
General problem solving



Multiple Demand system



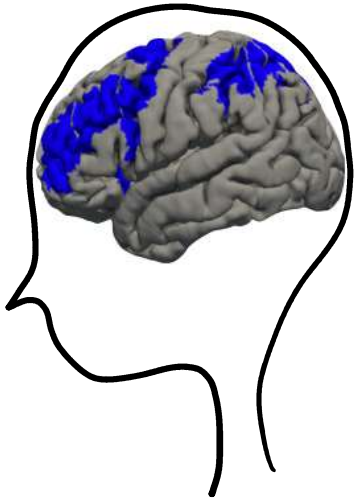
Logic Theorist - 1956



Allan Newell

Herbert Simon

Multiple Demand system



Result

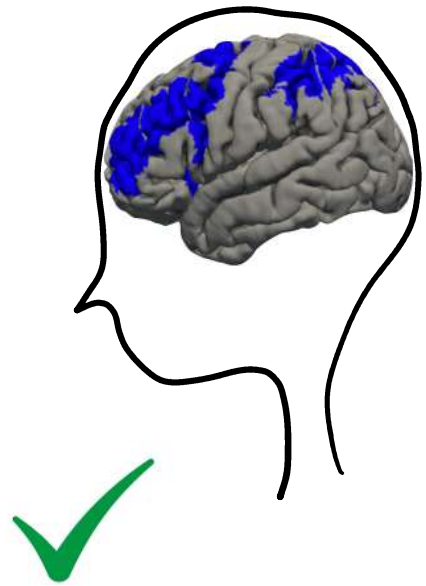
loops

branching

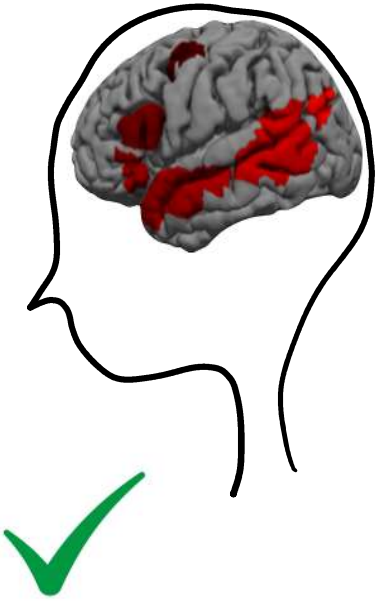
data dependencies

…

Multiple Demand system



Language system



Convergent Representations of Computer Programs in Human and Artificial Neural Networks

Shashank Srikant[*1,4] **Benjamin Lipkin**[*2] **Anna A. Ivanova**[1,2,3]
Evelina Fedorenko[2,3] **Una-May O'Reilly**[1,4]
* Equal contribution
[1]CSAIL, MIT [2] BCS, MIT
[3] McGovern Institute for Brain Research MIT-IBM Watson AI Lab
{shash, lipkinb, annaiv, evelina9}@mit.edu, unamay@csail.mit.edu

NeurIPS, 2022

Contribution

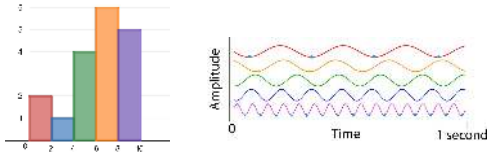
Decoding program concepts from the MD and LS

First steps at specifying the nature of stimuli the MD system responds to

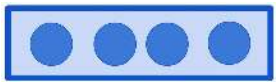
COGNITIVE NEUROSCIENCE

COMPUTATIONAL

Behavioral, Neural representation



Machine representation



Bridging the two perspectives

Question 5

What is “important” to programmers when reading code?

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-10, NO. 5, SEPTEMBER 1984 595

Empirical Studies of Programming Knowledge

ELLIOT SOLOWAY AND KATE EHRLICH

Int. J. Man-Machine Studies (1986) **25**, 697–709

Beacons in computer program comprehension

SUSAN WIEDENBECK

Delocalized Plans and Program Comprehension

A maintainer's understanding can go awry when it is based on purely local clues. How can we spell out the intentions behind a piece of coc

Int. J. Man-Machine Studies (1986) 25, 697–709

Beacons in computer program comprehension

SUSAN WIEDENBECK

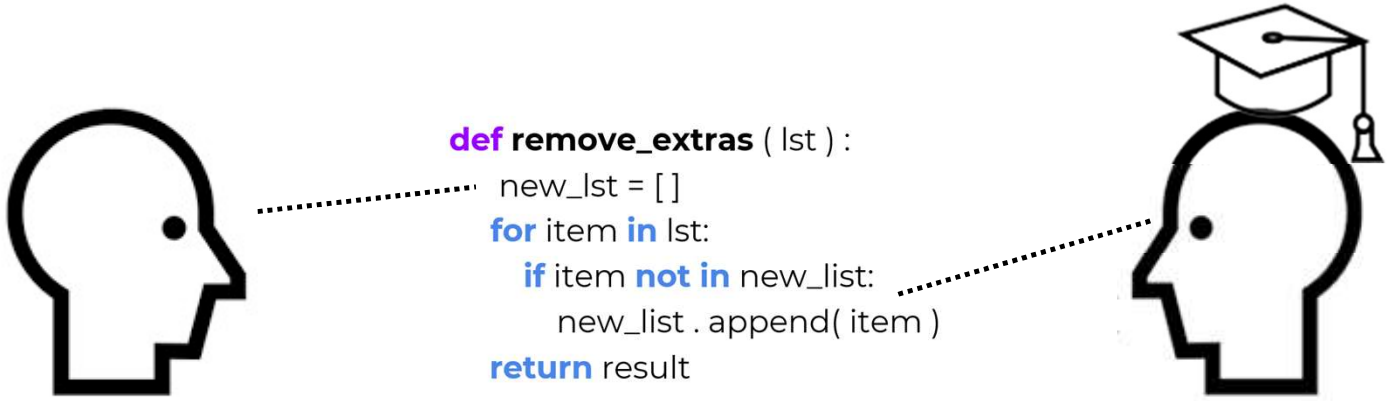
```
def remove_extras ( lst ) :
    new_list = [ ]
  for item in lst:
     if item not in new_list:
         new_list . append( item )
  return new_list
```

Non experts

Experts

Can automate the discovery of *beacons*?

Can *beacons* help us understand more about code comprehension?



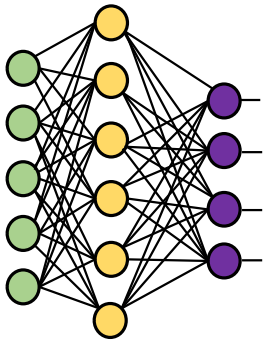
Non experts

Experts

Can automate the discovery of *beacons*?

Can *beacons* help us understand more about code comprehension?

```
def remove_extras ( lst ) :
    new_list = [ ]
    for item in lst:
        if item not in new_list:
            new_list . append( item )
    return new_list
```



Non experts

Experts

Experts

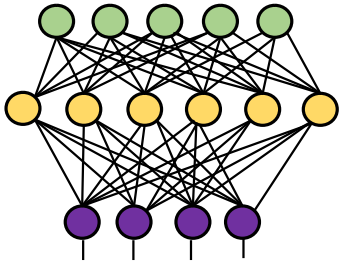
```
def remove_extras ( lst ) :
    new_list = [ ]
    for item in lst:
        if item not in new_list:
            new_list . append( item )
    return new_list
```

new_list . append(item)

for item in lst:

new_list . append(item)

```
def remove_extras ( lst ) :
    new_list = [ ]
    for item in lst:
        if item not in new_list:
            new_list . append( item )
    return new_list
```



new_list . append(item)

for item in lst:

new_list . append(item)

?

Results

Human experts agree when identifying *beacons* (r = 0.5)

Model representations can predict these *beacons* (r = 0.7)

Contribution

Demonstrates how to use code models as proxies of experts.

Helps understand our behavioral responses to code

Question 6

Can program/stimuli generation be optimized for cognitive behavior?

Hypothetical

Model which can predict *beacons*

Model which can predict *mental load*

Can I generate programs which

minimize # of beacons?

minimize mental load?

Model which can predict *beacons*

Model which can predict *confusion*

Can I generate programs which

minimize # of beacons?

minimize mental load?

Yes!

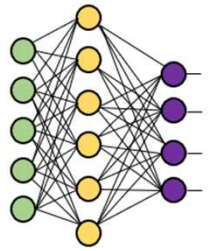
Model which can predict *beacons*

Model which can predict *confusion*

COMPUTATIONAL

What is a test of a code model's basic understanding of code?

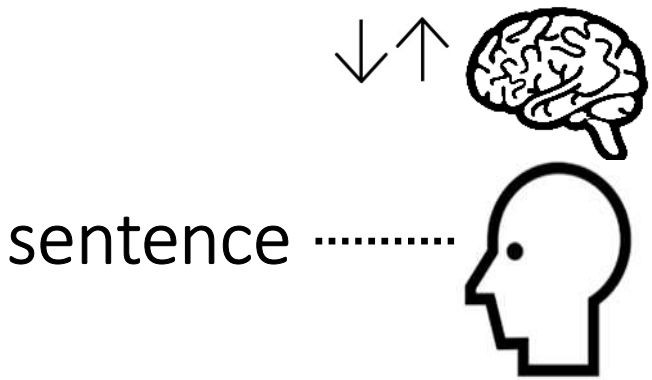
```
def remove_extras ( ? ) :
    ? = [ ]
    for ? in ?:
        if ? not in ?:
            ?. append( ? )
    return ?
```



Model which can predict *beacons*

Model which can predict *confusion*

Generate sentences that control brain responses!



GOLI: Goal-Optimized Linguistic Stimuli for Psycholinguistics and Cognitive Neuroscience

Shashank Srikant[1,2] **Greta Tuckute**[3] **Sijia Liu**[2,4] **Una-May O'Reilly**[1,2]
[1]CSAIL, MIT [2]MIT-IBM Watson AI Lab BCS, MIT [4]Michigan State University
{shash, gretatu}@mit.edu, liusiji5@msu.edu, unamay@csail.mit.edu

Driving and suppressing the human language network using large language models

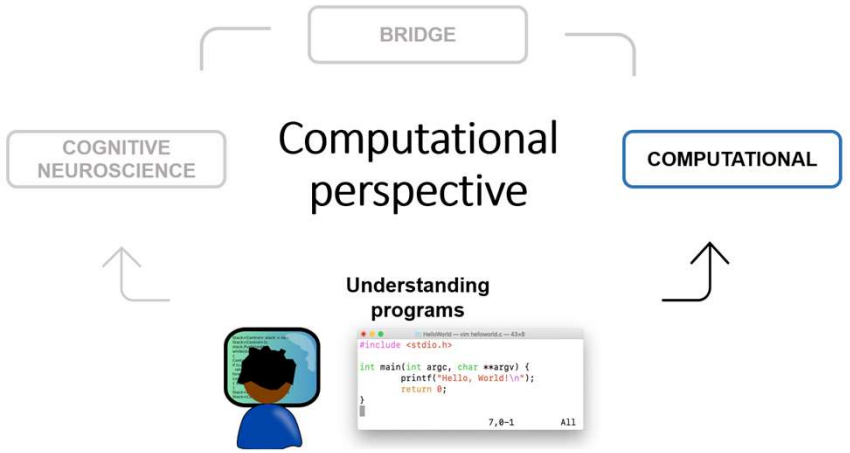
Greta Tuckute[1,2], Aalok Sathe[1,2], Shashank Srikant[3,4], Maya Taliaferro[1,2], Mingye Wang[1,2], Martin Schrimpf[5], Kendrick Kay[6], Evelina Fedorenko[1,2,7]

In submission

Contributions: Big picture

Methods to test and improve code understanding in code models

First steps towards training code models to understand concurrent program behavior

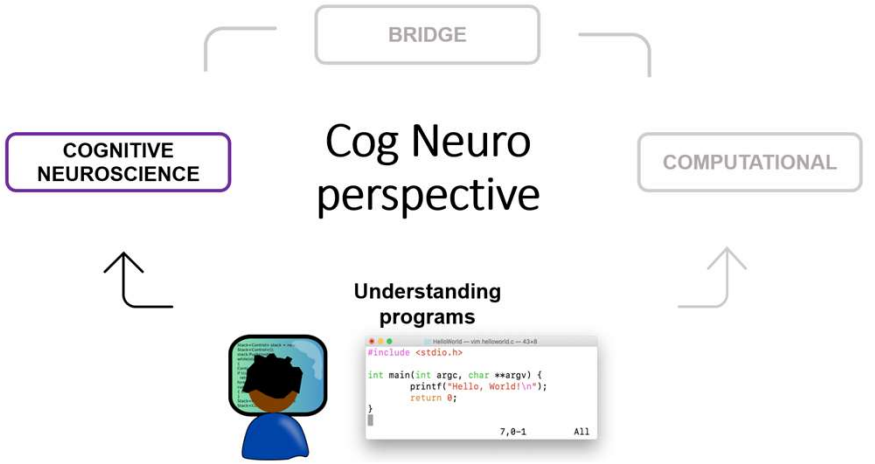


Contributions: Big picture

Better understanding of where code comprehension happens in the brain

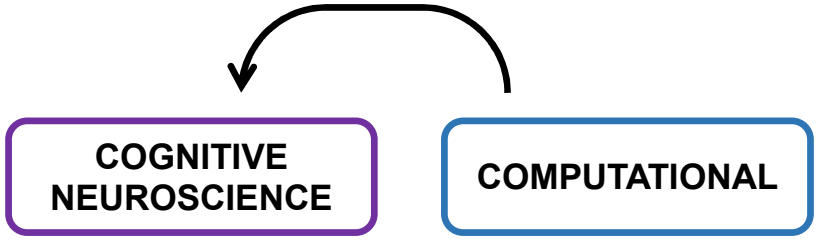
Better understanding of what code properties are encoded in the brain

First steps at specifying the nature of stimuli the MD system responds to

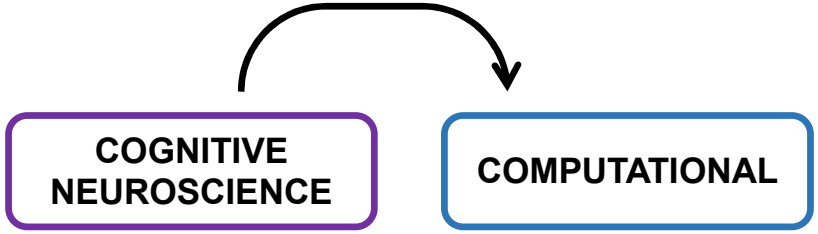


Contributions: Big picture

Using code models to better understand behavioral responses to code comprehension



Using models of human behavior to generate code



Where to from here?

Probing code models for the concepts they acquire

Integrating the role of the MD system in model architecture design?

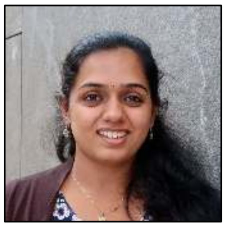
Where to from here?

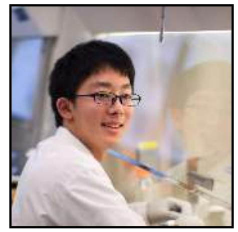
Other behavioral responses when understanding code.
Models to explain these behaviors

What happens we write code? Debug code?

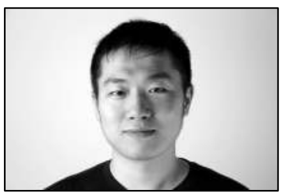
The A-team













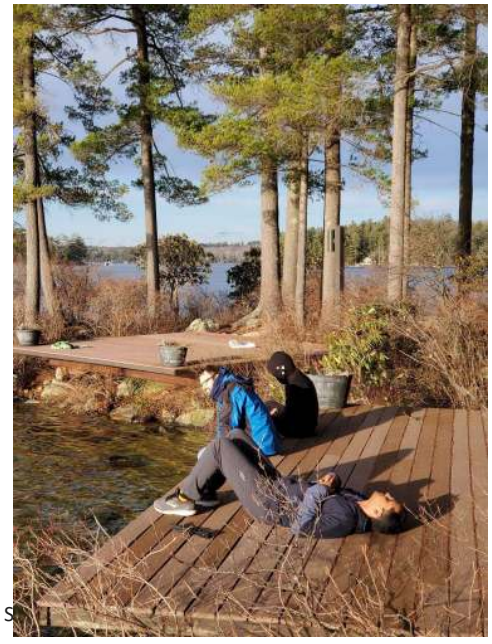


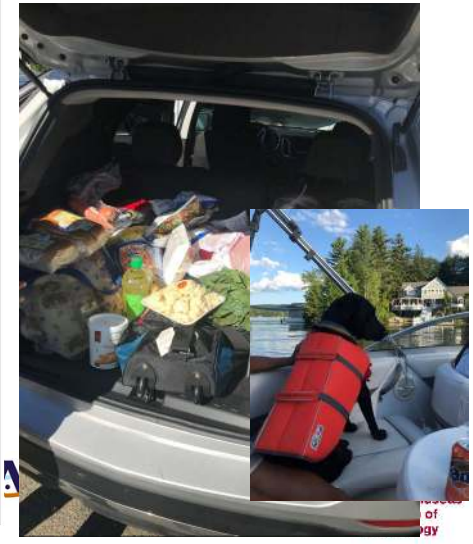

MIT-IBM Watson AI Lab

MIT CSAIL FinTech Initiative

sonatype

ALFA MIT CSAIL Massachusetts Institute of Technology

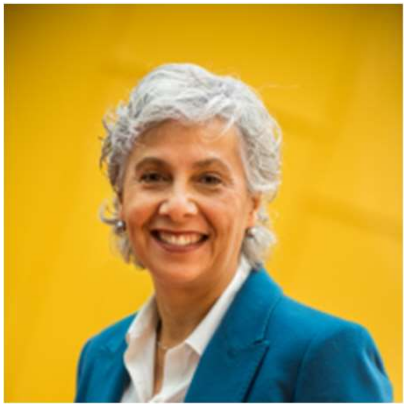




My letter writers

Family

Friends



Evolution as a design strategy for nonlinear architecture : Generative modeling of 3-D surfaces

Una-May O'Reilly *
Artificial Intelligence Lab
MIT
545 Tech Sq
Cambridge, MA, 02143
unamay@ai.mit.edu

Girish Ramachandran
School of Architecture
Massachusetts Institute of Technology
77 Mass Avenue
Cambridge, MA 02139
girish@mit.edu

Meta Optimization: Improving Compiler Heuristics with Machine Learning

Mark Stephenson and Saman Amarasinghe
Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge, MA 02139
{mstephen, saman}@cag.lcs.mit.edu

Martin Martin and Una-May O'Reilly
Massachusetts Institute of Technology
Artificial Intelligence Laboratory
Cambridge, MA 02139
{mcm, unamay}@ai.mit.edu

Genetic Programming Applied to Compiler Heuristic Optimization

Mark Stephenson[1], Una-May O'Reilly[2],
Martin C. Martin[2], and Saman Amarasinghe[1]

Autotuning Algorithmic Choice for Input Sensitivity

Yufei Ding*, Jason Ansel◇, Kalyan Veeramachaneni◇, Xipeng Shen*
Una-May O'Reilly◇, Saman Amarasinghe◇

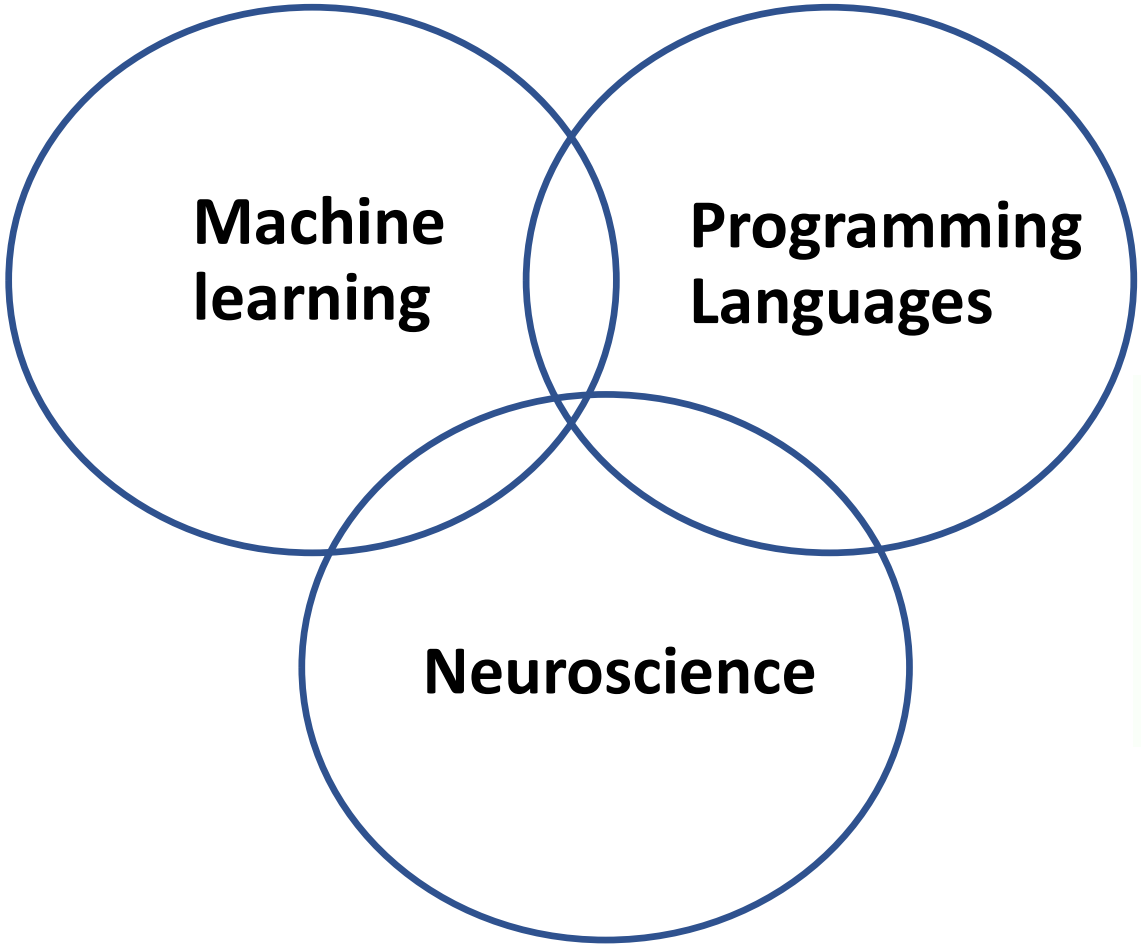
ALFA MIT CSAIL Massachusetts Institute of Technology

MIT developing first humanoid personal assistant

November 4, 2003

Una -May O'Reilly, co-principal investigator for MIT's newest robot, watches as Cardea lets itself out the door of the lab to go for a stroll down the hall.

Shashank Srika


Machine learning
Programming Languages
Neuroscience

Questions

COMPUTATIONAL

1. What is a test of a code model's basic understanding of code?

2. Can code models learn concurrent programs?

BRIDGE

5. What is "important" to programmers when reading code?

6. Can program/stimuli generation be optimized for cognitive behavior?

COGNITIVE NEUROSCIENCE

3. Which parts of our brains are involved in code comprehension?

4. Are program concepts encoded in the brain?