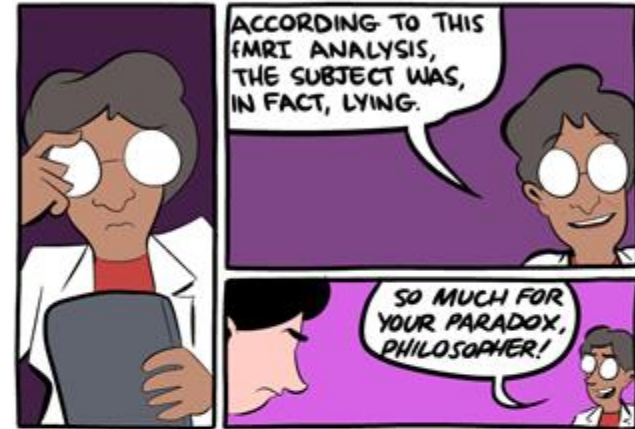# Code Inspection and the Brain

The Story So Far …

- We want to deliver and support a quality software product

- We have covered many process and technical aspects (measurement, testing, static analysis, code inspection, design patterns, etc.)

- And we have focused on many human biases and weaknesses

- But we have not considered human expertise and productivity
  - SE is performed by humans: how do humans (our brains) work?

One-Slide Summary

- We can investigate neural correlates of software engineering activities using **medical imaging**.

- **Top-down comprehension** based on semantic cues is more efficient (easier) than bottom-up comprehension.

- Neural representations of programming and natural languages are **distinct**. Classifiers can distinguish them based solely on brain activity. The same brain locations distinguish different programming tasks. Greater **skill** accompanies a less-differentiated neural representation.

Code Review and Comprehension

- Developers spend more time understanding and comprehending code than any other activity
  - NASA: understanding > correctness for reuse
- Code review is a de facto standard
  - "Should we accept this commented patch?"
  - Mandated in Facebook, Google, etc.
  - One of the most effective techniques in software development

Previously: Readability Model

- We considered a descriptive readability metric
  - Blank lines are good?
  - Identifier length had no effect?
  - Identifier content entirely unexplored?

- Produced an award-winning metric
  - But still somewhat unsatisfactory

- Why not just ask programmers which features make code easy to read?

Why Use Medical Imaging?

- Unreliable Self-Reporting

- Inform Pedagogy

- Understand Expertise

- Retrain Aging Humans

- Guide Technology Transfer

- Fundamental Understanding

Self-Reporting is Unreliable

- Economics: **revealed choice** (e.g., people say they would not support Wal-Mart but shop there anyway)

- Computer Science: 3 of top 4 features that people self-report (e.g., shorter functions) as making code maintainable are irrelevant

- Psychology: "A review of 55 studies in which self-evaluations of ability were compared with measures of performance showed a low mean validity coefficient (mean r=.29) with high variability (SD=0.25)."

- [Mabe and West. *Validity of self-evaluation of ability: a review and meta-anslysis.* J. Applied Psych.]

Pedagogy and Expertise

- Informal model: your mind is a computer, with the brain and neurons as hardware and your memories as software

- When you learn something new, is that just new "software", or does mental "hardware" change?

- Answer: both! We can observe "learning" by looking at physical brain structures.

- [Bassett et al. *Dynamic reconfiguration of human brain networks during learning.* PNAS.]
  [Ritchey et al. *Functional connectivity relationships predict similarities in task activation and pattern information during associative memory encoding.* J. Cognitive Neuroscience.]

Retraining Aging Engineers

- Older workers are retraining into engineering and companies are hiring older engineers

  - [J. Wright. *In-demand and aging: A look at engineers and engineering technicians in the workforce.*]

- Older humans show more diffuse patterns of neural activity, recruiting nearby parts of the brain to help

  - "The occipital reduction is consistent with the view that sensory processing decline is a common cause in cognitive aging, and the prefrontal increase may reflect **functional compensation**."

  - [Cabeza et al. *Task-independent and task-specific age effects on brain activity during working memory, visual attention and episodic retrieval.* Cereb. Cortex.]
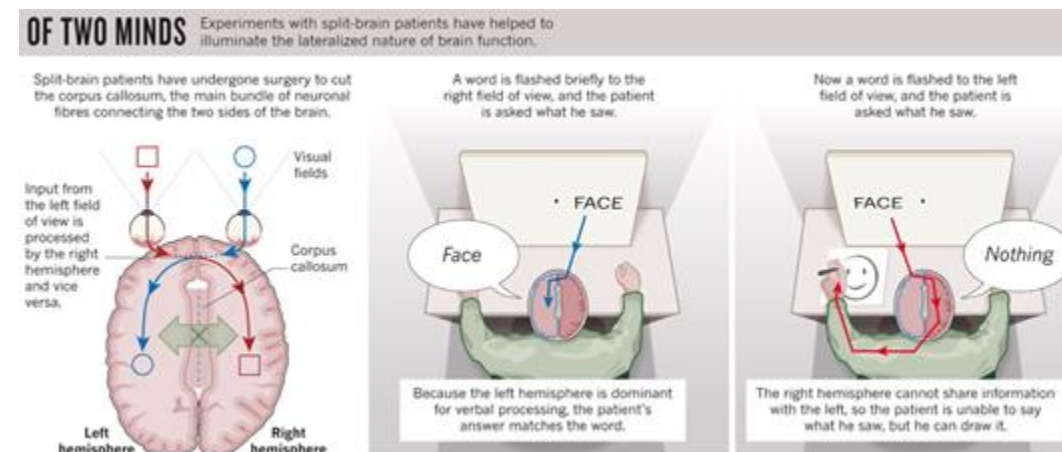
Guiding Technology Transfer

- **Technology transfer** involves turning a research idea (like a dynamic analysis) into an effective product that is actually used

- Fault localization can produce ranked lists and false positives that humans dislike (and don't use) [Parnin and Orso reading]

- Humans and tools can disagree on what is a false positive [Bessey et al. reading]

- We need better models so that tools can produce results that humans like, because human champions are essential
  - "A key part of technology transfer between research and development organizations is to have champions …" [Ball et al. *SLAM and static driver verifier: technology transfer of formal methods inside Microsoft.* Integrated Formal Methods.]

Examining the Brain
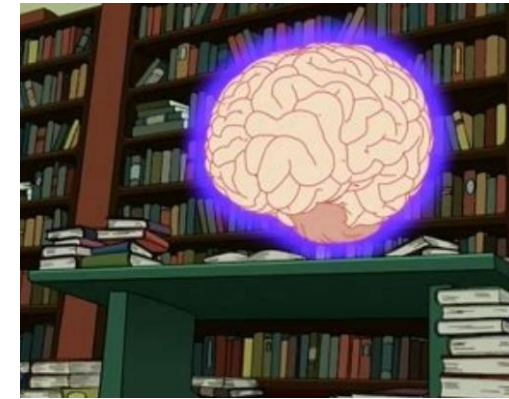
- In psychology, many neural studies involve patients with epilepsy who have the corpus calossum severed to treat seizures
  - "Since you've already cut the skull open, while you're in there, put in this electrode …"
- We need a **non-invasive** way to observe what is happening inside the human brain

OF TWO MINDS — Experiments with split-brain patients have helped to illuminate the lateralized nature of brain function.

Split-brain patients have undergone surgery to cut the corpus callosum, the main bundle of neuronal fibres connecting the two sides of the brain.

Input from the left field of view is processed by the right hemisphere and vice versa.

A word is flashed briefly to the right field of view, and the patient is asked what he saw.

Because the left hemisphere is dominant for verbal processing, the patient's answer matches the word.

Now a word is flashed to the left field of view, and the patient is asked what he saw.

The right hemisphere cannot share information with the left, so the patient is unable to say what he saw, but he can draw it.

Feed me, Seymour!

- Your brain uses energy to carry out activities

- Energy must be transported in to the brain

- This is done via **oxygen** in the blood

- If we could only tell oxygen-rich blood from oxygen-poor blood, we could tell which part of the brain is using energy and is thus active!
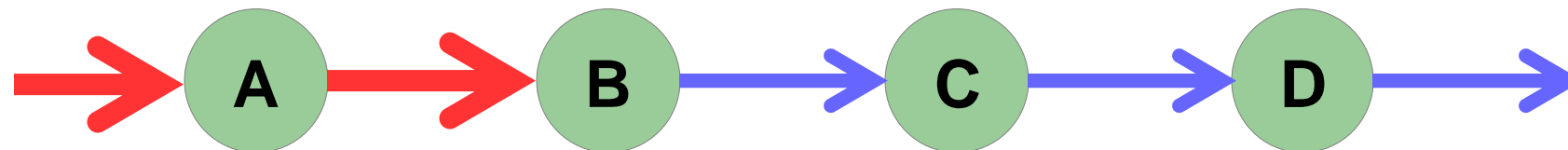
The BOLD and the Beautiful

- Oxygen-rich and oxygen-poor blood have different electromagnetic properties

- That is, they respond differently to certain powerful magnetic fields
  - Can be detected by a **magnetic resonance** (**MR**) scanner
  - Note: powerful magnets do not hurt humans

- The **blood-oxygen level dependent** (**BOLD**) response is the ratio of oxygenated to deoxygenated hemoglobin

fMRI

- **Functional magnetic resonance imaging** (**fMRI**) is a non-invasive technique for probing the neurobiological substrates of various cognitive functions *in vivo* by measuring the BOLD signal
  - Millimeter scale (>> EEG or PET, etc.)
  - It is a relatively recent technique; pioneering uses are associated with psychology
  - Especially in CS: first research using fMRI to investigate software engineering was in 2014
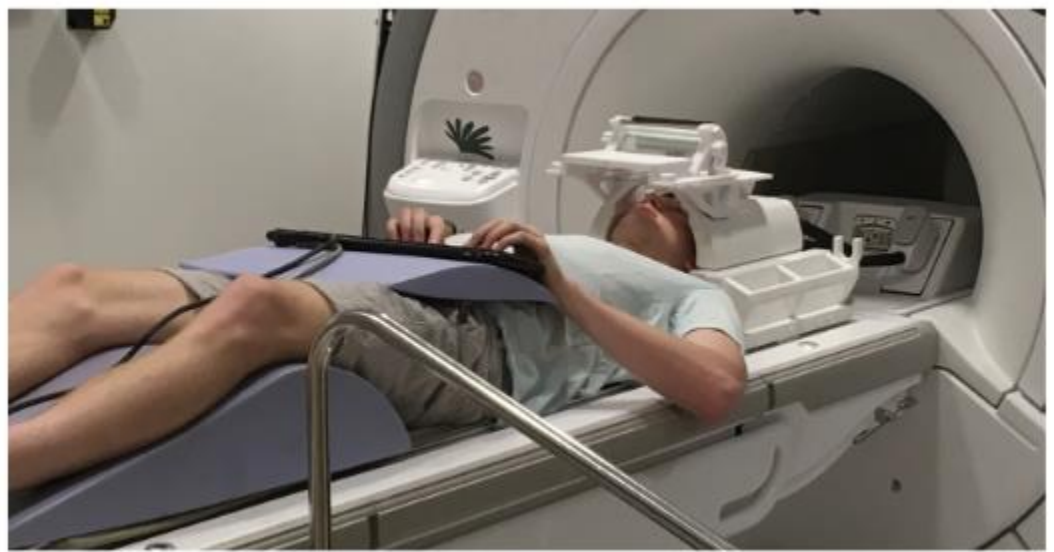
A Study in Contrasts

- A subject might be doing multiple things
  - Reading code, breathing, being nervous
- How can we tell if an observed pattern of activation corresponds to one action?

- Experimental design and control
  - A = "reading code + breathing + … "
  - B = "writing code + breathing + … "
- The contrast A-B shows patterns of brain activation that *vary* between the stimuli/tasks

High-Level Question #1

What actually makes it easier to read code?

One Model of Code Comprehension

- **Top-down comprehension** refers to cognitive processes in which experience and expectation and semantic cues (beacons) guide the understanding of source code

- **Plans** are knowledge structures representing semantic and syntactic software patterns
  - Example: the identifier **bubbleSort** encourages (primes) programmers to expect elements of that algorithm, such as array element swaps

Another Model of Code Comprehension

- **Bottom-up comprehension** refers to cognitive processes in which meaning is obtained from every individual statement and then synthesized into a holistic understanding

- Programmers may hold these elements in working memory and then abstract those pieces of information into higher-order concepts: this is called **semantic chunking**

Which is Correct?

- Researchers have debated and theorized

- Perhaps …
  - You mostly use bottom-up comprehension because meaning must be extracted from perceptual and syntactic information
  - You mostly avoid bottom-up comprehension because it is tedious

- Let's find out!

- [Siegmund et al. *Measuring Neural Efficiency of Program Comprehension*. ESEC/FSE.]

Neural Efficiency

- **Neural efficiency** is the phenomenon where lower brain activation indicates that a cognitive process is more efficient and thereby perceived as easier

- Reference last lecture on productivity!

- Informally, one difference between experts and novices is that experts use less energy than novices to solve the same task
  - The task feels easier to the expert
  - Experts literally do not have to work as hard

Trivia: Culture

- Although the tooth fairy is a common fixture in the United States and other Western countries, there are some pretty fun traditions around the world.

- What form does the tooth fairy take in Mexico?
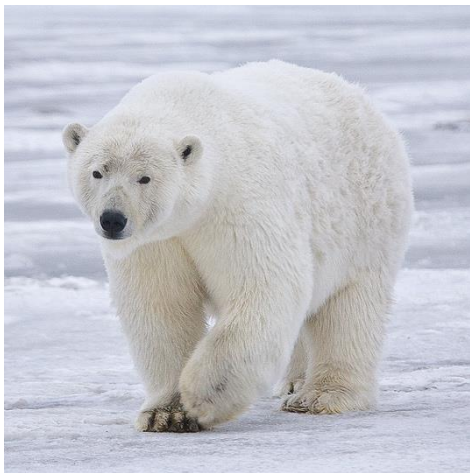
A

B

C

D

Trivia: Video Games

- This charming game was released in 2006 for the Xbox 360 and quickly won fans with its colorful, imaginative world where players tend to a garden filled with piñata creatures. Players could then breed and evolve their piñatas. Each piñata species had a fun name that combined an animal with a type of food like Doenut (Deer and Doughnut), S'morepion (Scorpion and S'mores), or Raisant (Ant and Raisin).

Trivia: Westley Weimer Fun Facts

- Professor Westley Weimer has a lot of fun facts about himself he loves mentioning and talking about (e.g., Did he write a romance novel? Have you found his gaming mods? Have you asked him about Magic the Gathering? What about fencing?)

- With that in mind, what kind of phone do you think he owns?
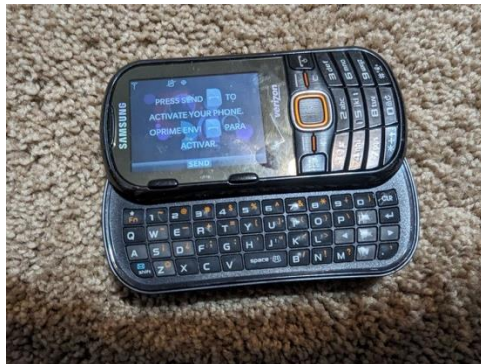
None of the above, because he doesn't own a phone.

A

B

C

D

Experimental Setup 1
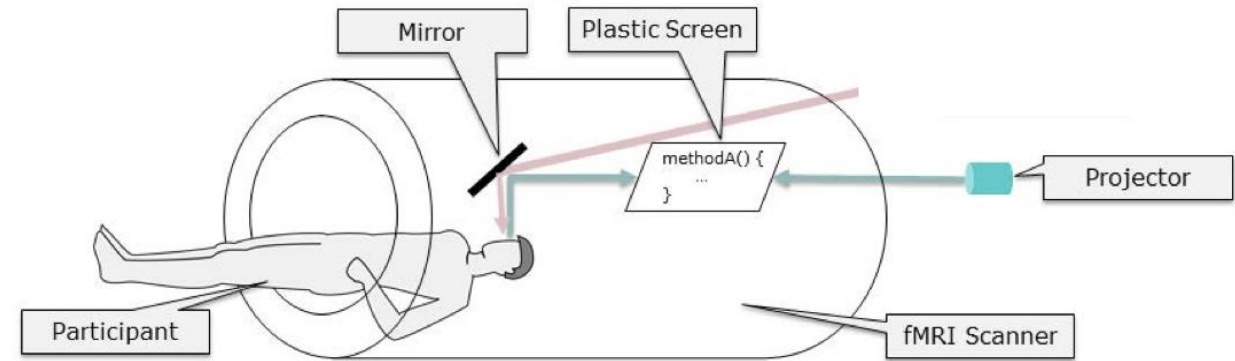
- Start with code snippets
  - 8-19 lines, 20-30 seconds to read, etc.
- Participants look at code and determine if it implements the same thing as a snippet shown earlier during a pre-training session
  - No scrolling, no typing, etc.
- N=11 student participants

Experimental Setup 2

- Two-way controlled experiment
  - **Top-down** comprehension
  - **Bottom-up** comprehension

- Four-way controlled experiment
  - Code with beacons and pretty-printing
  - Code with beacons and disrupted layout
  - Code without beacons but with pretty-printing
  - Code without beacons but with disrupted layout

Example Stimuli

Listing 1: Code snippet with beacons and pretty-printed layout (BY, LP)

```
1   public float arrayAverage(int[] array) {
2       int counter = 0;
3       int sum = 0;
4
5       while (counter < array.length) {
6           sum = sum + array[counter];
7           counter = counter + 1;
8       }
9
10      float average = sum / (float) counter;
11      return average;
12  }
```

Listing 2: Code snippet with no beacons and disrupted layout (BN, LD)

```
1   public float ayyaoAwyyaky(int[] array) {
2       int
3               mgqakyy
4       = 0;
5       int sum = 0;
6
7       while (mgqakyy
8               < array.length) {
9           sum =
10          sum + array[mgqakyy];
11          mgqakyy
12          = mgqakyy + 1;
13      }
14
15      float average
16              = sum /
17              (float) mgqakyy;
18      return
19              average;
20  }
```

Bottom-up: obfuscate identifier names so that they show usage but not meaning
Top-down: beacons like method names and layout

Result 1

- What is the difference between bottom-up program comprehension and comprehension via semantic cues?

- Brain area 39 in both hemispheres deactivates during comprehension based on semantic cues and activates during bottom-up comprehension

- Comprehension based on **semantic cues requires less cognitive effort** than bottom-up comprehension:
  - Response times are identical
 - Energy use "across the board" is lower

Same time

Easier for your brain
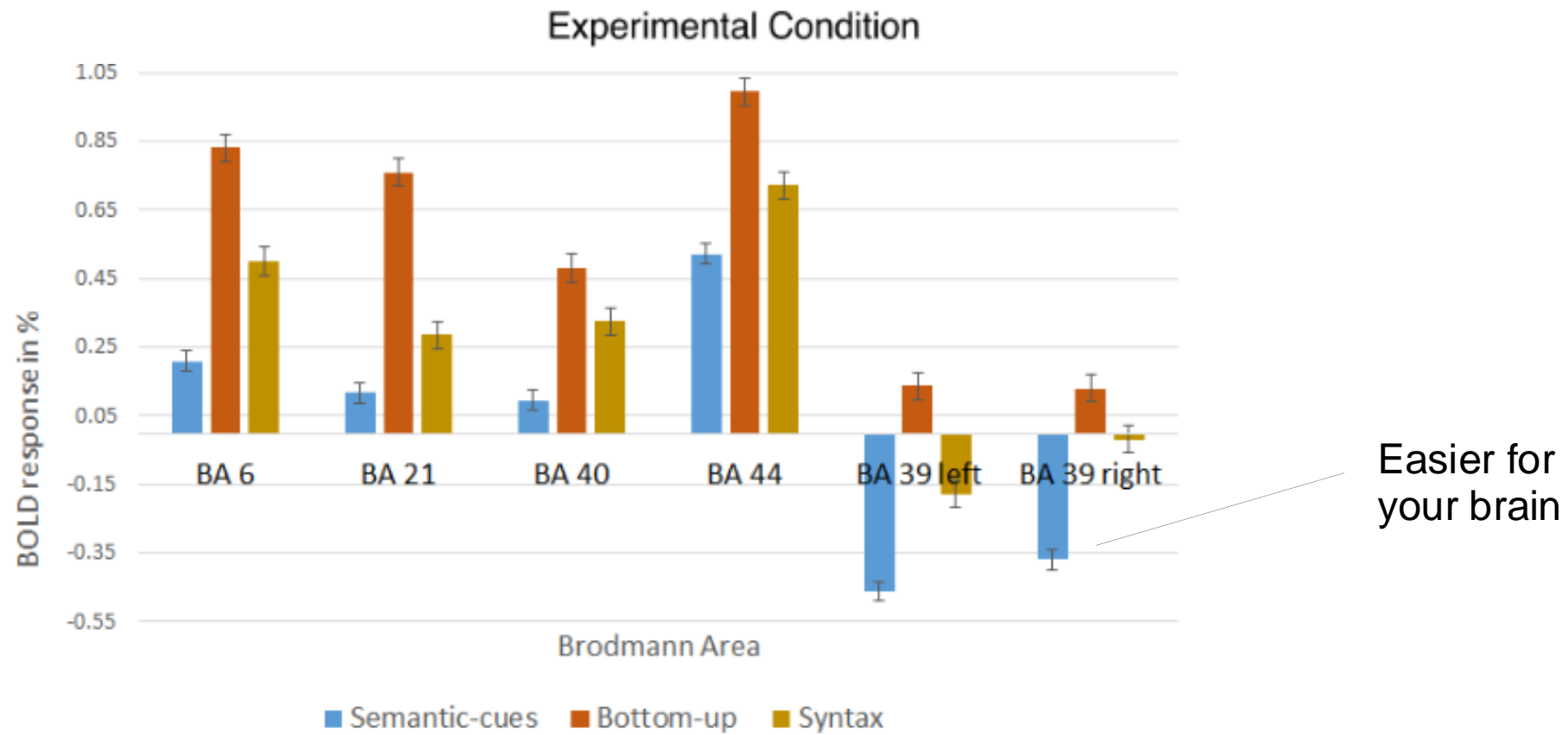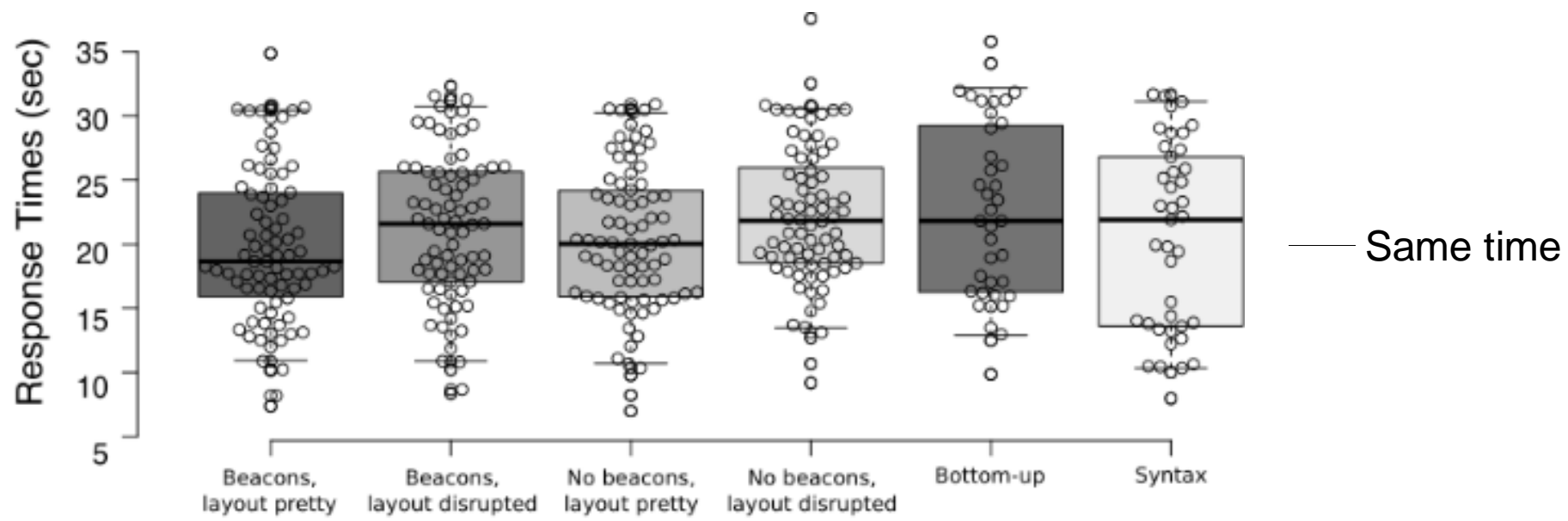
Result 2

- How do layout and beacons in source code influence program comprehension?

- As far as we can tell, they do ***not***. No significant differences in brain activation.
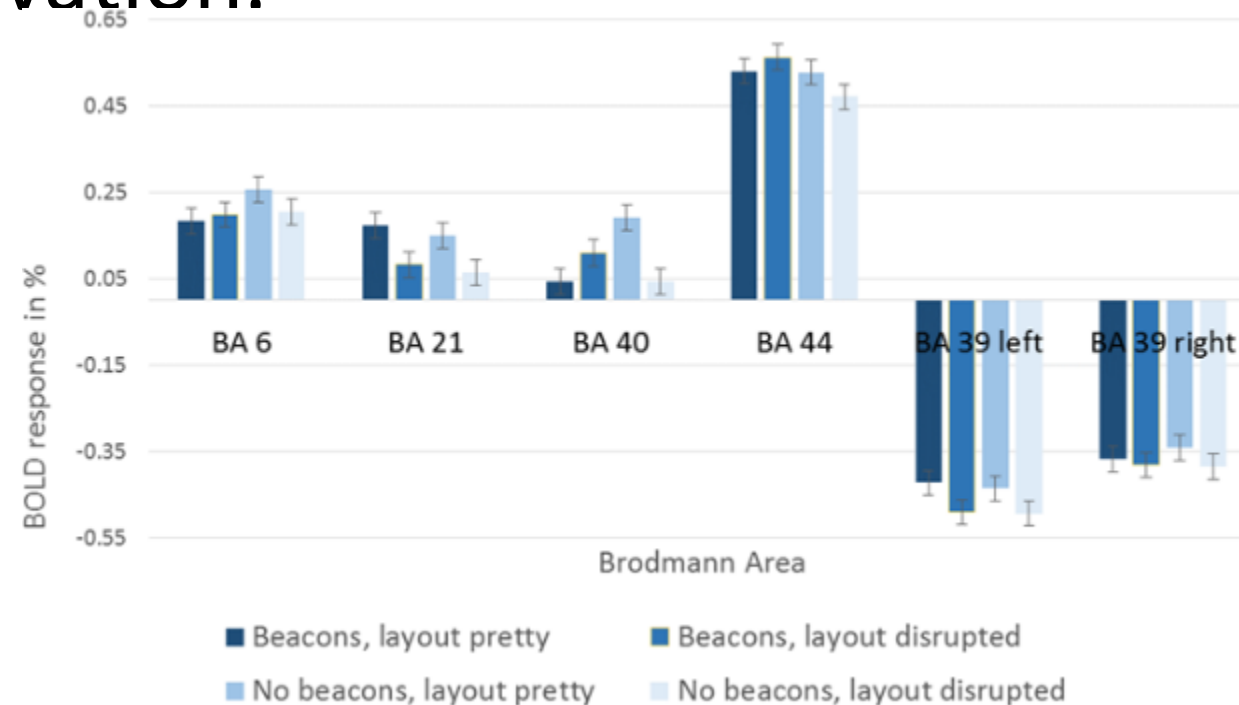
Comprehension Take-Home #1

- Program comprehension based on semantic cues is a very efficient process for understanding source code compared with the tedious, statement-by-statement process employed during bottom-up comprehension.

- When you are choosing identifier names and perhaps writing comments and documentation, emphasize beacons, cues and semantic plans: hint at the program's purpose or idiom
  - cf. high-level "why" vs. low-level "what" documentation

Code Complexity Metrics and the Brain
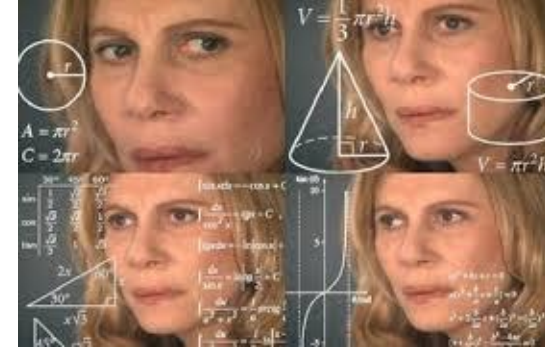
- Code complexity metrics are quantitative measures that aim to assess the difficulty level in understanding and maintaining a software.

- While we could use these metrics to predict how programmers understand code, it is debated.
  - Scalabrino et al. found minuscule correlations between complexity metrics and code comprehension

- Proposal: Let's use fMRI to assess how these metrics reflect difficulty in code comprehension!

[Siegmund et al. Program Comprehension and Code Complexity Metrics: An fMRI Study. ICSE. Best paper award.]

Experimental Setup and Results

- Participants were shown code snippets with varying code complexity scores of 4 different metrics

- Participants were asked to determine output AND subjectivity judge complexity

- What did they find - Are complexity metrics connected to how programmers process code?
  - Yes and no?

Complexity Metrics

LOC   Halstead   McCabe   DepDegree

- [Read the paper!](#)

High-Level Question #2

- Is reading code more like doing math or more like reading prose?

Experimental Design: 3 Tasks

- Code Comprehension

- Code Review   (top 100 GitHub repos)

- Prose Review   (College Board SAT, etc.)

(a) Code Comprehension

(b) Code Review

(c) Prose Review

Experiment Setup and Data

- 29 grads and undergrads (38% women)
  - Right-handed, native English speakers, corrected-to-normal vision, etc.

- Placed in fMRI, computer projection displayed via mirror

- A single participant completing four 11-minute runs produces 399,344,400 floating point numbers of data (153,594 voxels × 650 volumes × 4 runs)
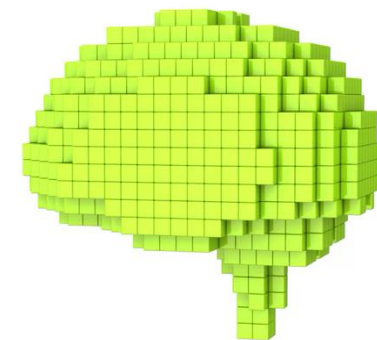
Dead Fish and Software Bugs

Neural correlates of interspecies perspective taking in the post-mortem Atlantic Salmon: An argument for multiple comparisons correction

Craig M. Bennett[1], Abigail A. Baird[2], Michael B. Miller[1], and George L. Wolford[3]
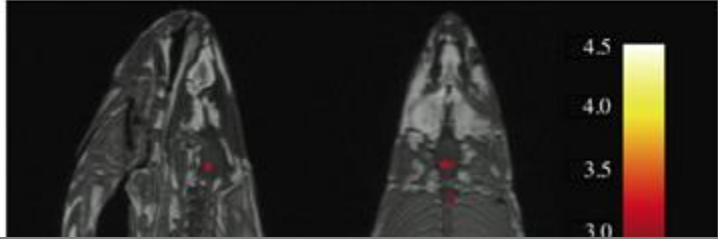
[1] Psychology Department, University of California Santa Barbara, Santa Barbara, CA; [2] Department of Psychology, Vassar College, Poughkeepsie, NY; [3] Department of Psychological & Brain Sciences, Dartmouth College, Hanover, NH

INTRODUCTION

With the extreme dimensionality of functional neuroimaging data comes extreme risk for false positives. Across the 130,000 voxels in a typical fMRI volume the probability of a false positive is almost certain. Correction for multiple comparisons should be completed with these datasets, but is often ignored by investigators. To illustrate the magnitude of the problem we carried out a real experiment that demonstrates the danger of not correcting for chance properly.

GLM RESULTS

Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates

Anders Eklund[a,b,c,1], Thomas E. Nichols[d,e], and Hans Knutsson[a,c]

[a]Division of Medical Informatics, Department of Biomedical Engineering, Linköping University, S-581 85 Linköping, Sweden; [b]Division of Statistics and Machine Learning, Department of Computer and Information Science, Linköping University, S-581 83 Linköping, Sweden; [c]Center for Medical Image Science and Visualization, Linköping University, S-581 83 Linköping, Sweden; [d]Department of Statistics, University of Warwick, Coventry CV4 7AL, United Kingdom; and [e]WMG, University of Warwick, Coventry CV4 7AL, United Kingdom

The most widely used task functional magnetic resonance imaging (fMRI) analyses use parametric statistical methods that depend on a variety of assumptions. In this work, we use real resting-state data

(FWE), the chance of one or more false positives, and empirically measure the FWE as the proportion of analyses that give rise to any significant results. Here, we consider both two-sample and

Preprocessing and Overfitting

- A significant challenge in fMRI analysis is processing the data correctly

- We cannot naively build a model from 150,000 features and 100 labeled instances

- Align and unwarp data, coregistered with a high-resolution anatomical scan, generalized linear models, high pass filters, robust weighted least squares, multivariate Gaussian process classification, feature selection via Automated Anatomical Labeling atlas, kernel function, expectation propagation …

Results: Mind Reading

- We can classify which task a participant is undertaking based solely on brain activity

  - Balanced accuracy 79%, p < .001

- These results suggest that Code Review, Code Comprehension, and Prose Review all have largely distinct neural representations
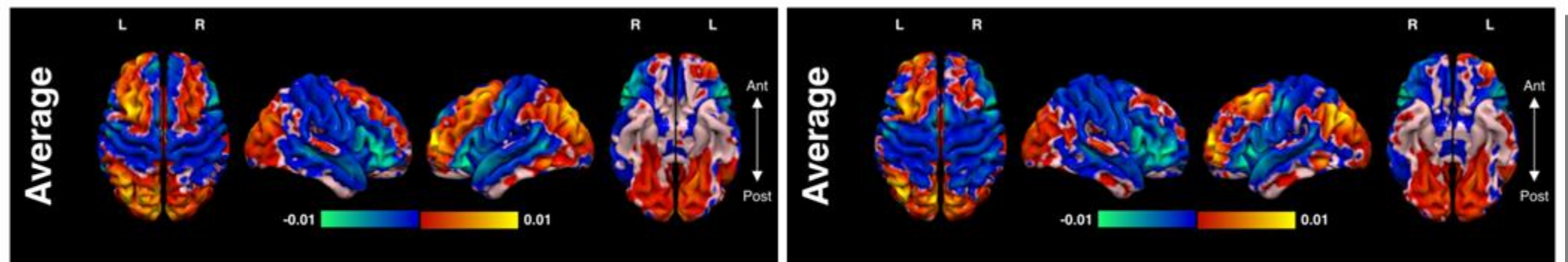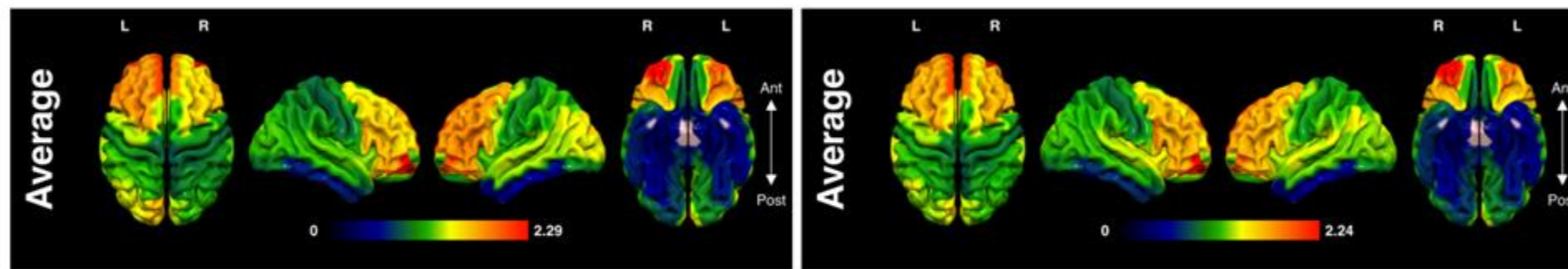

(a) Code Comprehension vs. Prose Review   (b) Code Review vs. Prose Review

Results: Can we relate tasks to brain regions?

- Near-perfect correspondence: r=0.99, p<.001

(a) Code Comprehension vs. Prose Review   (b) Code Review vs. Prose Review

- A wide swath of prefrontal regions known to be involved in higher-order cognition (executive control, decision-making, language, conflict monitoring, etc.) were highly weighted
  - Activity in those areas strongly drove the distinction between code and prose processing

Results: Can we relate expertise to classification accuracy?

- "Expertise" = (CS GPA) * (CS Credits Taken)

- How accurately our model distinguishes between Code Comprehension and Prose significantly predicted expertise (r = -0.44, p=0.016)

- The inverse relationship between accuracy and expertise suggests that, as one develops more skill in coding, the neural representations of code and prose are less differentiable. That is, programming languages are treated more like natural languages with greater expertise.

Comprehension Take-Home #2

- Neural representations of programming and natural languages are **distinct**. Classifiers can distinguish them based solely on brain activity. The same brain locations distinguish all three tasks. Greater **skill** accompanies a less-differentiated neural representation.

- These studies are still exploratory

- The area is wide open for future work
  - Social relationships, experts, writing code

- [Floyd et al. *Decoding the representation of code in the brain: An fMRI study of code review and expertise.* ICSE. Best paper award.]

Current and Future Studies

- Social relationships (boss over shoulder)
- Spatial processing (trees vs. tetris)
- Debugging code and varied Identifier names
- Industrial expertise (replicate protocol)
- Writing code (fMRI-safe keyboard)
- Transcranial magnetic stimulation (read-write)
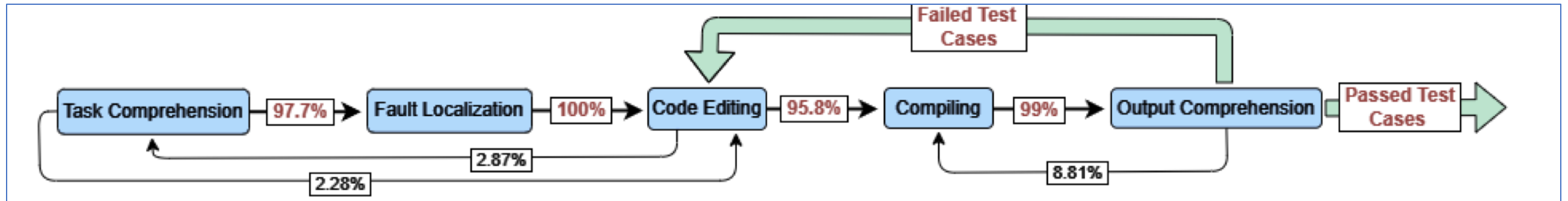
- Does any of this sound interesting? …
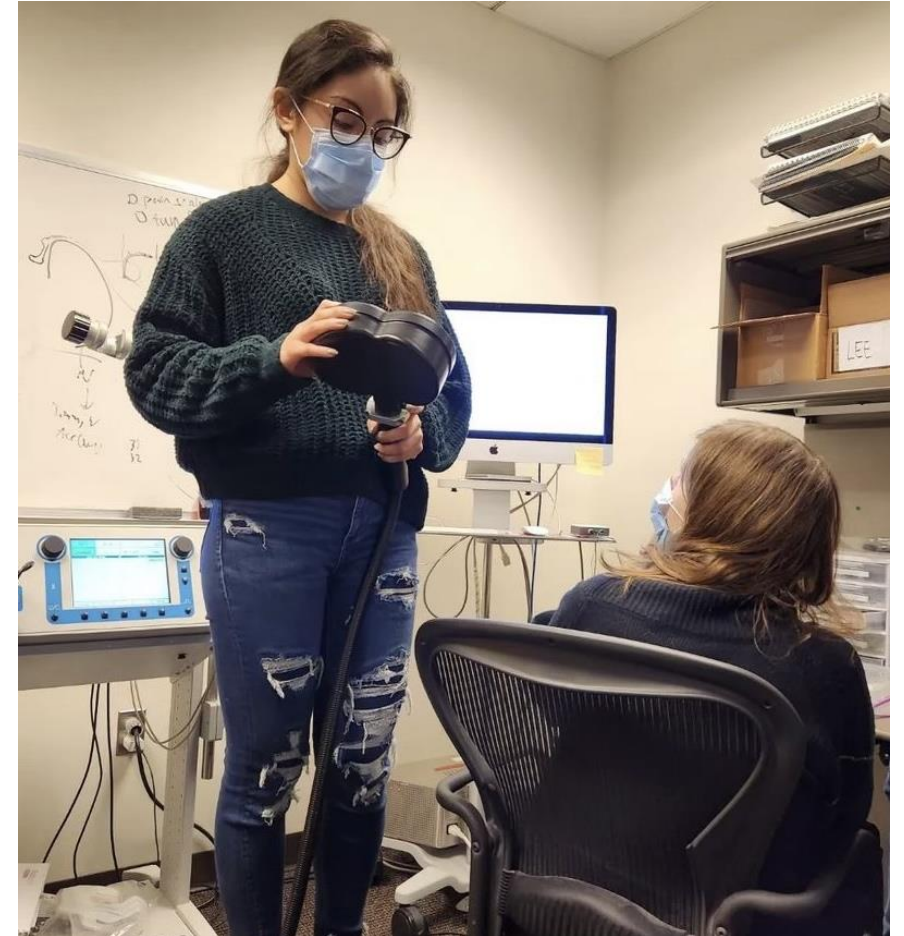
Code and Prose Writing
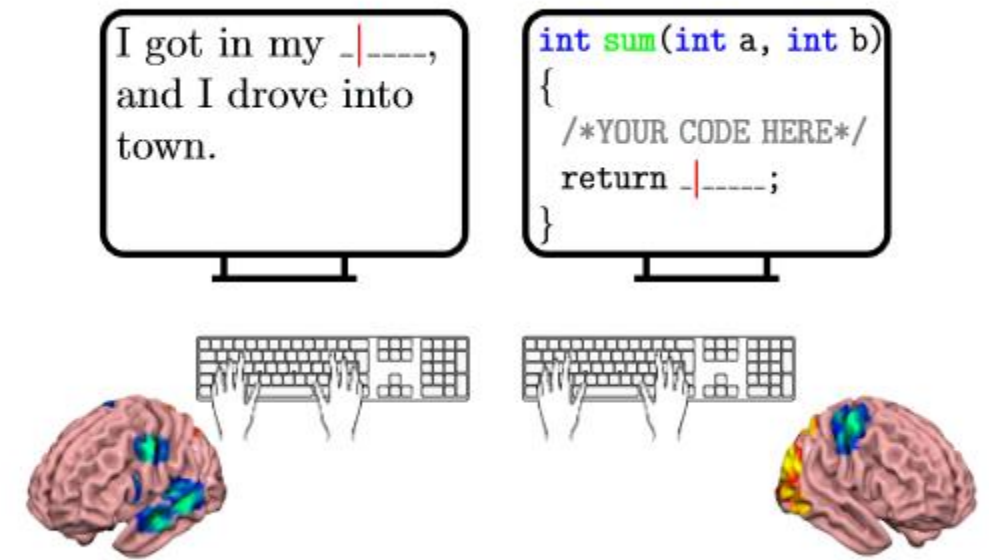
- **Code writing** requires significantly more activity in parts of the brain associated with **top-down control**, **planning**, and **categorization** than **prose writing**

- **Coding** involves right-lateralized brain regions associated with **attention**, **memory**, **planning**, and **spatial ability**

- [Krueger et al. *Neurological Divide: An fMRI Study of Prose and Code Writing.* ICSE.]

Figure 4: Example stimuli with experimentally-controlled variations corresponding to identifier morphology.

- fNIRS shows that the **stages of debugging are neurally distinct** (p < 0.001)

- Identifier Names that vary in morphological structure show increase neural activity - increase cognitive load (t-value = 2.71 – 5.75)

[Hu et al. Towards a Cognitive Model of Debugging: Does Identifier Construction Matter? *TSE*.]
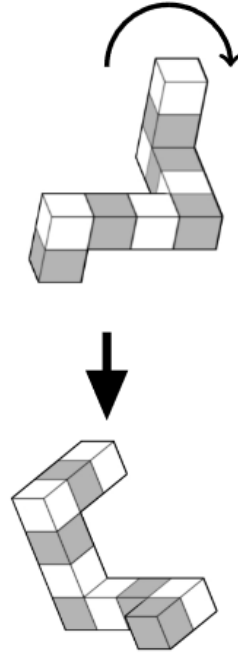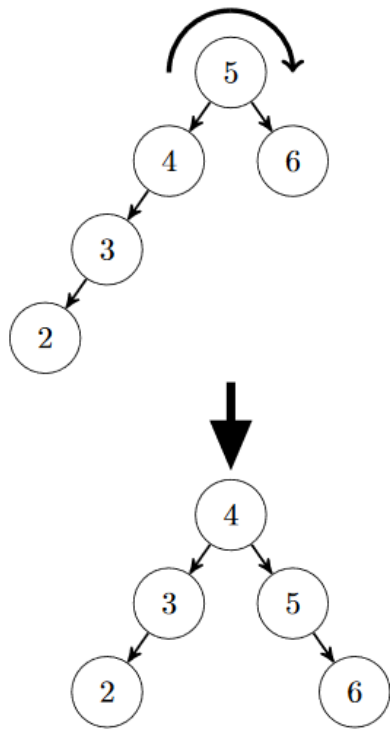
- fMRI and fNIRS show that mental rotation and data structure tasks use the same parts of the brain (e.g., 95% voxel similarity, $p < 0.01$)

- The brain works harder (cognitive load) to solve more difficult (Big-Oh) CS problems

[Huang et al. *Distilling Neural Representations of Data Structure Manipulation using fMRI and fNIRS.* ICSE.]
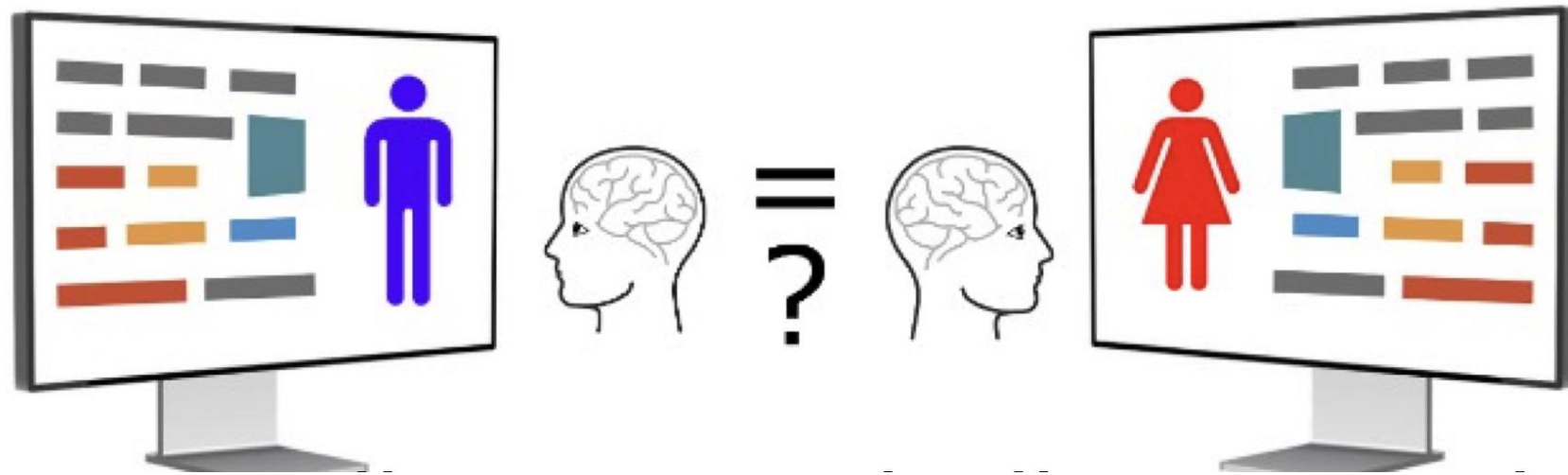
- Take the same pull request and tell some people it was written by a man, tell other it was written by a women?

- Men and women participants employ different high-level problem-solving strategies during code review
  - Men fixated more frequently (p <0.001), while women spent more time analyzing messages and authors (p = 0.02)

- All participants spent less time evaluating the pull requests of women (t = -2.759)
  - But participants do not self-acknowledge a gender bias
  - Paper presentation Video Here!

[Huang et al. Biases and Differences in Code Review using Medical Imaging and Eye-tracking: Genders, Humans, and Machines. FSE.]

Questions?

- Guest next lecture!

- HW6a Report Due TODAY!