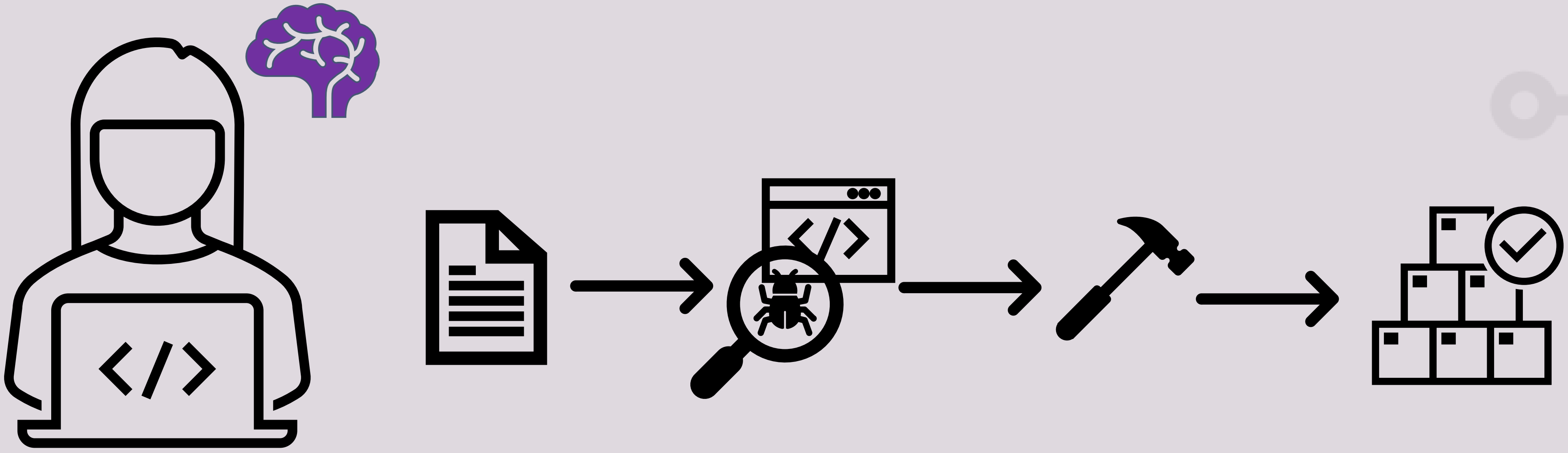


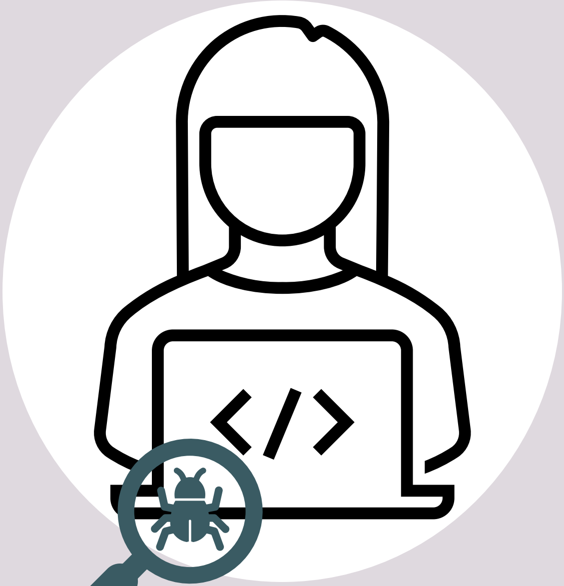
Read Paper!

Towards a Cognitive Model of Dynamic Debugging: Does Identifier Construction Matter?

Danniell Hu, **Priscila Santiesteban**, Madeline Endres, Westley Weimer

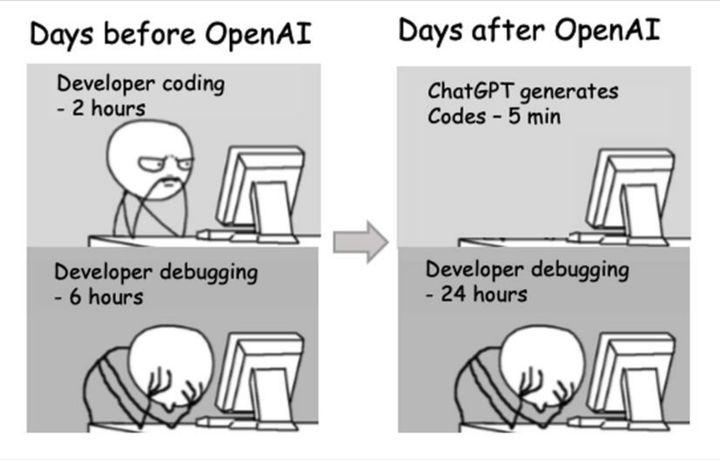
University of Michigan

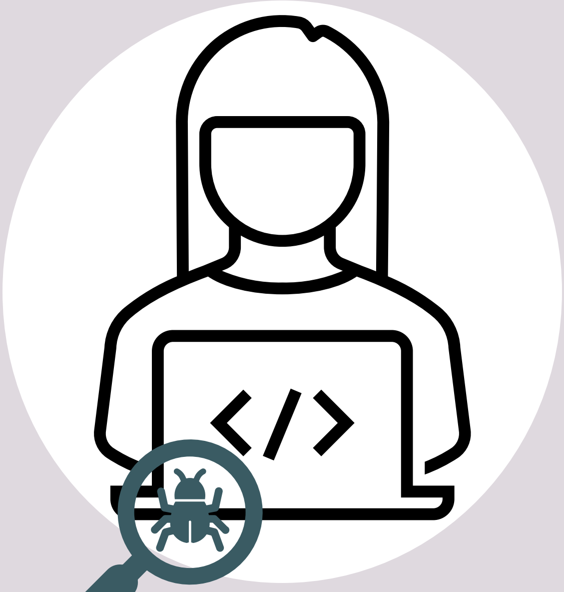
Journal First – ICSE Ottawa 2025



Software Engineer

Problem:
Despite advances in development tools,
debugging remains a human-driven process





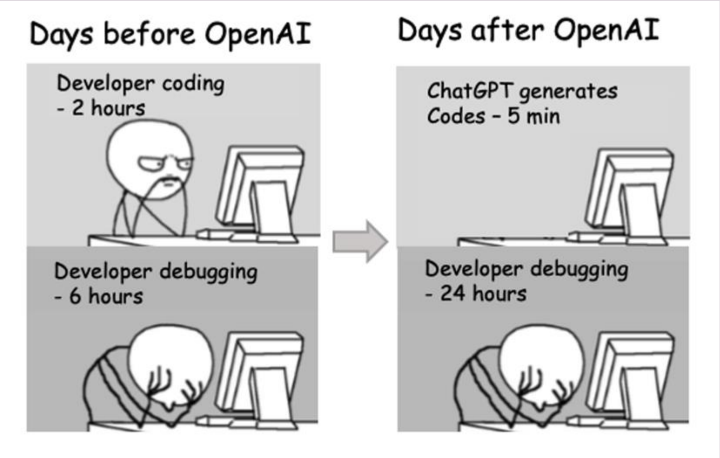
Software Engineer

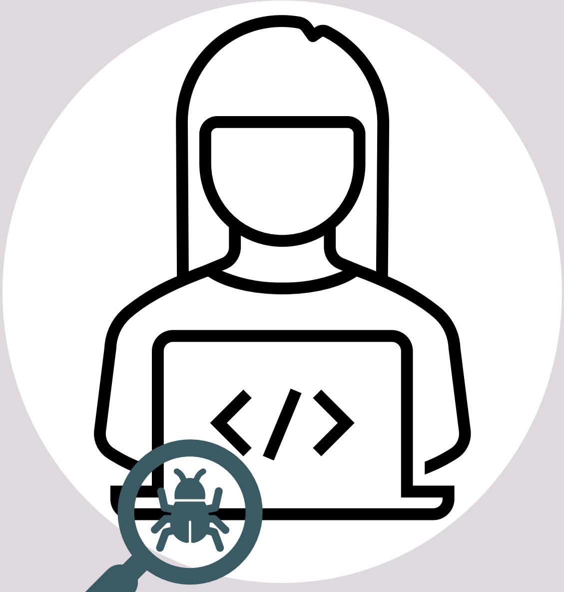
Problem:
Despite advances in development tools,
debugging remains a human-driven process

Automated Program Repair, What Is It Good For? Not Absolutely Nothing!

Hadeel Eladawy Claire Le Goues Yuriy Brun

As a result, at Facebook, APR is a part of what is ultimately, a human-driven debugging process. Developers act as oracles of patch appropriateness for both internally deployed Getafix [3] and SapFix [44] APR tools. However, for some categories of bugs, only





Software Engineer

Problem:
Despite advances in development tools,
debugging remains a human-driven process

Exploring ChatGPT's code refactoring capabilities: An empirical study

Kayla DePalma, Izabel Miminoshvili, Chiara Henselder, Kate Moss, Eman Abdullah AlOmar *

Stevens Institute of Technology, Hoboken, NJ, USA

eous

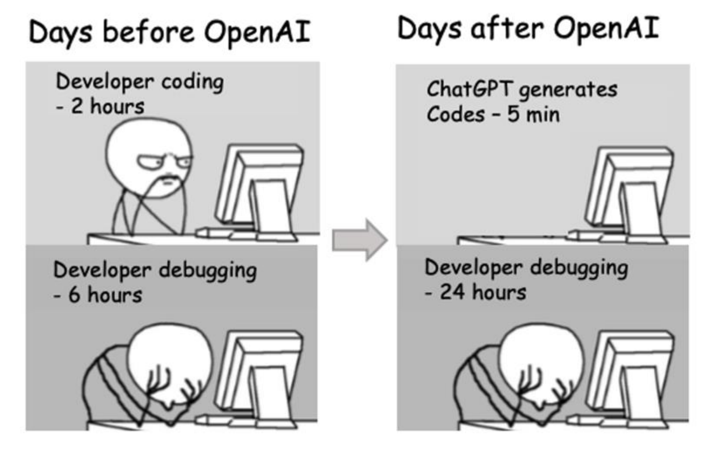
ded

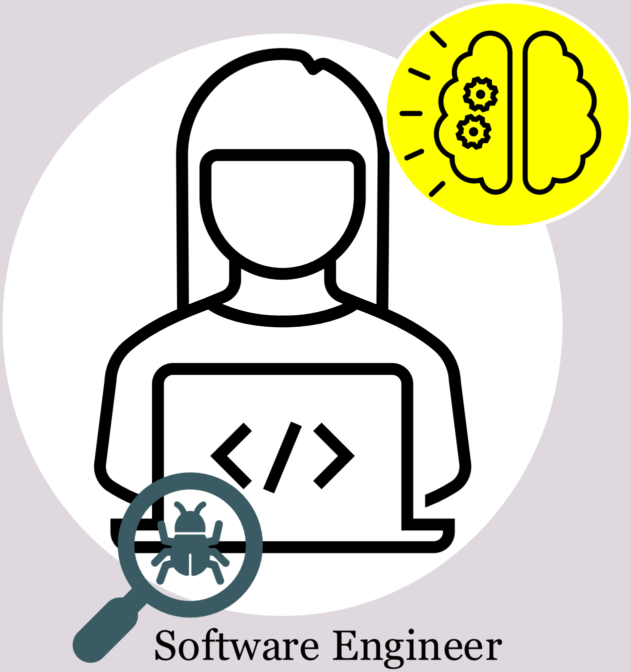
to oversee these changes and determine their significance. ChatGPT should be used as an aid to programmers since we cannot completely depend on it yet.

Automated Program Repair, What Is It Good For? Not Absolutely Nothing!

Hadeel Eladawy Claire Le Goues Yuriy Brun

As a result, at Facebook, APR is a part of what is ultimately a human-driven debugging process. Developers act as oracles of patch appropriateness for both internally deployed Getafix [3] and SapFix [44] APR tools. However, for some categories of bugs, only

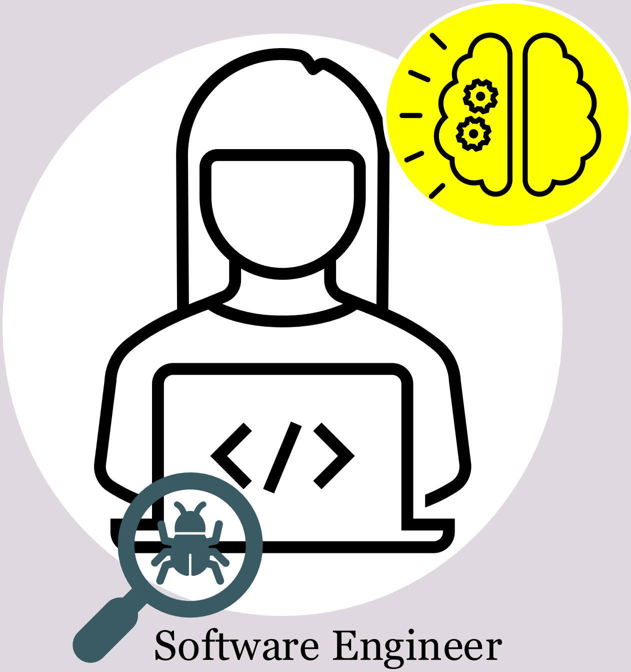




Software Engineer

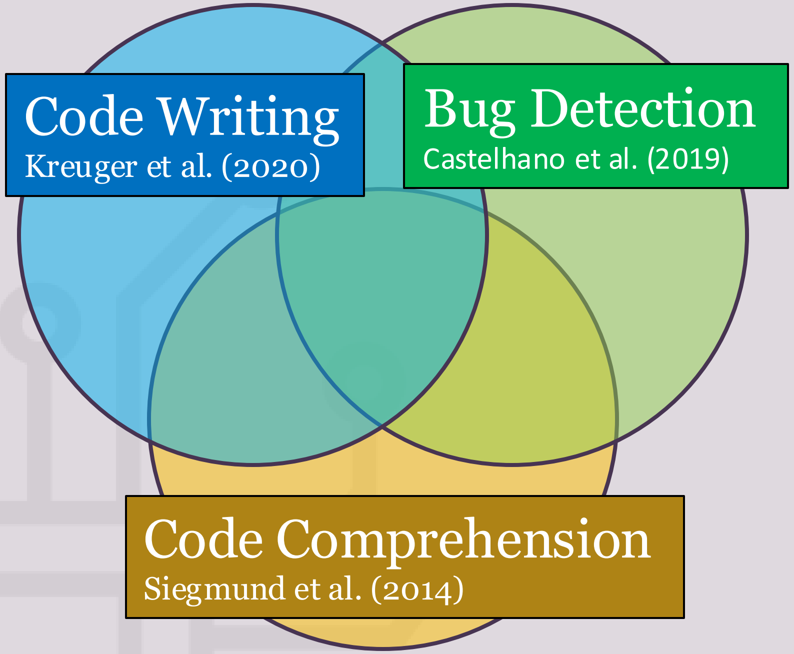
Problem:
Despite advances in development tools,
debugging remains a human-driven process

Approach:
If debugging is mostly a **human-reasoning task**, we can **understand it better by looking at the brain!**



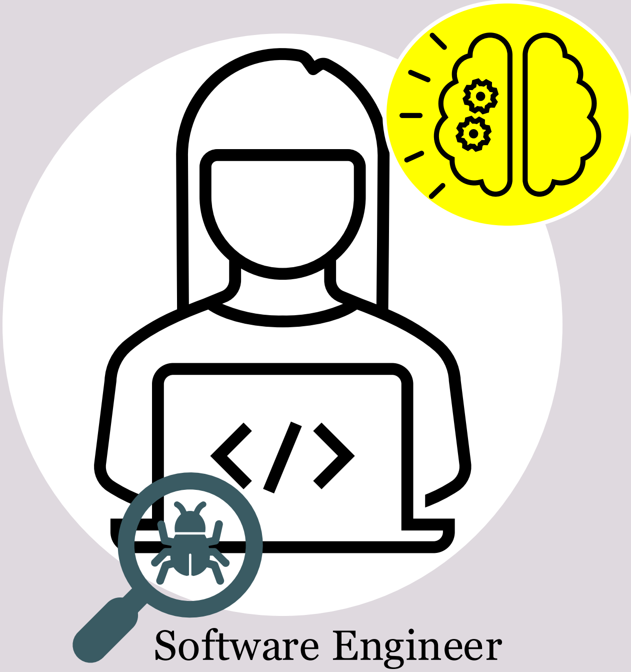
Problem:
Despite advances in development tools,
debugging remains a human-driven process

Approach:
If debugging is mostly a **human-reasoning task**, we can **understand it better by looking at the brain!**



State of the Art – Issues & Gaps:

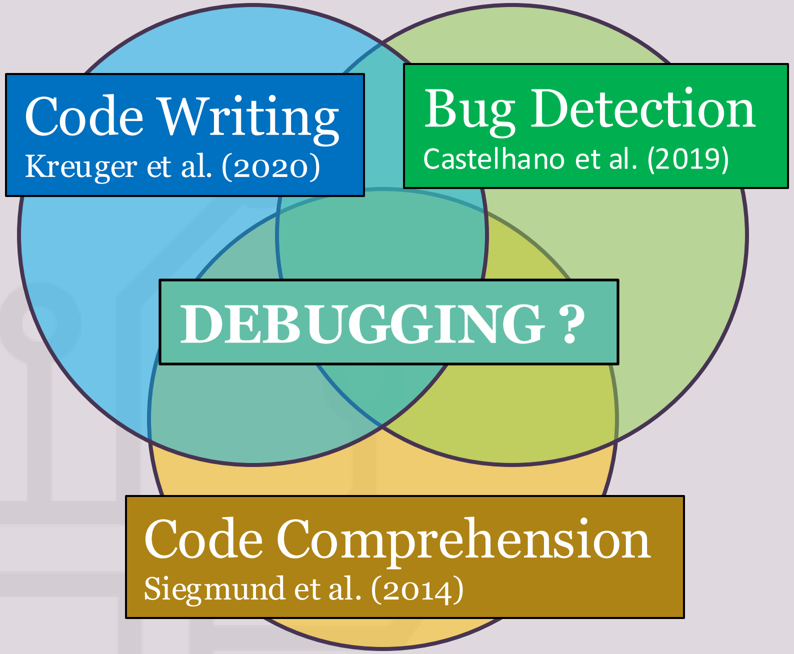
- **Issue:** Debugging is *dynamic* — not yet cognitively modeled as a whole
- **Issue:** Existing cognitive models don't scale to *real-word debugging*
- **Gap:** We need models that reflect *diverse developer experiences*



Software Engineer

Problem:
Despite advances in development tools,
debugging remains a human-driven process

Approach:
If debugging is mostly a **human-reasoning task**, we can **understand it better by looking at the brain!**

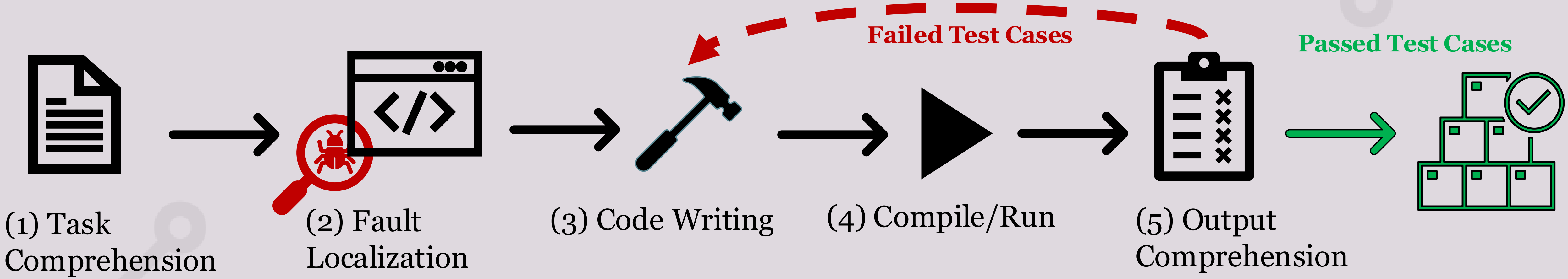


State of the Art – Issues & Gaps:

- **Issue:** Debugging is *dynamic* — not yet cognitively modeled as a whole
- **Issue:** Existing cognitive models don't scale to *real-word debugging*
- **Gap:** We need models that reflect *diverse developer experiences*

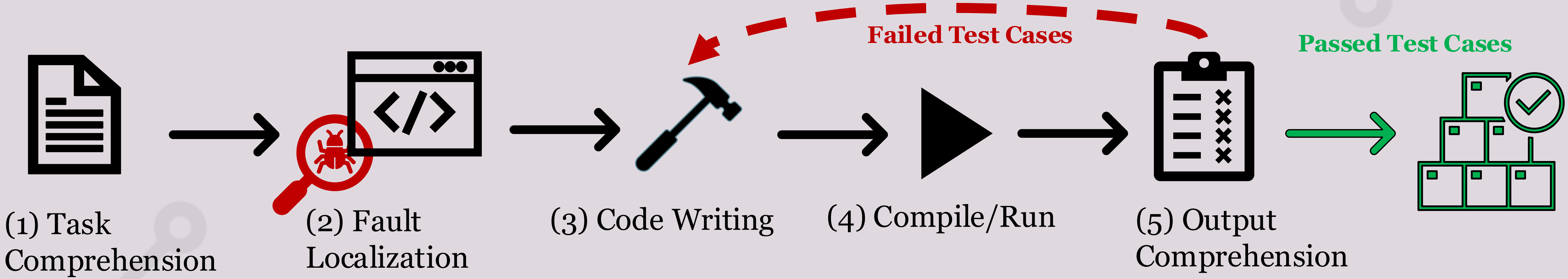
To Address Issue – Existing Cognitive Models Do Not Scale to Real-World Debugging

(1) We **propose a direct, FIVE staged model of end-to-end debugging** that may generalize to more realistic code for which programmers transition in :



To Address Issue – Existing Cognitive Models Do Not Scale to Real-World Debugging

(1) We **propose a direct, FIVE staged model of end-to-end debugging** that may generalize to more realistic code for which programmers transition in :



Does each stage of debugging present distinct neural and behavioral activity?

To Address Gap - Existing models ignore individual differences

(2) Using our debugging model, we investigate how **variable naming and reading ability affects the debugging experience**

To Address Gap - Existing models ignore individual differences

(2) Using our debugging model, we investigate how **variable naming and reading ability affects the debugging experience**

What we know from Psychology:

Morphemes in English words impact English prose comprehension in some people.

Morpheme : a unit of meaning

To Address Gap - Existing models ignore individual differences

(2) Using our debugging model, we investigate how **variable naming and reading ability affects the debugging experience**

What we know from Psychology:

Morphemes in English words impact English prose comprehension in some people.

Morpheme : a unit of meaning

Examples:

Single-Morpheme: Father

Multi-Morpheme: Teacher

To Address Gap - Existing models ignore individual differences

(2) Using our debugging model, we investigate how **variable naming and reading ability affects the debugging experience**

What we know from Psychology:

Morphemes in English words impact English prose comprehension in some people.

Do morpheme-varied identifiers and reading ability affect debugging performance?

Morpheme : a unit of meaning

Examples:

Single-Morpheme: Father

Multi-Morpheme: Teacher

High-Level Study Overview

We present the first neuroimaging human study (n=28) of end-to-end debugging!

<u>Primary Contribution:</u>
We develop the first cognitively and behaviorally validated cognitive model of debugging. *(RQ1)*

<u>Secondary Contribution:</u>
We develop insights into how variable naming related to morphemes and reading ability contribute to debugging outcomes. *(RQ2) (RQ3)*

High-Level Study Overview - Methods

How do did we measure brain activity?

We use Functional Near Infrared Spectroscopy (**fNIRS**) to **capture distinct brain activation** patterns of **programmers while conducting real-world debugging.**

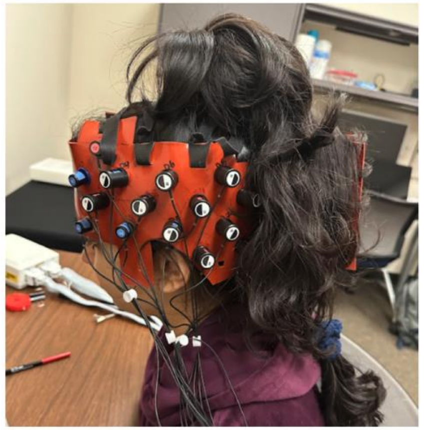
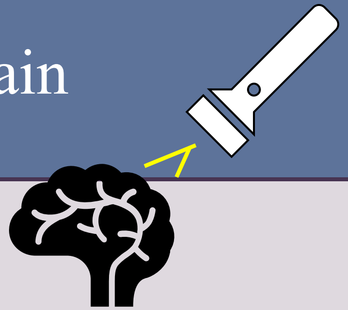
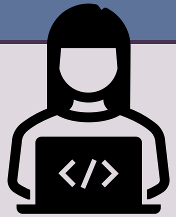


Figure 2: Image of our fNIRS cap on a participant. The cap goes around the head covering frontal and temporal regions.

fNIRS uses light to measure the oxygen levels in different parts of the brain



fNIRS allows for subjects to realistically program (e.g., on a laptop, with IDE) (unlike fMRI)



High-Level Study Overview - Methods

How do did we measure brain activity?

We use Functional Near Infrared Spectroscopy (**fNIRS**) to **capture distinct brain activation** patterns of **programmers while conducting real-world debugging.**

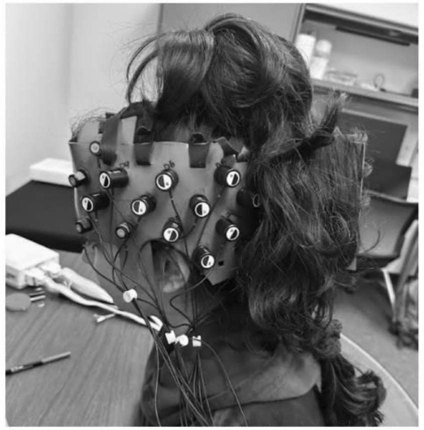
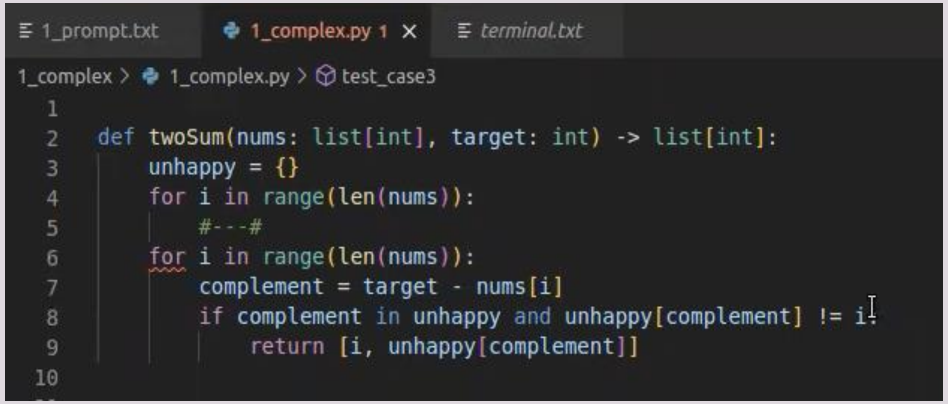


Figure 2: Image of our fNIRS cap on a participant. The cap goes around the head covering frontal and temporal regions.

How do did we measure behavior outcomes?

We use **VSCode** with extensions.

We track keystrokes, output files, compiled files, window-switching, **time spent at each file**, and **time spent to successfully fix bug**



Outline

1. Research Motivation and Overview

2. Experimental Design and Data Collection

3. Results

4. Research Implications

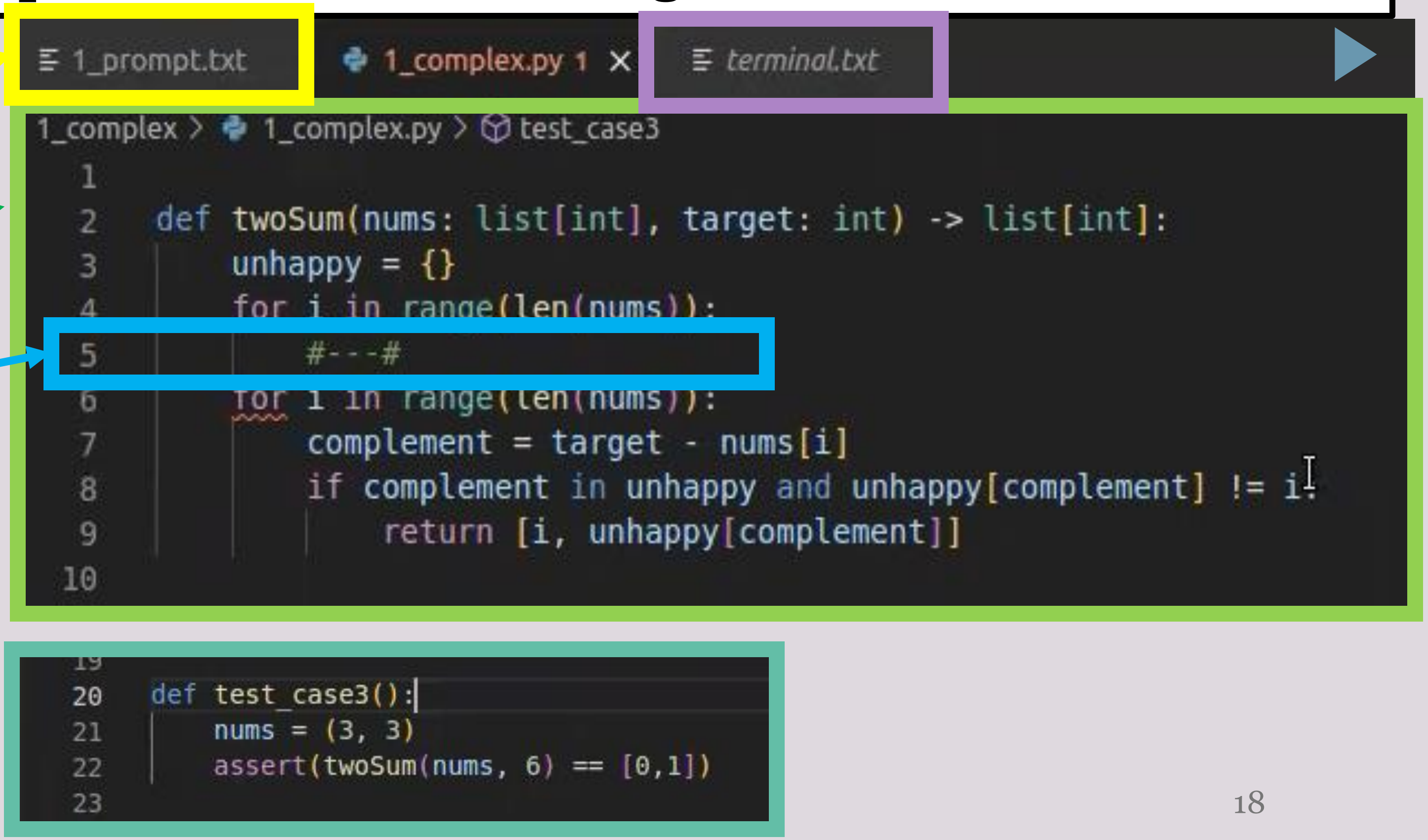
Experimental Design – Debugging Task

To assess the cognitive and behavior outcomes during debugging:

Participants were tasked with debugging 13 faulty Python programs in VSCode IDE in ~50 minutes (10 min per problem) while wearing an fNIRS device

<u>Each stimuli contains:</u>

- Problem Description
  (text file)
- Leetcode-Style Python Problem
  (<15 lines)
- 1 Seeded Defect
  (i.e., line of missing code)
- 1-3 Test Cases
- Error Message
  (text file)



Experimental Design – Morpheme-Identifier Conditions

To assess the impact of morpheme-related identifier naming on debugging:

We designed 4 identifier treatment conditions per debugging problem

```
def singleNumber(nums):
    lstDup = []
    for i in nums:
        # --- #
            lstDup.append(i)
        else:
            lstDup.remove(i)
    return lstDup.pop()
```

(a) Original variable name

```
def singleNumber(nums):
    brunly = []
    for i in nums:
        # --- #
            brunly.append(i)
        else:
            brunly.remove(i)
    return brunly.pop()
```

(b) Pseudoword

```
def singleNumber(nums):
    family = []
    for i in nums:
        # --- #
            family.append(i)
        else:
            family.remove(i)
    return family.pop()
```

(c) Single-morpheme

```
def singleNumber(nums):
    mostly = []
    for i in nums:
        # --- #
            mostly.append(i)
        else:
            mostly.remove(i)
    return mostly.pop()
```

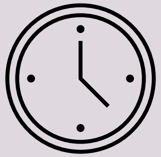
(d) Multi-morpheme

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

- We used a validated list of morphemes from the established corpus of Marks *et al.*
- Randomly assigned to participants (no participant saw the same problem more than once)

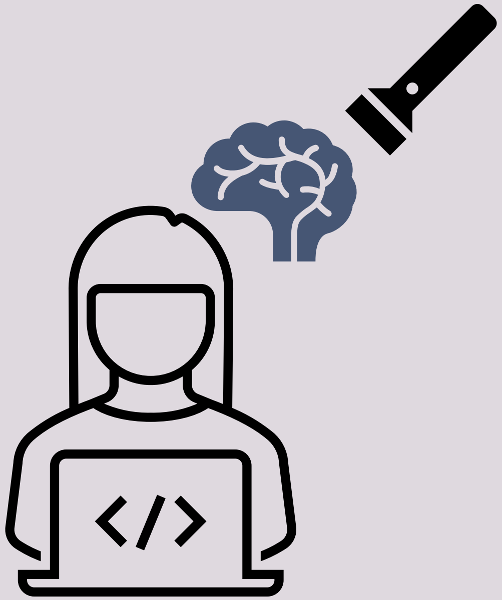
Experimental Design – Data Collection & Analysis

Each study session lasted ~90 minutes



- **13 stimuli** – debugging tasks with each having
  - **4 variations** per identifier condition
- Reading Ability Test

Data Analysis - 28 Participants (7 women, 21 men)



- Compare activation by each debugging phase and identifier condition by brain area using best practices from psychology
- Compare debugging outcomes within conditions
- **Significance threshold: $p < 0.05$**
- **FDR to correct for multiple comparisons: $q < 0.05$**

Outline

1. Research Motivation and Overview

2. Experimental Design and Data Collection

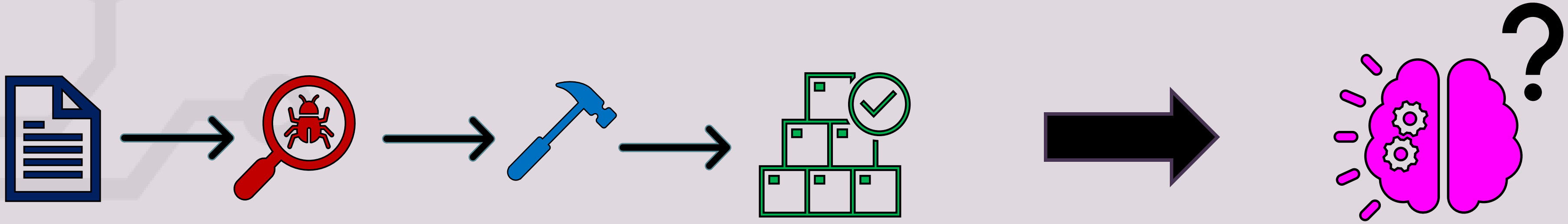
3. Results

4. Research Implications

Research Questions

We aim to answer the following research questions:

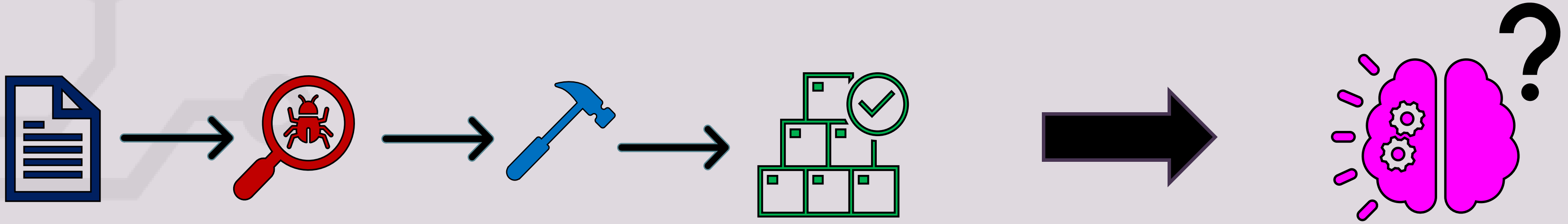
- RQ1 - Do our stages of debugging exhibit distinct (a) behavioral and (b) neural patterns?
- RQ2 - During debugging, how do morpheme-varied identifier names affect debugging (a) behaviorally and (b) cognitively?
- RQ3 - During debugging, how do individual skills (i.e., reading ability and programming experience) affect neural activity?



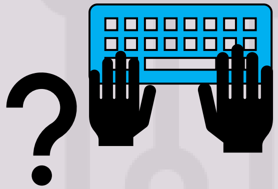
Research Questions

We aim to answer the following research questions:

- RQ1 - Do our stages of debugging exhibit distinct (a) behavioral and (b) neural patterns?
- RQ2 - During debugging, how do morpheme-varied identifier names affect debugging (a) behaviorally and (b) cognitively?
- RQ3 - During debugging, how do individual skills (i.e., reading ability and programming experience) affect neural activity?

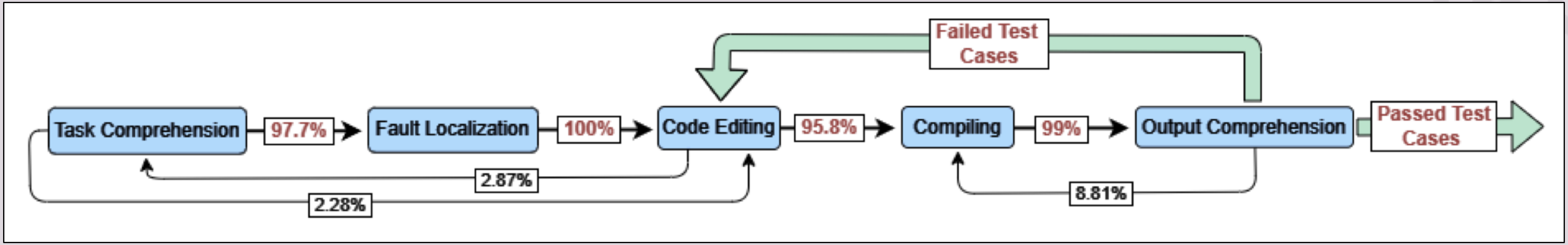


RQ1a: Are our stages of debugging **BEHAVIORALLY** distinct?



RQ1a: Are our stages of debugging BEHAVIORALLY distinct?

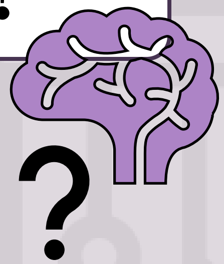
Answer: Yes!



(1) **Participants followed the expected stage-by-stage flow of the debugging model** in over **97% of cases**.

(2) **Participants spend statistically significantly different amounts of time** in each stage (**p<0.001**)

RQ1b: Are our stages of debugging COGNITIVELY distinct?



RQ1b: Are our stages of debugging COGNITIVELY distinct?

Answer: Yes!

(1) Debugging stages *are* correlated to different patterns of neural activity **($p<0.05$)**

- (i.e., each stage the model presents patterns of neural activity that vary from each other)

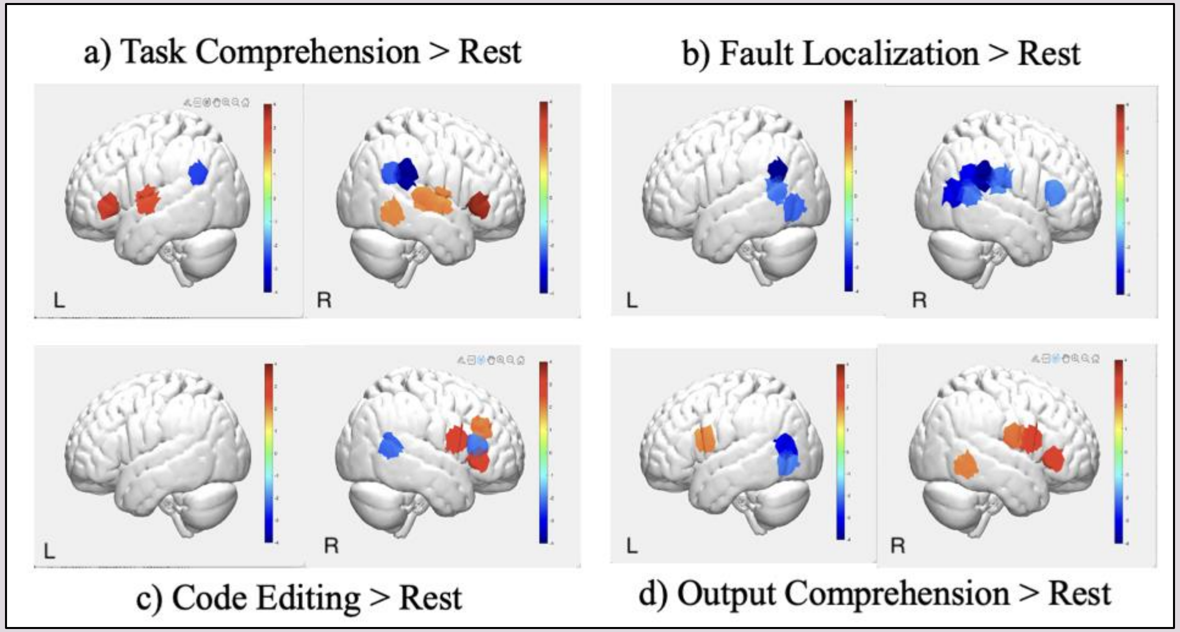
RQ1b: Are our stages of debugging **COGNITIVELY** distinct?

Answer: Yes!

(1) Debugging stages *are* correlated to different patterns of neural activity **($p<0.05$)**

- (i.e., each stage the model presents patterns of neural activity that vary from each other)

Red regions indicate statistically significantly different neural activity for that debugging state contrasted to “Rest” (i.e., doing nothing while waiting for compilation)



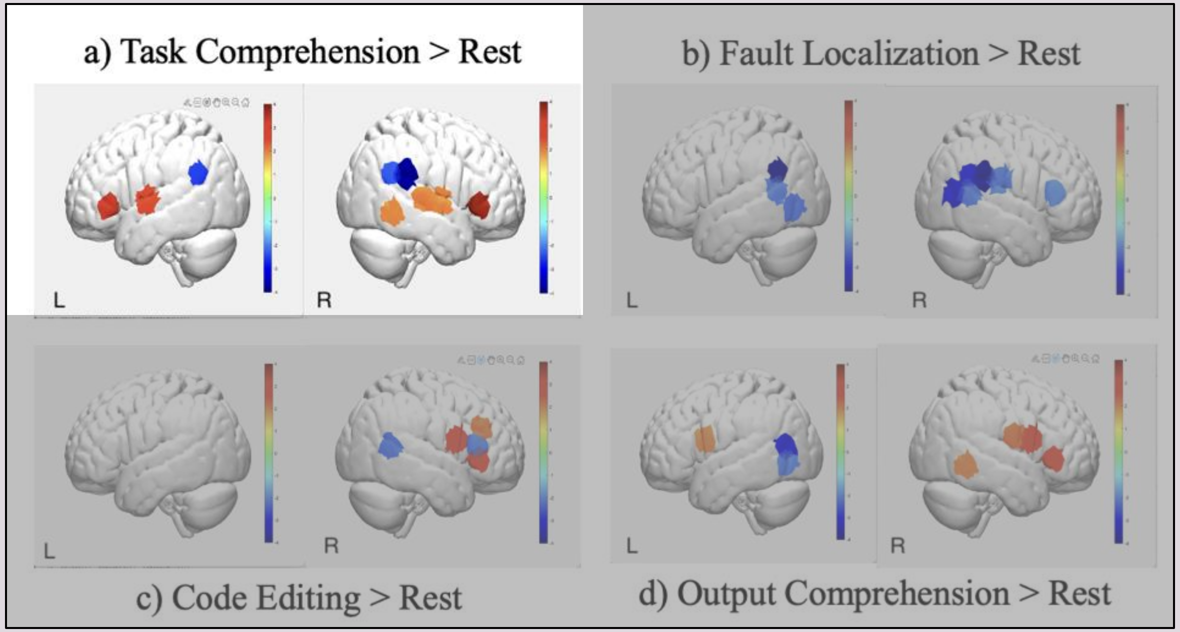
RQ1b: Are our stages of debugging COGNITIVELY distinct?

Answer: Yes!

(1) Debugging stages *are* correlated to different patterns of neural activity **($p<0.05$)**

- (i.e., each stage the model presents patterns of neural activity that vary from each other)

Debugging Stage	Key Brain Region	Cognitive Function
Task Comprehension	Temporal cortex (Wernicke's, Broca's, BA 21, 52)	Language comprehension, auditory processing



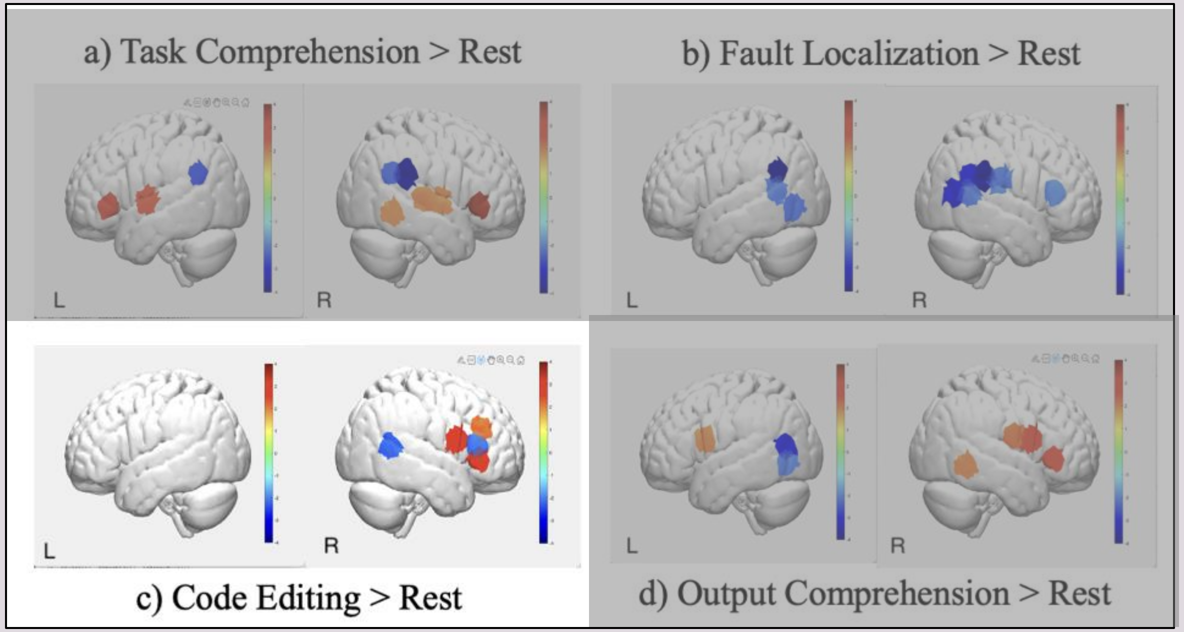
RQ1b: Are our stages of debugging COGNITIVELY distinct?

Answer: Yes!

(1) Debugging stages *are* correlated to different patterns of neural activity **($p<0.05$)**

- (i.e., each stage the model presents patterns of neural activity that vary from each other)

Debugging Stage	Key Brain Region	Cognitive Function
Task Comprehension	Temporal cortex (Wernicke's, Broca's, BA 21, 52)	Language comprehension, auditory processing
Code Editing	Angular gyrus (BA 39)	Spatial cognition, problem-solving



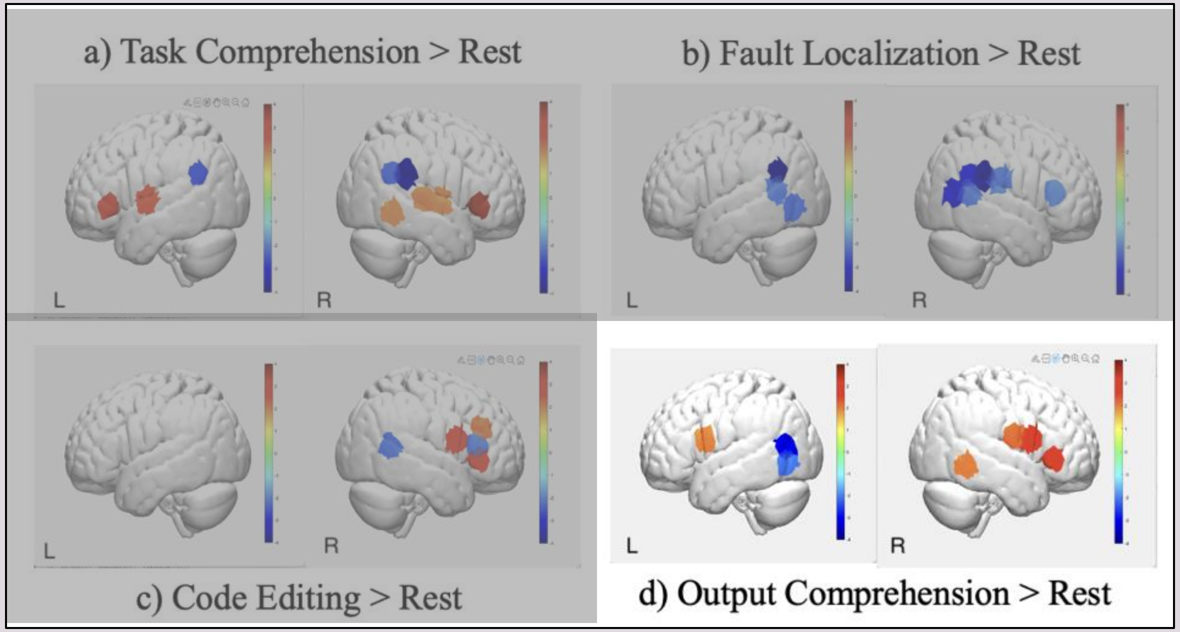
RQ1b: Are our stages of debugging **COGNITIVELY** distinct?

Answer: Yes!

(1) Debugging stages *are* correlated to different patterns of neural activity **($p<0.05$)**

- (i.e., each stage the model presents patterns of neural activity that vary from each other)

Debugging Stage	Key Brain Region	Cognitive Function
Task Comprehension	Temporal cortex (Wernicke's, Broca's, BA 21, 52)	Language comprehension, auditory processing
Code Editing	Angular gyrus (BA 39)	Spatial cognition, problem-solving
Output Comprehension	Right DLPFC (BA 46)	Working memory



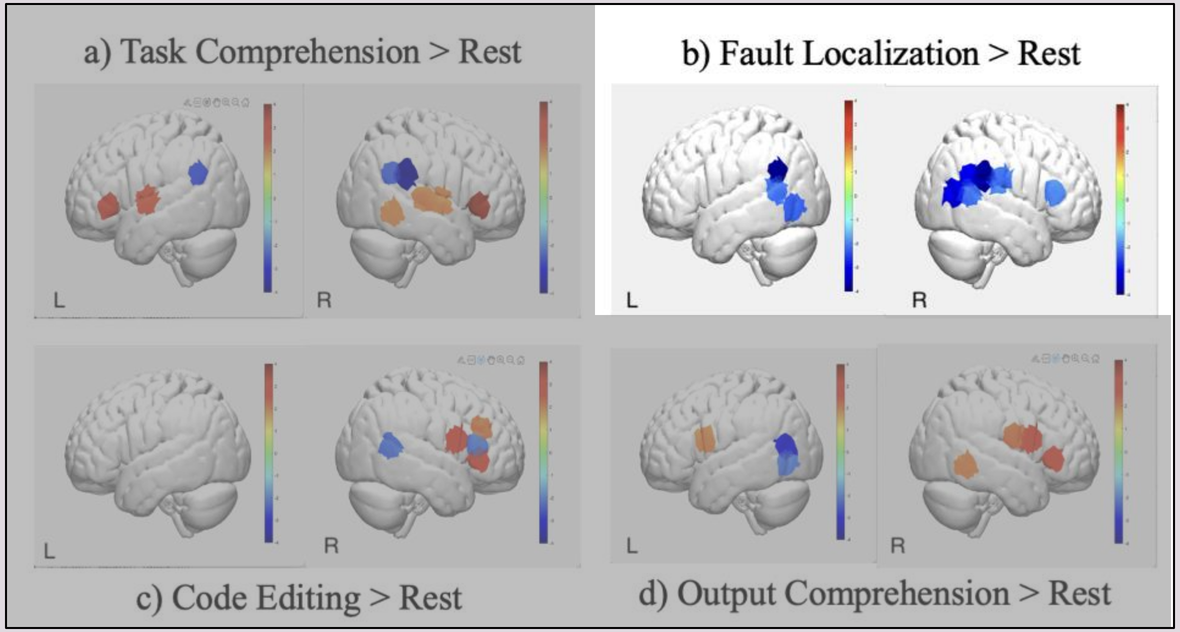
RQ1b: Are our stages of debugging COGNITIVELY distinct?

Answer: Yes!

(1) Debugging stages *are* correlated to different patterns of neural activity **($p<0.05$)**

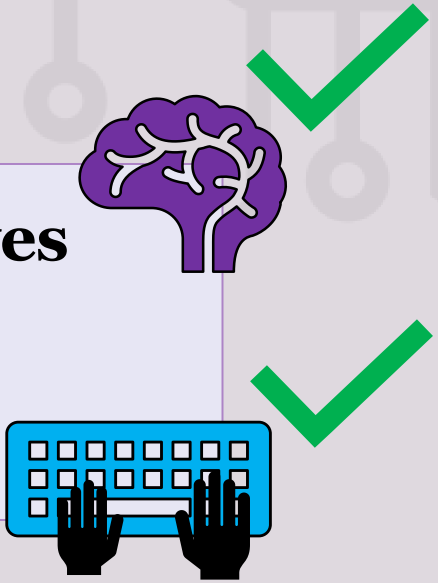
- (i.e., each stage the model presents patterns of neural activity that vary from each other)

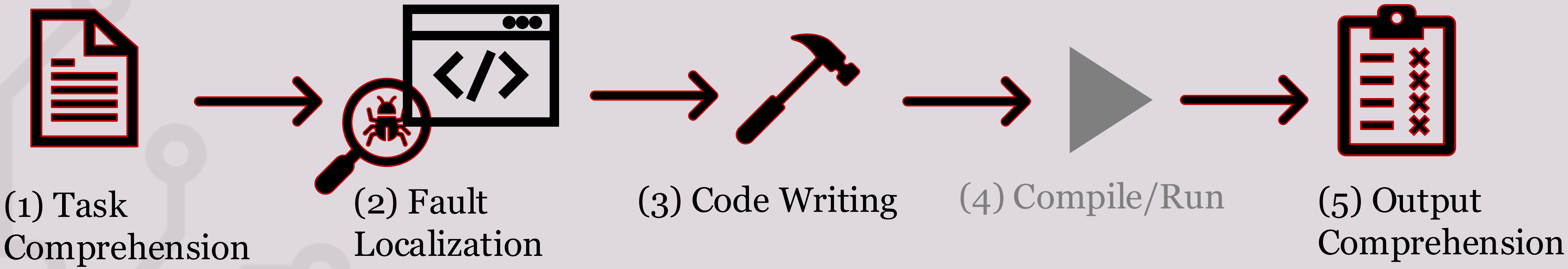
Debugging Stage	Key Brain Region	Cognitive Function
Task Comprehension	Temporal cortex (Wernicke's, Broca's, BA 21, 52)	Language comprehension, auditory processing
Fault Localization	No significant activation	???
Code Editing	Angular gyrus (BA 39)	Spatial cognition, problem-solving
Output Comprehension	Right DLPFC (BA 46)	Working memory



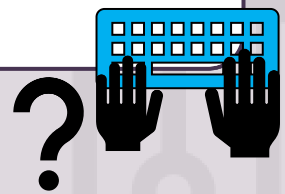
RQ1: Summary

These findings support the idea that the **debugging process involves behaviorally and cognitively distinct stages**, forming a robust foundation for the proposed end-to-end debugging model.

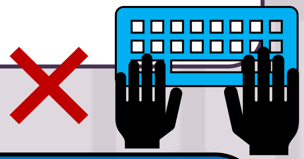




RQ2a: How do morpheme-varied identifier names correlate with specific patterns in BEHAVIORAL outcomes?

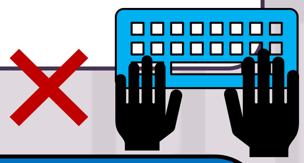


RQ2a: How do morpheme-varied identifier names correlate with specific patterns in BEHAVIORAL outcomes?



Answer: We find **no statistically-significant differences** in behavioral outcomes as a function of reading ability OR identifier conditions **($p > 0.09$)**.

RQ2a: How do morpheme-varied identifier names correlate with specific patterns in BEHAVIORAL outcomes?

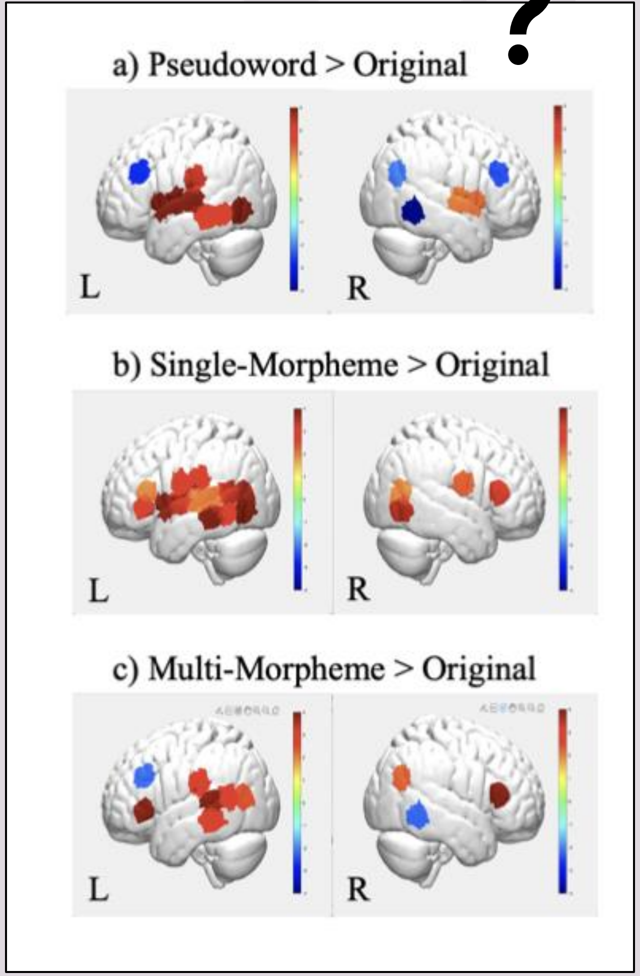


Answer: We find **no statistically-significant differences** in behavioral outcomes as a function of reading ability OR identifier conditions **($p > 0.09$)**.

This finding is interesting!

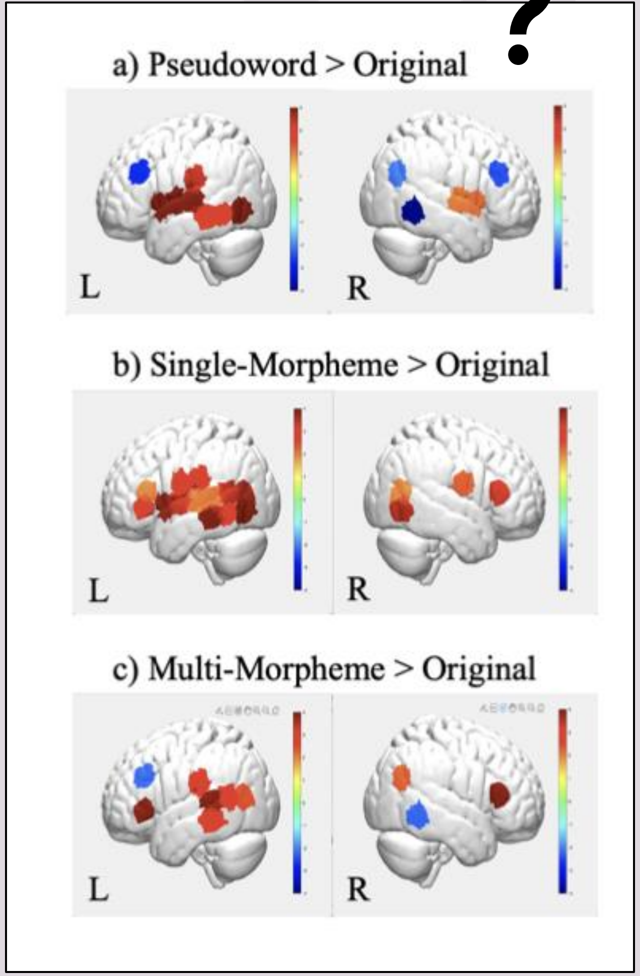
- Individuals with reading difficulties may struggle with complex prose.
- However, our findings suggest lower English reading ability do not significantly affect debugging speed or accuracy.
- Implications for hiring.

RQ2b: How do morpheme-varied identifier names correlate with specific patterns in COGNITIVE outcomes?



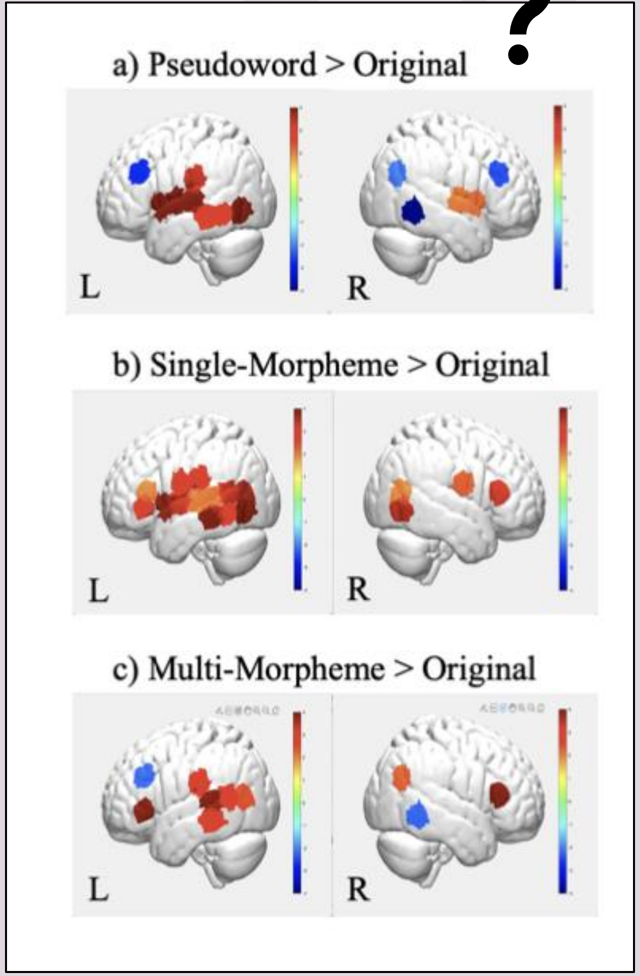
RQ2b: How do morpheme-varied identifier names correlate with specific patterns in COGNITIVE outcomes?

Answer: All three conditions show **increased neural activity** compared to original variables in language regions **($p < 0.05$)** with simple-morpheme showing a greater contrast.



RQ2b: How do morpheme-varied identifier names correlate with specific patterns in COGNITIVE outcomes?

Answer: All three conditions show **increased neural activity** compared to original variables in language regions **($p < 0.05$)** with simple-morpheme showing a greater contrast.



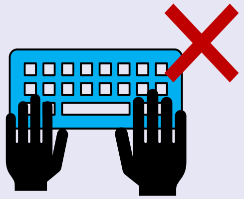
This finding is interesting!

- Poor naming might not slow people down behaviorally, but it still makes their brains work harder.
  - These findings replicate prior work that found that less meaningful identifiers lead to increased cognitive load (Siegmund et al. and Fakhoury et al.)

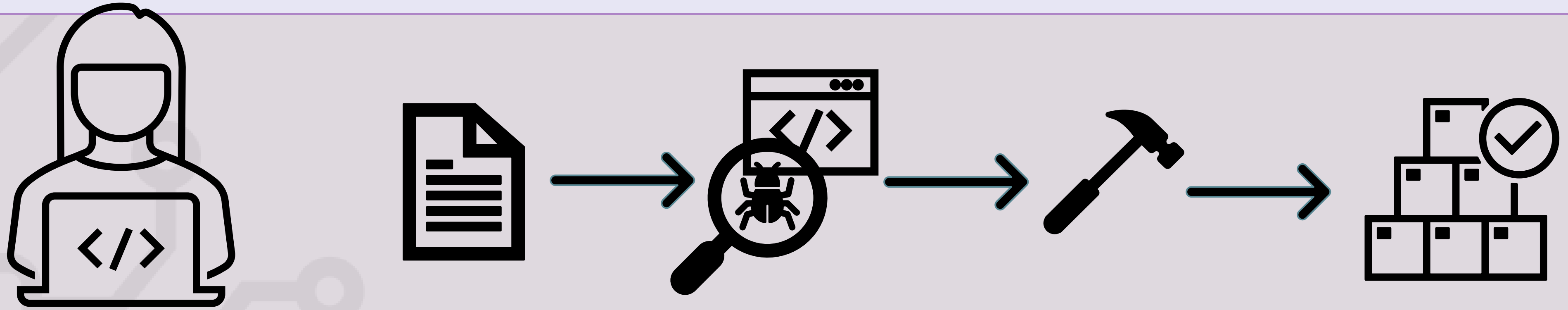
RQ2: Summary

Findings emphasize the importance of careful identifier naming

- Variable naming variations lead to no significant impact on debugging performance, even for those with reading difficulties



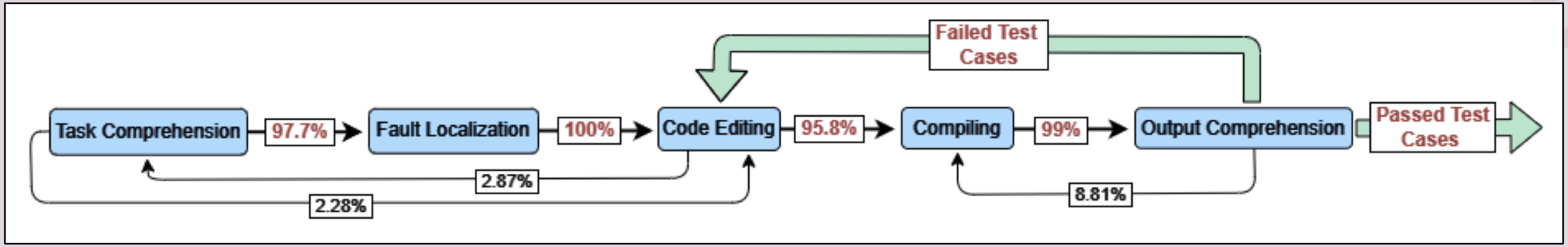
- All naming variations increase neural activity → higher cognitive load
- Single-morpheme identifiers trigger distinct activation → likely due to misleading semantics



Towards a Cognitive Model of Dynamic Debugging: Does Identifier Construction Matter?

In summary, **we use neuroimaging** of programmers (n=28) **to find**:

- **A cognitively and behaviorally backed dynamic model of debugging** RQ1



- Morpheme-varied identifiers have no significant impact on debugging performance, regardless of reading ability RQ2
- Morpheme-varied identifiers induce greater cognitive load than original RQ2
- Lack of expertise induces more cognitive load than reduced reading ability RQ3

Danniell Hu, **Priscila Santiesteban (pasanti@umich.edu)**, Madeline Endres, Westley Weimer

End.

Special Thanks To My Research Team !

Danniell Hu

PhD Student
University of Michigan

Priscila Santiesteban

PhD Candidate
University of Michigan

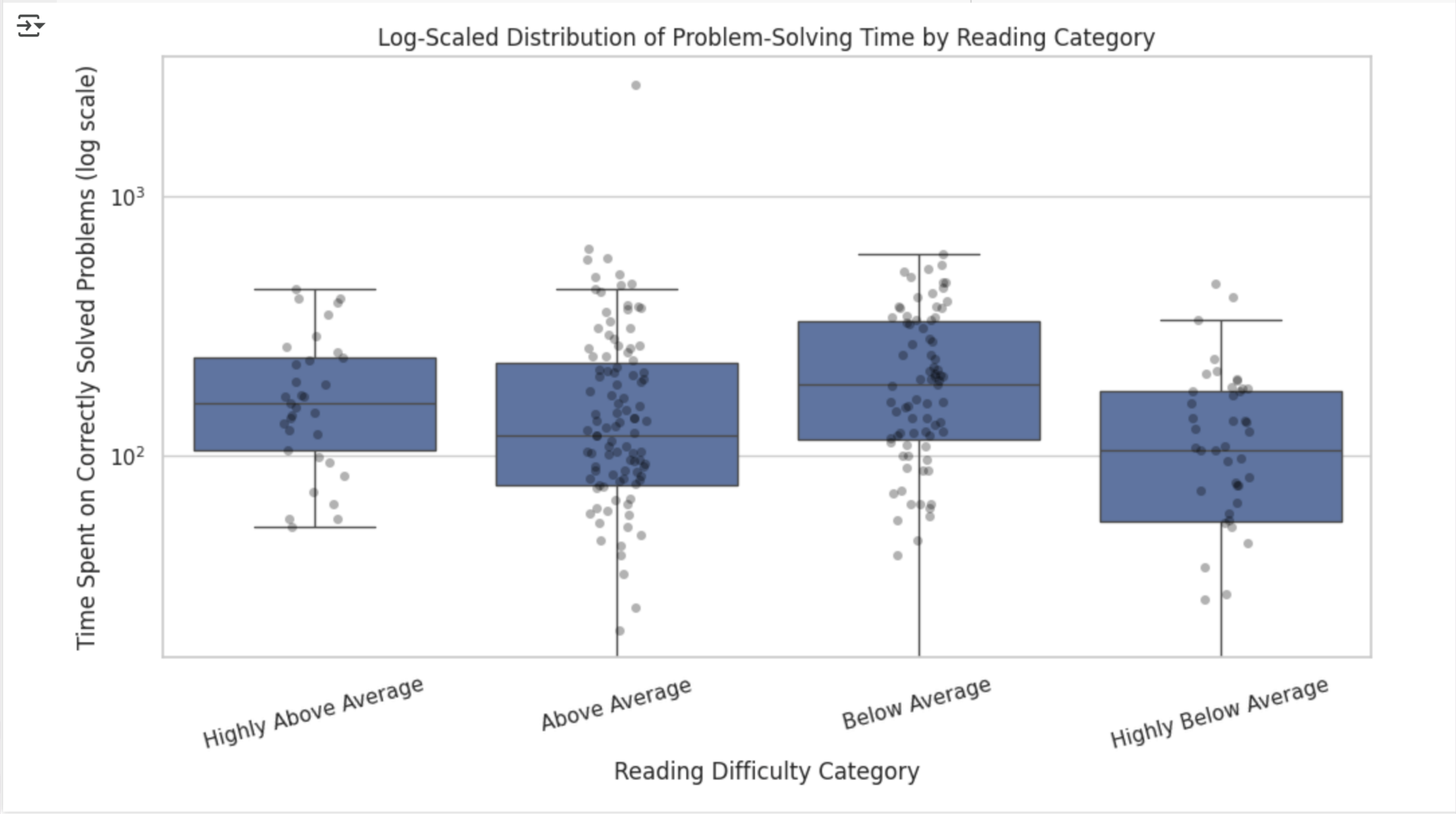
Madeline Endres

Assistant Professor
University of Massachusetts-Amherst

Westley Weimer

Professor
University of Michigan

Results – Reaction Time Distribution vs Reading Ability



Overall Stats:
Mean: 185.79s
Median: 138.23s

Results – Reaction Time Distribution vs Reading Ability

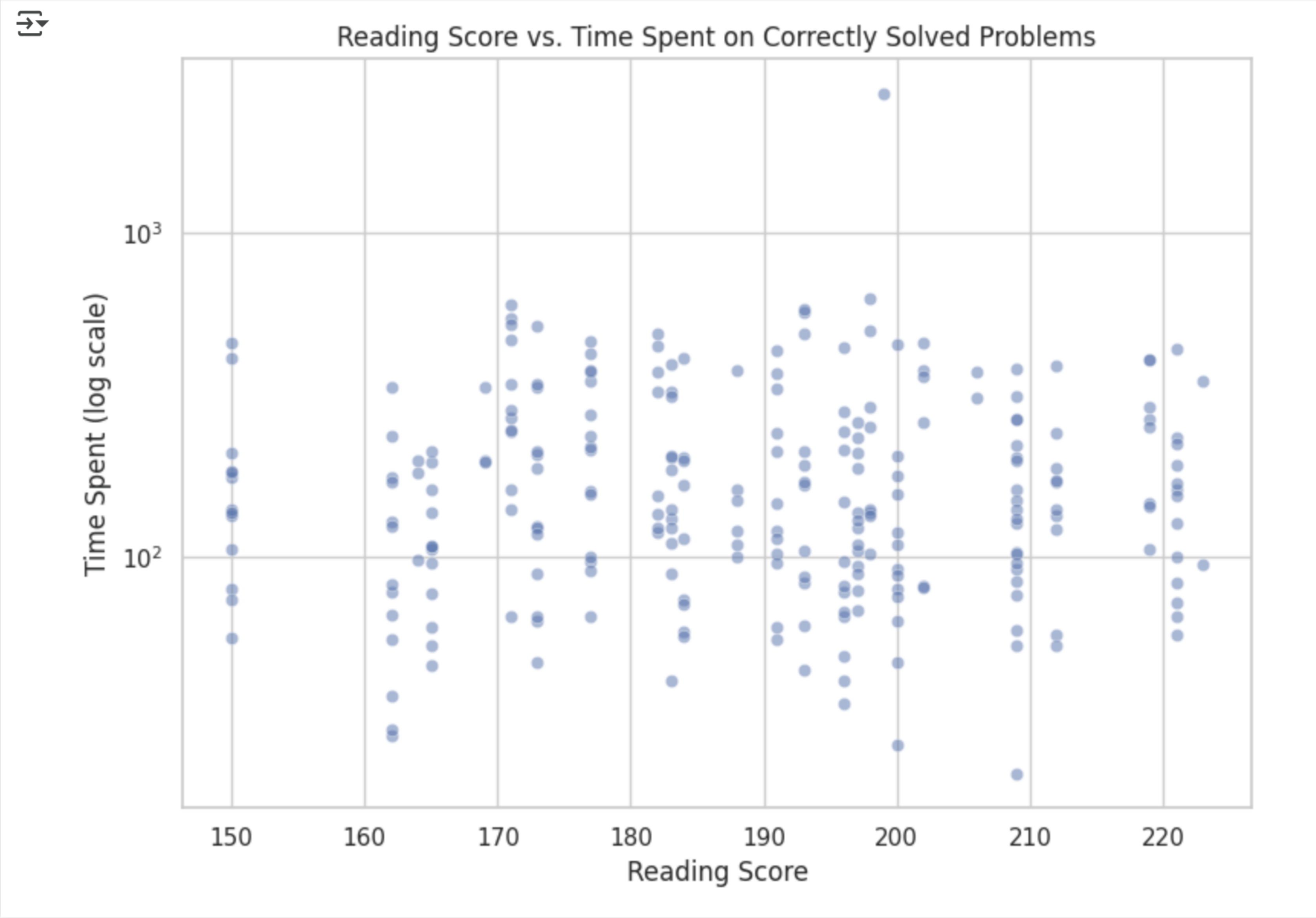
Overall Stats:
Mean: 185.79s
Median: 138.23s

Results – Reaction Time Distribution vs Reading Ability

Overall Stats:
Mean: 185.79s
Median: 138.23s

Categorical Reading	mean_time	median_time
Above Average	187.598468	119.50
Below Average	220.620633	188.32
Highly Above Average	187.665152	160.28
Highly Below Average	121.650213	105.20

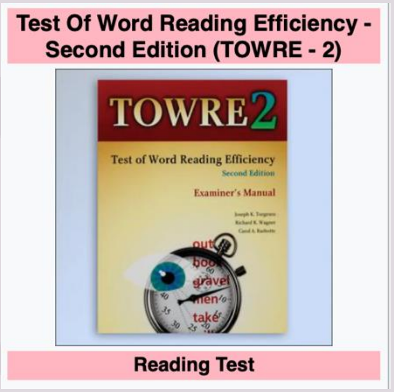
Results – Reaction Time Distribution vs Raw Reading Score



Experimental Design – Reading Assessment

We measured participants' English reading skills using the **TWRE**, a validated test widely used in research and practice.

- Lower TWRE scores indicate lower reading ability, while higher scores indicate stronger reading skills.



Decoding.

Nonsense words (try to read as many as possible within 45 seconds)

ip		din
ga		nup
ko		fet
ta		bave
om		pate
ig		herm
ni		dess
pim		chur
wum		knap