

Performance, Workload, Emotion, and Self-Efficacy of Novice Programmers Using AI Code Generation

Nicholas Gardella, Raymond Pettit, and Sara L. Riggs

Introduction

AI is great at simple programming tasks.

Default (GPT-3.5)



N


write a python script to read in a csv file that contains columns a,b,c and finds a grand sum of all the values of a,b, and c added together by row and then adding up all the rows



ChatGPT

```
import csv
# Initialize the variables to store the grand sum and row sums
grand_sum = 0
row_sums = []
# Specify the path to your CSV file
csv_file_path = 'your_csv_file.csv'
# Open the CSV file and read its contents
with open(csv_file_path, 'r') as csv_file:
    csv_reader = csv.reader(csv_file)
    # Skip the header row if it exists
    next(csv_reader, None)
    for row in csv_reader:
        try:
            # Assuming that columns a, b, and c are at index 0, 1, and 2 respectively
            a, b, c = map(float, row[0:3])
            # Calculate the sum of a, b, and c for this row
            row_sum = a + b + c
            # Add the row sum to the list of row sums
            row_sums.append(row_sum)
            # Add the row sum to the grand sum
            grand_sum += row_sum
        except ValueError:
            # Handle cases where the CSV contains non-numeric values
            print(f"Skipping invalid row: {row}")
# Calculate the grand total of all row sums
grand_total = sum(row_sums)
# Print the results
print(f"Grand Total of all rows: {grand_total}")
```

Novices can use high quality AI for free...



ChatGPT


Free

For individuals just getting started with ChatGPT


- ✓ Assistance with writing, problem solving and more
- ✓ Access to GPT-3.5
- ✓ Limited access to GPT-4o

Limited access to data analysis, file uploads, vision, web browsing, and custom GPTs

\$0 / month



Amazon Q Developer



Free

- ✓ Code faster with code suggestions in the IDE and CLI
- ✓ Free for public CLI completions
- ✓ Review code licenses with reference tracking
- ✓ Use where you work: your IDE, CLI, the AWS Management Console, Slack, and more



code llama



CONTINUE

The open-source autopilot for software development



tabnine

AI Assistant For Software Developers

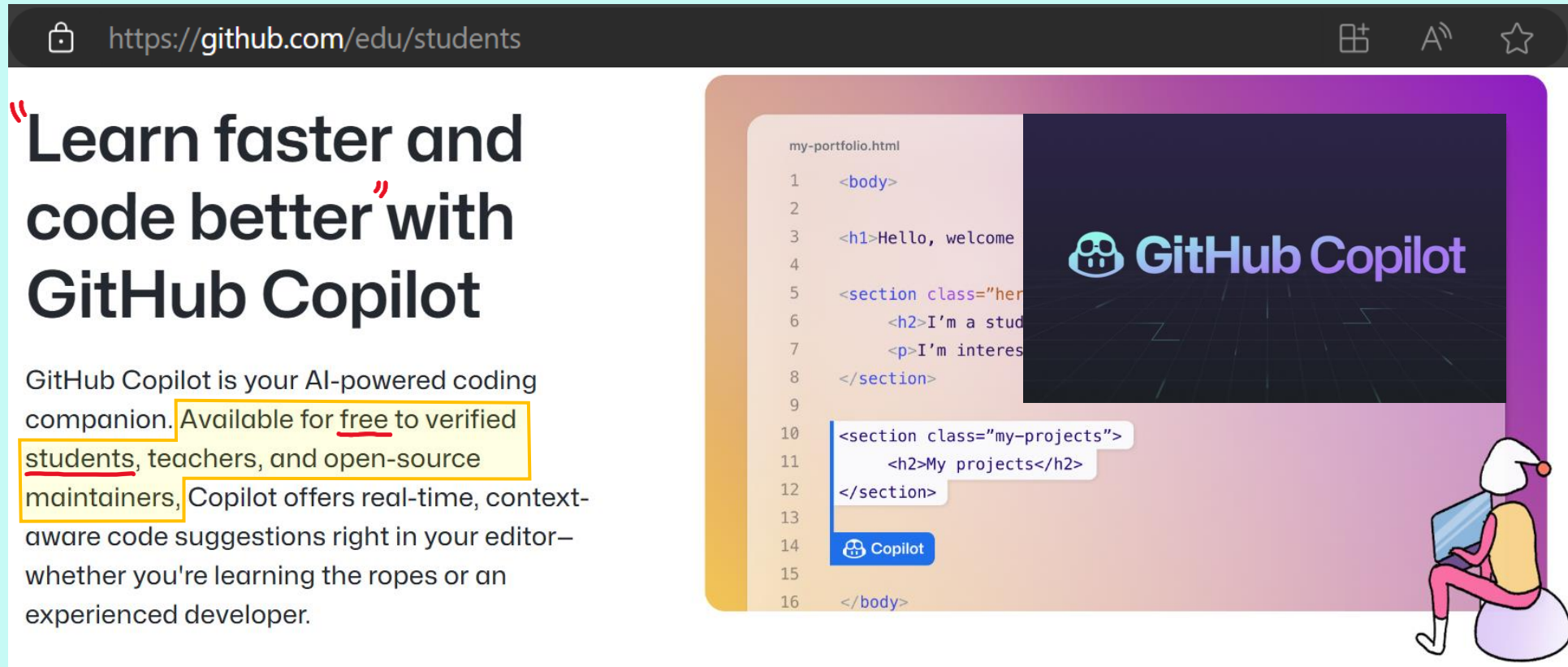
// **Basic** //

Our full-featured AI code assistant, completely free to use

- Basic AI code completions
- AI-powered chat using multiple models (may be rate limited)
- Support for all major IDEs

\$0 / {Free for everyone}

including the powerful GitHub Copilot AIDE




https://github.com/edu/students

“Learn faster and code better” with GitHub Copilot

GitHub Copilot is your AI-powered coding companion. Available for free to verified students, teachers, and open-source maintainers, Copilot offers real-time, context-aware code suggestions right in your editor—whether you're learning the ropes or an experienced developer.

```
my-portfolio.html
1 <body>
2
3 <h1>Hello, welcome
4
5 <section class="her
6 <h2>I'm a stud
7 <p>I'm interes
8 </section>
9
10 <section class="my-projects">
11 <h2>My projects</h2>
12 </section>
13
14 Copilot
15
16 </body>
```

GitHub Copilot



AIDE = Artificial Intelligence-driven Development Environment

But AI won't do students' work, right?

Default (GPT-3.5)

N

do my python programming hw for me



I can certainly help you with your Python programming homework by providing guidance, explanations, and code snippets. However, I can't complete your homework for you. If you have a specific question or problem related to your Python homework, please feel free to ask, and I'll do my best to assist you in understanding and solving it.



Unfortunately, it will...

see Chen et al. [8], Finnie-Ansley et al. [12], Kazemitabaar et al. [21]

The image shows a side-by-side comparison of a code editor and a web browser. The code editor on the left displays a Python file named `wof_if.py` with the following content:

```
17 will show to the lab TA. Spend at most 40 minutes for this loopless version before moving on to
18 the loop version.
19 2 Lab Description
20 For this lab, you will be creating two python files. One will be a loopless implementation of the
21 Wheel of Fortune game while the other will be a loop implementation of the game.
22 1. Create your loopless implementation in a file called wof_if.py.
23 2. Create a loop implementation in a file called wof_while.py.
24 3. Try the Extensions at the end of this lab page if you finish the loopless and looped
25 versions of the game before the lab ends.
26 Read the lab page in its entirety since there is helpful information on how to implement this
27 game.
28 """
29
30 """
31 3.1 Design
32 We need to keep track of two basic elements:
33  The word the user is trying to guess
34  The portion of the word they have guessed
35  How the two relate to each other
36 We can relate the two by storing the letters next to each other: the key, then a blank or their
37 guess; thus if the word is jellied, we'd start with j_e_l_l_i_e_d; after they guess e and d it
38 would be j_eel_l_i_eedd; etc.
39 We'll then proceed as follows:
40 1. ask them a letter
41 2. adjust the string, replacing letter-blank pairs with letter-letter pairs
42 3. display every other letter of the word so they see only the blanks and letters they guessed
43 We'll write
44 """
45
46 word = "jellied"
47
48
```

The web browser on the right displays a page titled "Lab 6 - Wheel of Fortune" with the following content:

3 Game Implementation

3.1 Design

We need to keep track of two basic elements:

- The word the user is trying to guess
- The portion of the word they have guessed
- How the two relate to each other

We can relate the two by storing the letters next to each other: the key, then a blank or their guess; thus if the word is jellied, we'd start with j_e_l_l_i_e_d; after they guess e and d it would be j_eel_l_i_eedd; etc.

We'll then proceed as follows:

- ask them a letter
- adjust the string, replacing letter-blank pairs with letter-letter pairs
- display every other letter of the word so they see only the blanks and letters they guessed

We'll write that code, copy-and-paste it a bunch of times, and we'll have basic Wheel of Fortune.

3.2 Implementation Pseudocode

Write the code to do one step of Wheel of Fortune: ask them the word to use, ask for one letter, and show the word with blanks and that one letter.

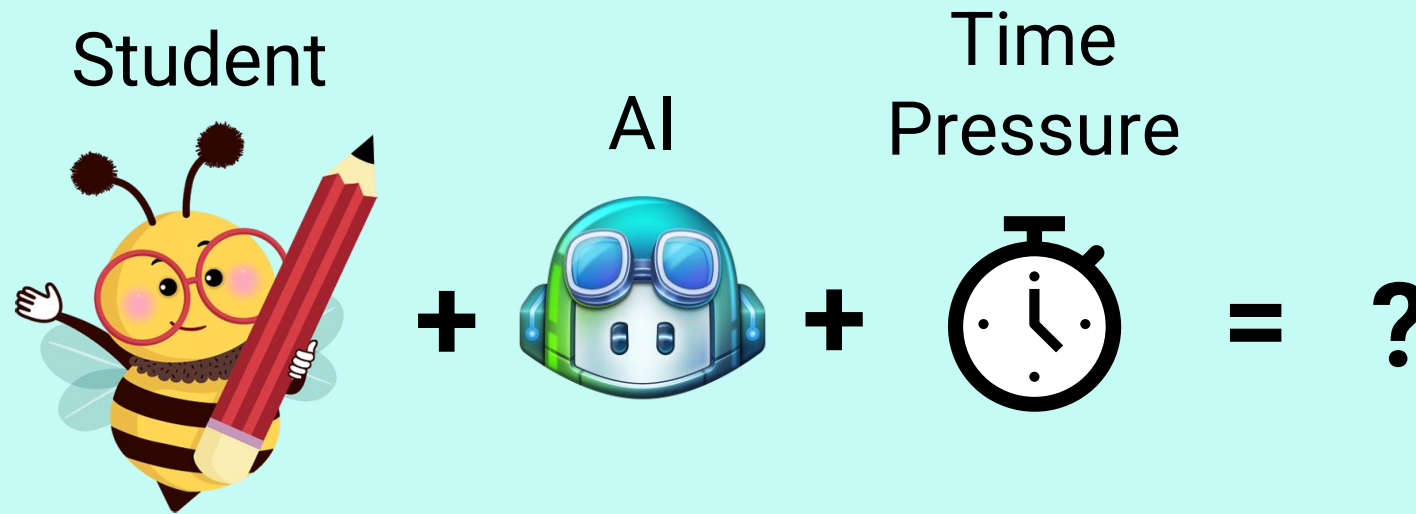
3.2.1 Initialization

Ask the user for the word. Make it all lower case, as Wheel of Fortune is traditionally case-insensitive.

Put an underscore after every letter. The `join` method of strings will do most of this; for example, try running `"Wow!".join("amazing")` and printing the result. What do you have to do to change this to insert underscores? One underscore will still be missing, but you can add it in another way you've been using for more than a month now...

3.2.2 Ask them a letter

So what is the impact of AI on a novice?



Literature at a glance: Performance



Objectively Measuring Performance

Vaithilingam & Glassman [35], Xu et al. [37]



Significant Performance *Benefit*

Kazemitabaar et al. [21]



Literature at a glance: Well-Being



Kazemitabaar et al. [21]

“slightly **less stressed**” ($p=.06$)

“**more eager** [...] to continue learning” ($p = .025$)

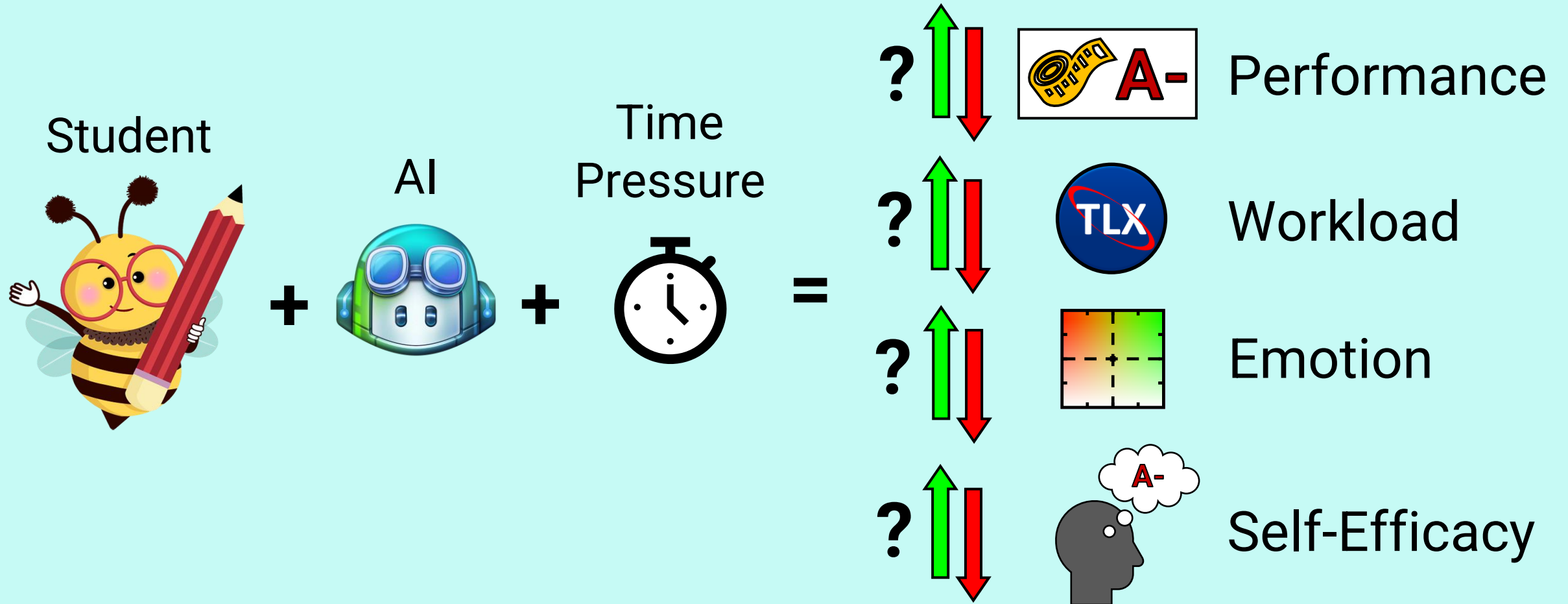
Vaithilingam & Glassman [35]

“Participants found code generated by Copilot **more helpful** than code generated by Intellisense” ($p < .001$)

Xu et al. [37]

“participants report having a **neutral** (15/31; 48.4%) **or** at least somewhat **positive** (15/31; 48.4%) experience”

But what measurable effects can we show?



Research Questions: How do AIDEs...

- **RQ1:** affect performance, workload, emotion, & self-efficacy of novice programmers under time pressure?
- **RQ2:** influence the effects of additional time spent programming on novices' performance and self-efficacy

Method

We recruited 17 students from a CS1 course.

- Ability reports (day of participation):

11 “novice”	6 “intermediate”
-------------	------------------

- Gender reports:

11 female	5 male	1	not shared
-----------	--------	---	------------

- Racial reports:

- Asian- 11
- White- 8
- Hispanic/Latinx- 2
- Black- 1

Tasks came from the HumanEval dataset.

The screenshot shows the GitHub repository page for 'openai / human-eval'. The repository is public and has 1.5k stars, 219 forks, and 122 watchers. The main content area displays a list of files and folders with their commit history:

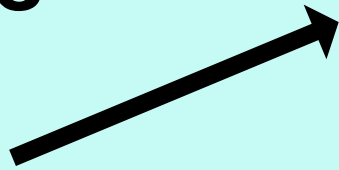
File/Folder	Commit Message	Commit Count	Time Ago
data	squash commits	5	2 years ago
human_eval	Fix type signature of read_problems (#9)	5	2 years ago
LICENSE	Add license file. (#2)	2	2 years ago
README.md	update readme	1	2 years ago
requirements.txt	squash commits	1	2 years ago
setup.py	squash commits	1	2 years ago

The right sidebar contains the 'About' section, which includes a description: 'Code for the paper "Evaluating Large Language Models Trained on Code"'. Other links in the sidebar include 'Readme', 'MIT license', 'Activity', '1.5k stars', '122 watching', '219 forks', and 'Report repository'.

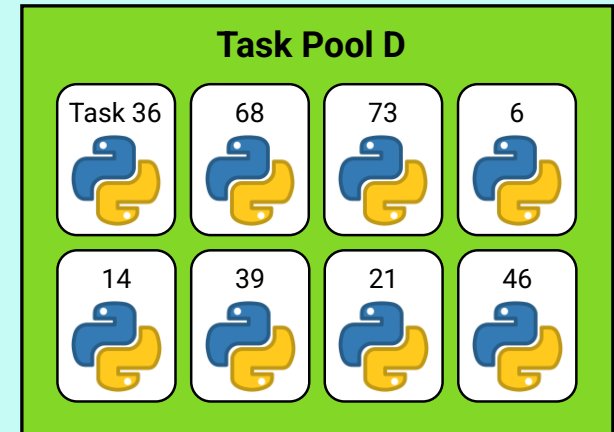
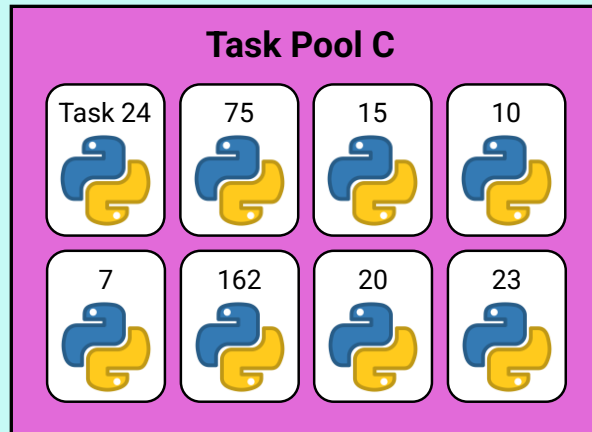
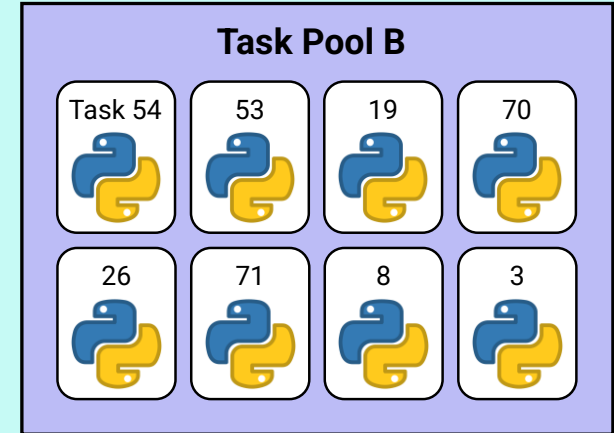
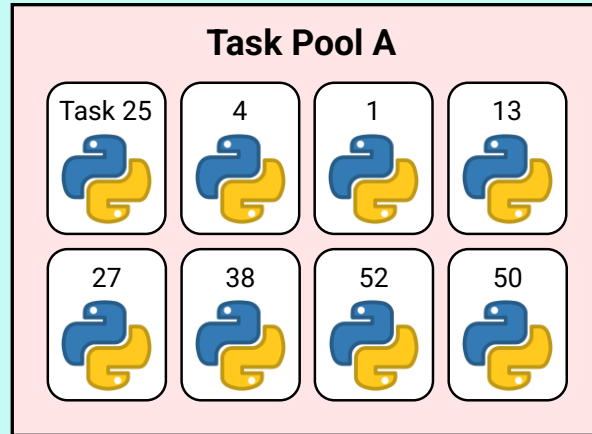
Chen et al. [8]

We created four pools of tasks.

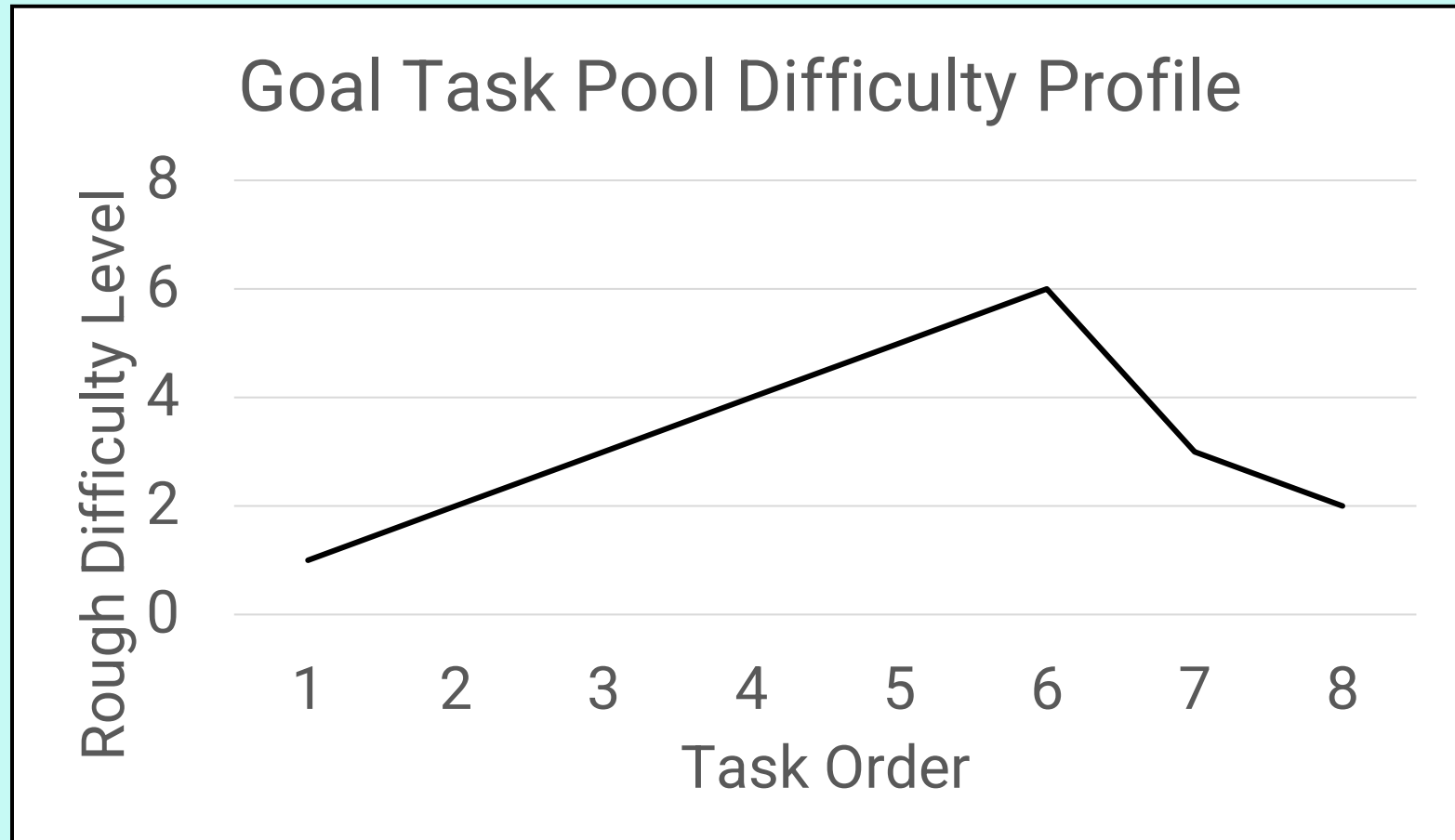
164 Tasks



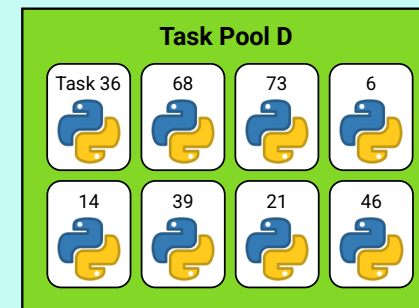
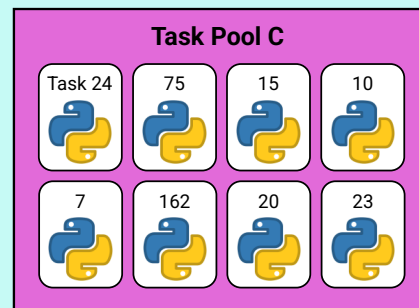
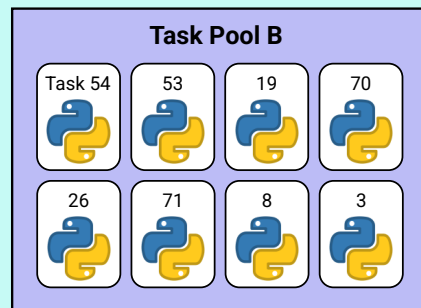
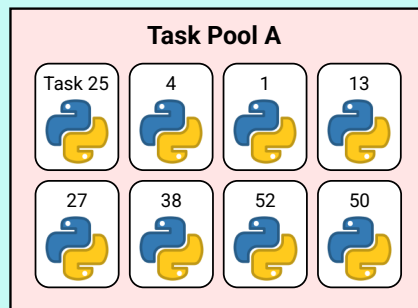
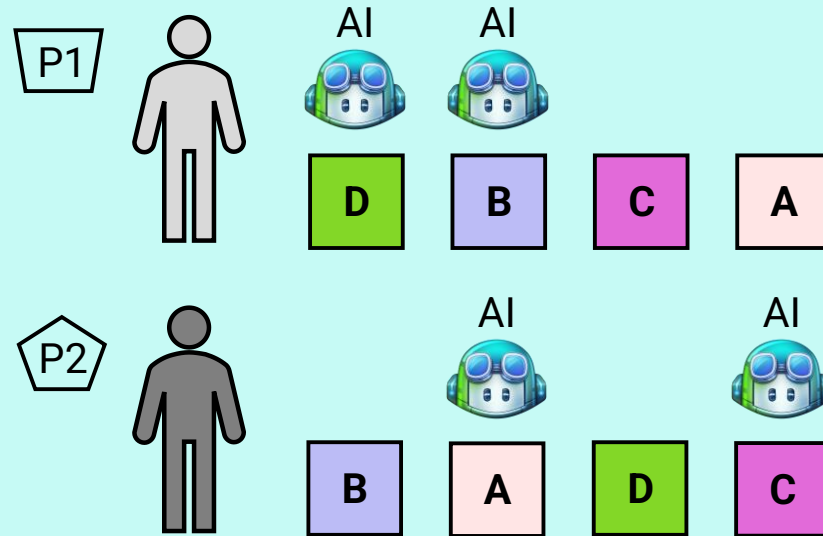
9 x



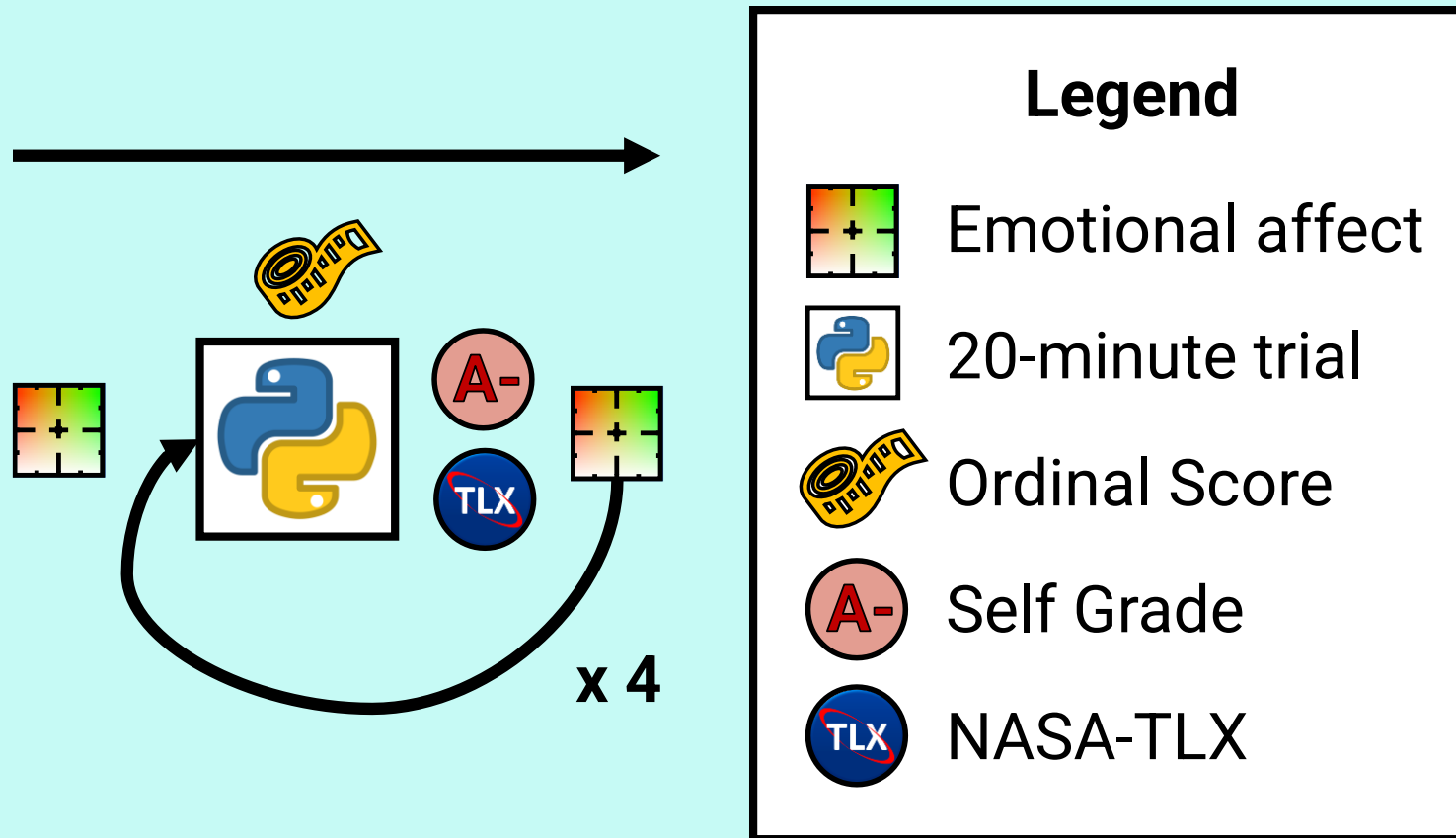
We placed easier tasks early in each pool.



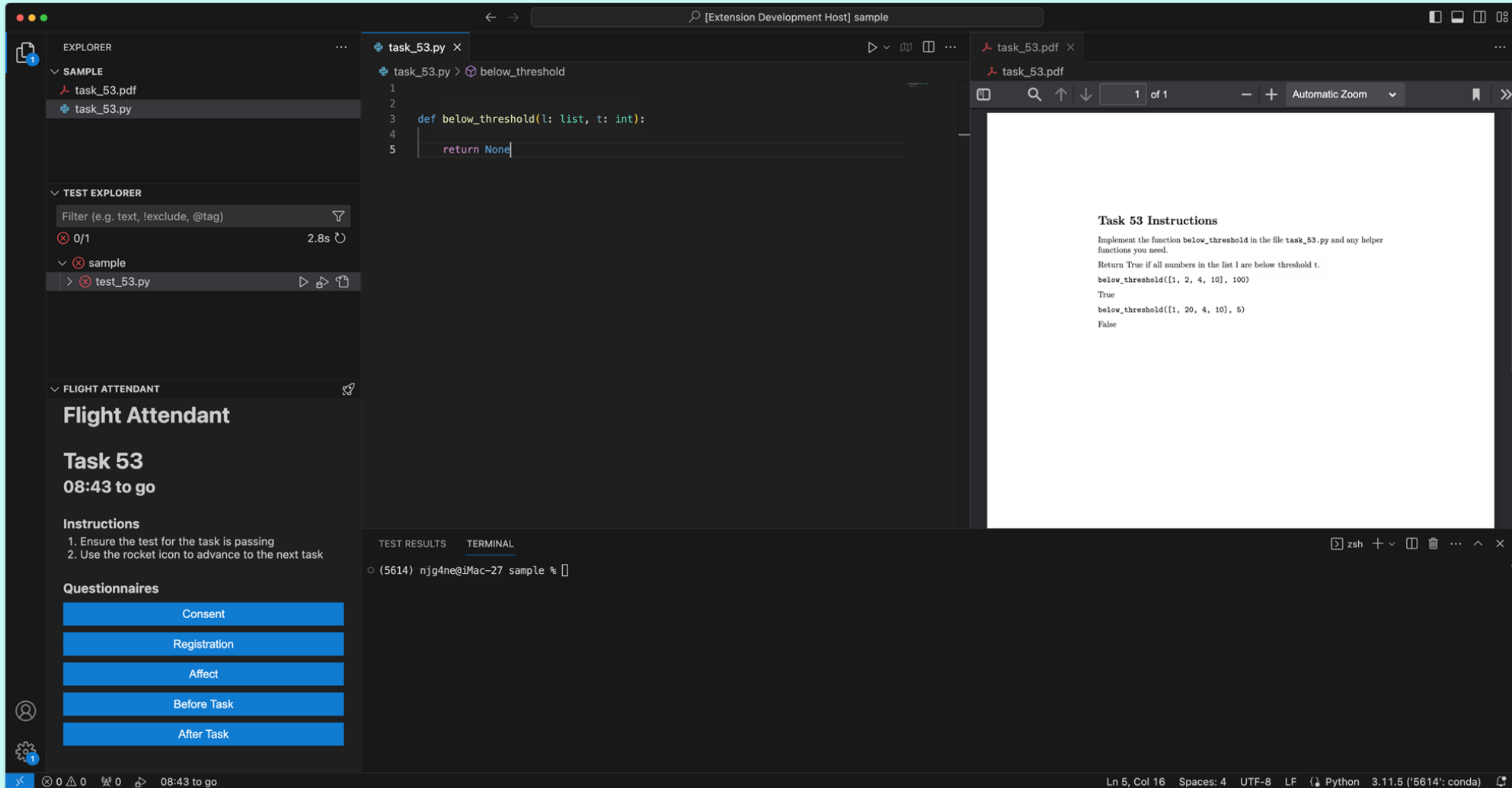
We used a within-subjects design.



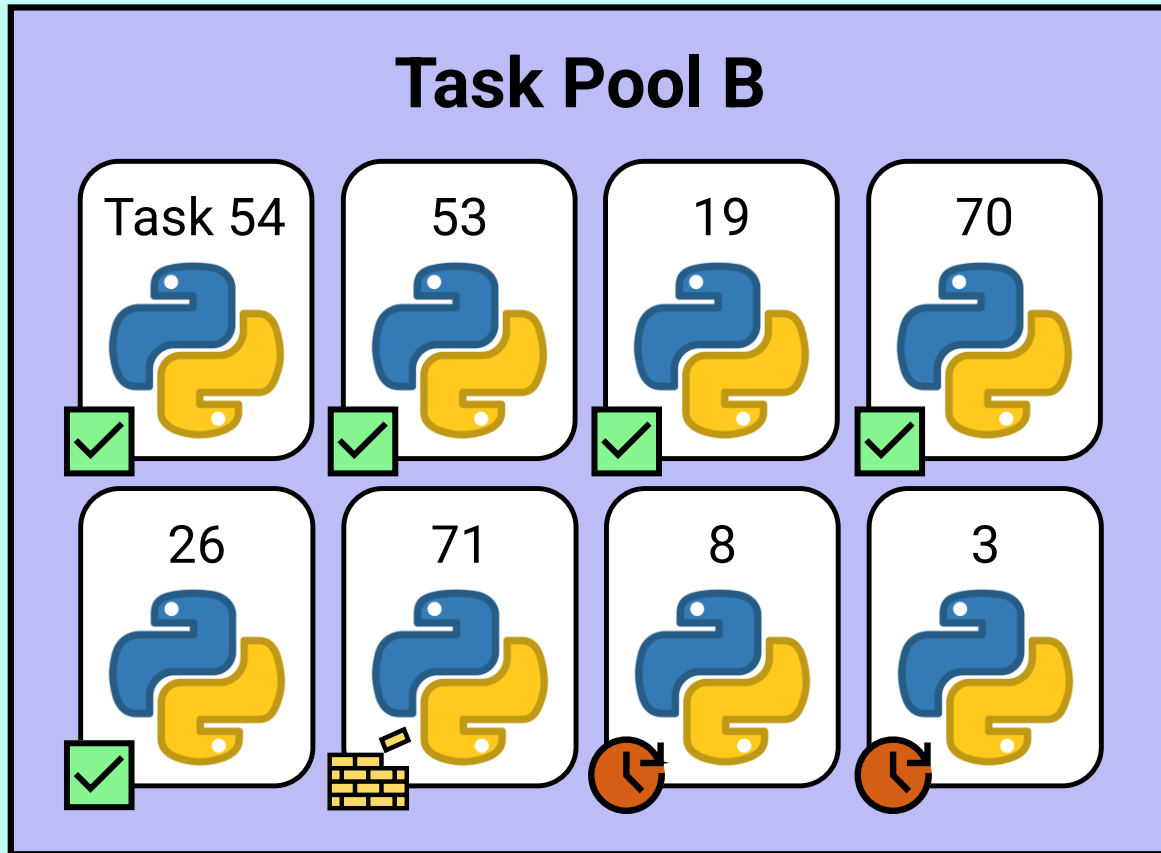
We took repeated measures four times.



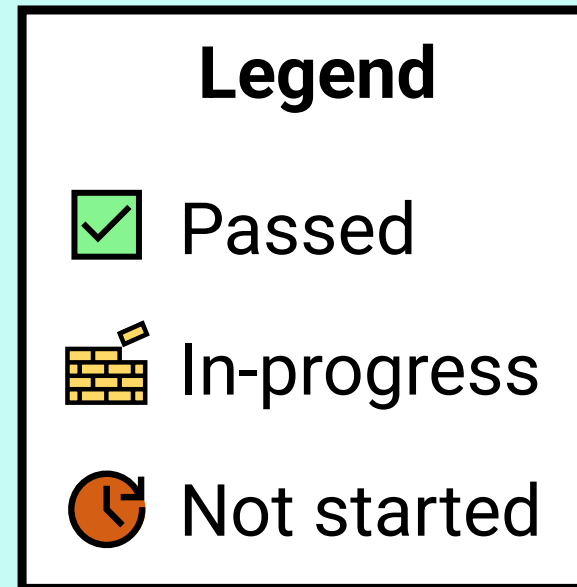
Participants worked in Visual Studio Code.



We scored task pools ordinally.




Score = 5





We measured workload with NASA-TLX.

Use the slider bars to estimate your mental, physical, and temporal demand on the challenge you just completed.

Very Low Very High


Mental Demand: How mentally demanding was the challenge?


Physical Demand: How physically demanding was the challenge?


Temporal Demand: How hurried or rushed was the pace of the challenge?



Use the slider bar to rate your own performance on the challenge you just completed. **Note that perfect is on the left, while failure is on the right.**


Perfect Failure

Performance: How successful were you in accomplishing what you were asked to do?


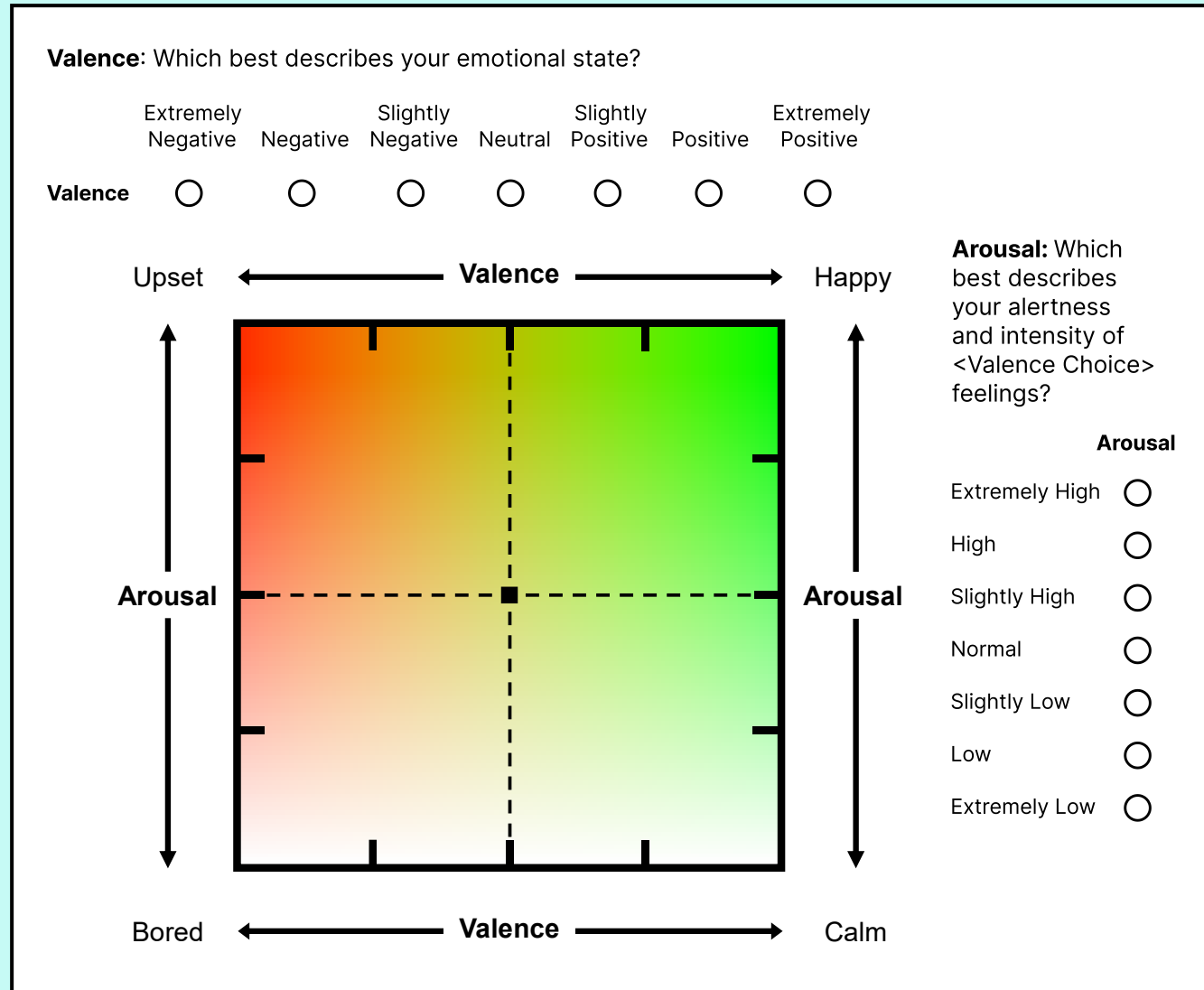
Use the slider bar to rate your levels of effort and frustration on the challenge you just completed.

Very Low Very High

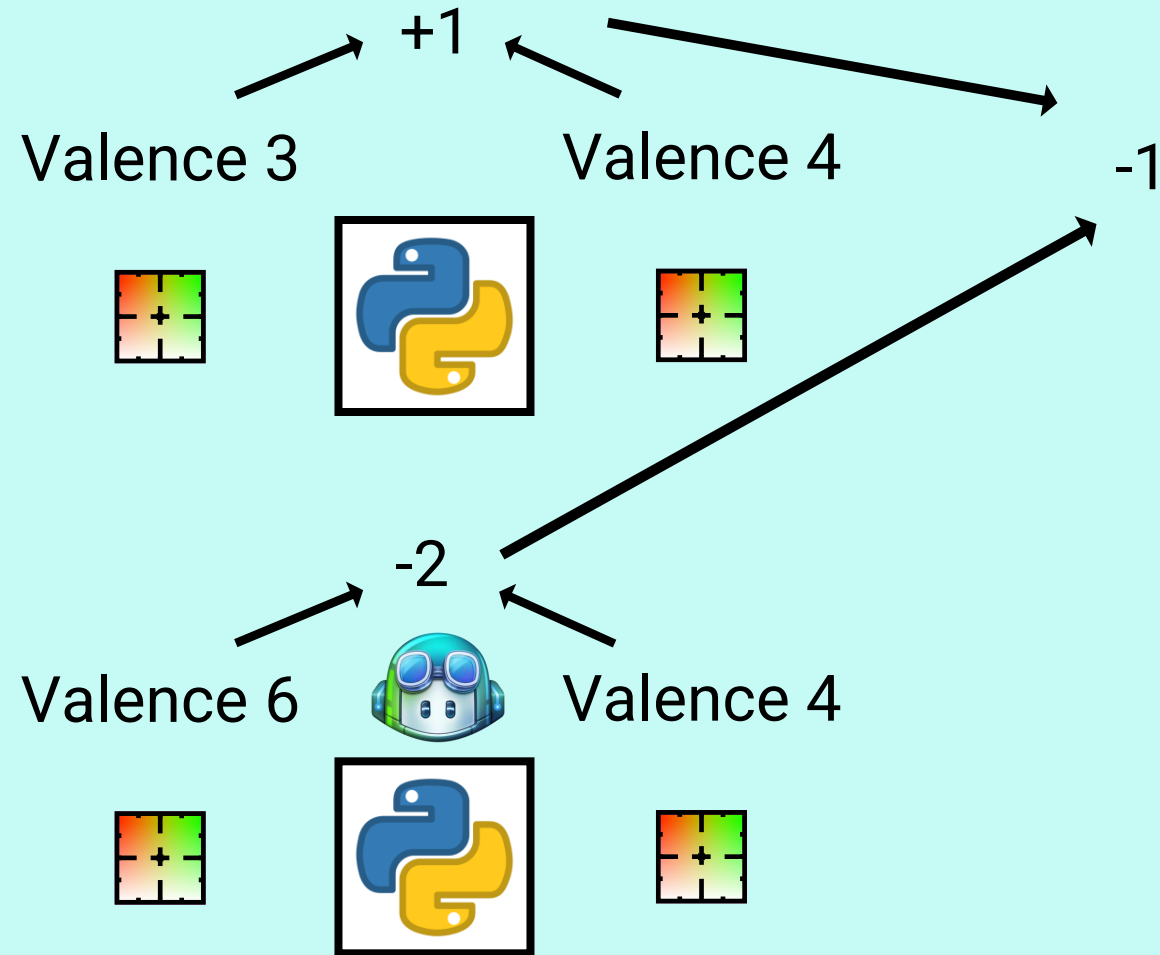
Effort: How hard did you have to work to accomplish your level of performance?


Frustration: How insecure, discouraged, irritated, stressed, or annoyed were you?


We measured emotion on two dimensions.




We measured **changes** in emotional state.



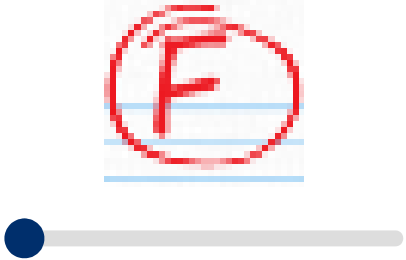
We measured self-efficacy with a letter grade.

If you had to give **yourself** a grade for your work on the last challenge, what would it be?



A hand-drawn red 'A-' grade is shown on a piece of lined paper. Below the paper is a horizontal slider with a blue dot positioned at approximately 75% of the way across.

If you had to give **GitHub Copilot** a grade for its contribution to the last challenge, what would it be?



A hand-drawn red 'F' grade is shown on a piece of lined paper. Below the paper is a horizontal slider with a blue dot positioned at approximately 10% of the way across.


Grade	Value
A+	13
A	12
A-	11
B+	10
...	...
D	3
D-	2
F	1

What percentage of the work for the challenge do you think **GitHub Copilot** was responsible for?

0% 100%

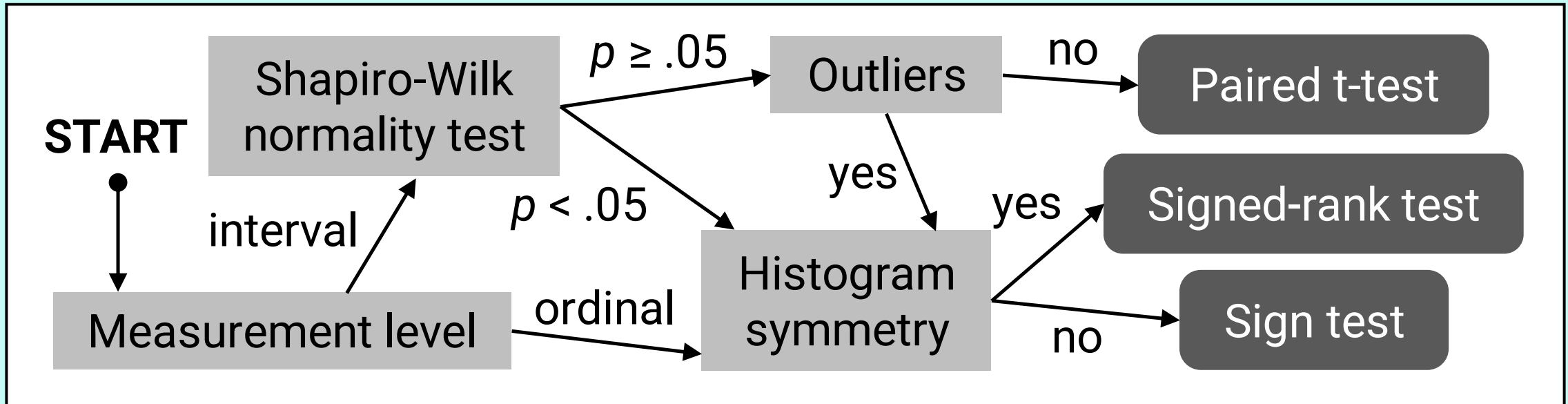
% of Work done by Copilot

75



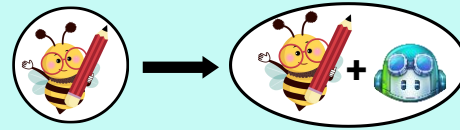
A horizontal progress bar is shown, with a dark blue segment representing 75% of the total length. A small box above the bar contains the number '75'.












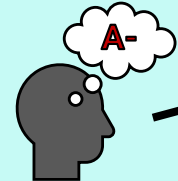


We took a conservative statistical approach.



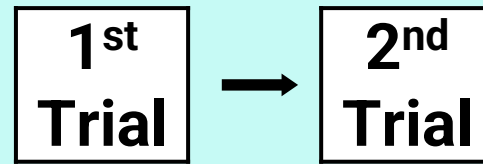
Results



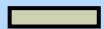







RQ1 Results



		Change	<i>p</i>	
	Score	 $\widetilde{\Delta x} = 1$.001 *	Be more successful
	Δ Valence	 $\overline{\Delta x} = 0.206$.436	Feel better
	Δ Arousal	 $\widetilde{\Delta x} = -.5$.191	Feel more relaxed
	TLX Mental	 $\overline{\Delta x} = -8.971$.024 *	Think less hard
	TLX Physical	 $\widetilde{\Delta x} = 0$.773	Exert no differently
	TLX Temporal	 $\overline{\Delta x} = -5.294$.200	Feel less rushed
	TLX Performance	 $\overline{\Delta x} = -6.029$.230	Feel more successful
	TLX Effort	 $\widetilde{\Delta x} = -5$.043 *	Try less hard
	TLX Frustration	 $\overline{\Delta x} = -2.206$.581	Feel less frustrated
	Self-Grade	 $\overline{\Delta x} = .265$.455	Feel more successful

RQ2 Results



		Change	p
Solo 	 Score	 $\widetilde{\Delta x} = 0$	1.000
	 Self-Grade	 $\widetilde{\Delta x} = 0$.594
with AI 	 Score	 $\widetilde{\Delta x} = 1$	0.046*
	 Self-Grade	 $\overline{\Delta x} = 1.588$	0.021*

Plateau over time

Feel the plateau

Improve over time

Feel the improvement

Discussion

Internet may explain performance differences.



Internet



Objectively Measuring Performance

Vaithilingam & Glassman [35], Xu et al. [37]



Internet



Significant Performance *Benefit*

Kazemitabaar et al. [21]

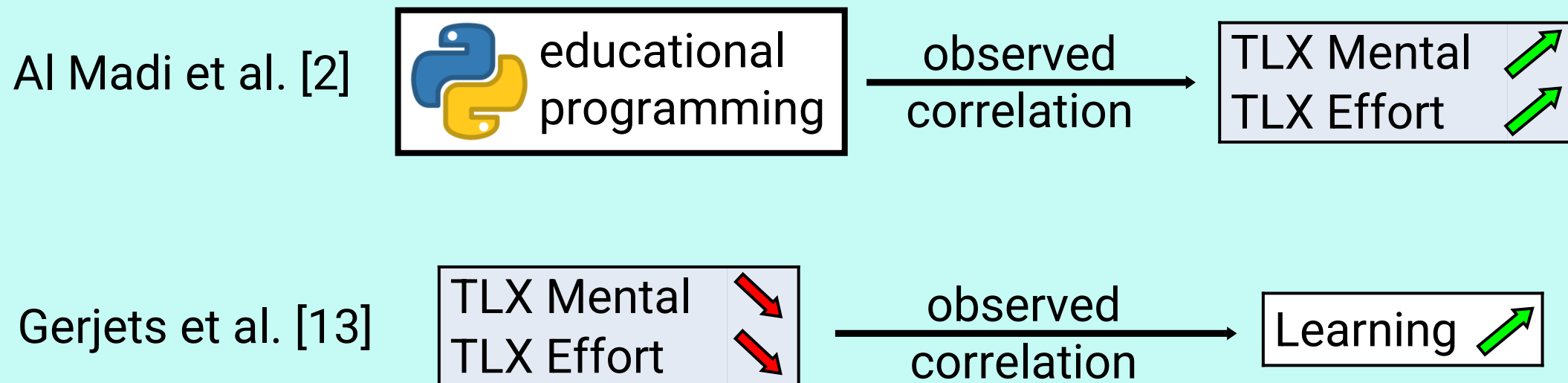


Internet

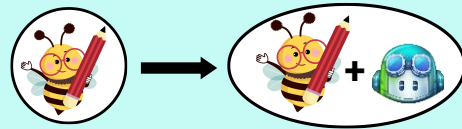


Our study

Workload decreases could aid learning.



AI may not make you feel much better.



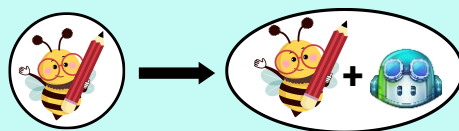
		Change	<i>p</i>
Δ Valence	↗	$\overline{\Delta x} = 0.206$.436
Δ Arousal	↘	$\widetilde{\Delta x} = -.5$.191
TLX Frustration	↘	$\overline{\Delta x} = -2.206$.581



Feel better

Feel more relaxed

Feel less frustrated

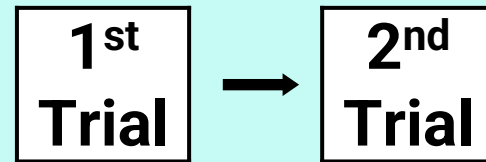
Novices don't take credit for AI's work.








	Change	<i>p</i>
TLX Performance	 $\bar{\Delta x} = -6.029$.230
Self-Grade	 $\bar{\Delta x} = .265$.455

Feel more successful

Novices take credit when they use AI better.



			Change	<i>p</i>	
with AI 	 Score		$\widetilde{\Delta x} = 1$	0.046*	Improve over time
	 Self-Grade		$\overline{\Delta x} = 1.588$	0.021*	Feel the improvement

Conclusion

Ecological and external validity are concerns.



Training Data

contained? →

Tasks from OpenAI HumanEval Dataset

Task 25	4	1	13	Task 54	53	19	70
27	38	52	50	26	71	8	3
Task 24	75	15	10	Task 36	68	73	6
7	162	20	23	14	39	21	46

n = 17 all from



UNIVERSITY
of VIRGINIA



Internet

Key Takeaways

- Novice programmers...
 - can **be more successful** with AI
 - can **think and try less hard** with AI
 - can **improve** at using AI **over time** and **take credit for the improvement**
- But...
 - **don't take credit** for AI's work
 - **don't feel much better** using AI

Nicholas Gardella, Raymond Pettit and Sara L. Riggs. 2024. Performance, Workload, Emotion, and Self-Efficacy of Novice Programmers Using AI Code Generation. In Proceedings of the 2024 Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), July 8- 10,2024, Milan, Italy. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3649217.3653615>



Contact: njg4ne@virginia.edu

Website: <https://n.gardella.cc>

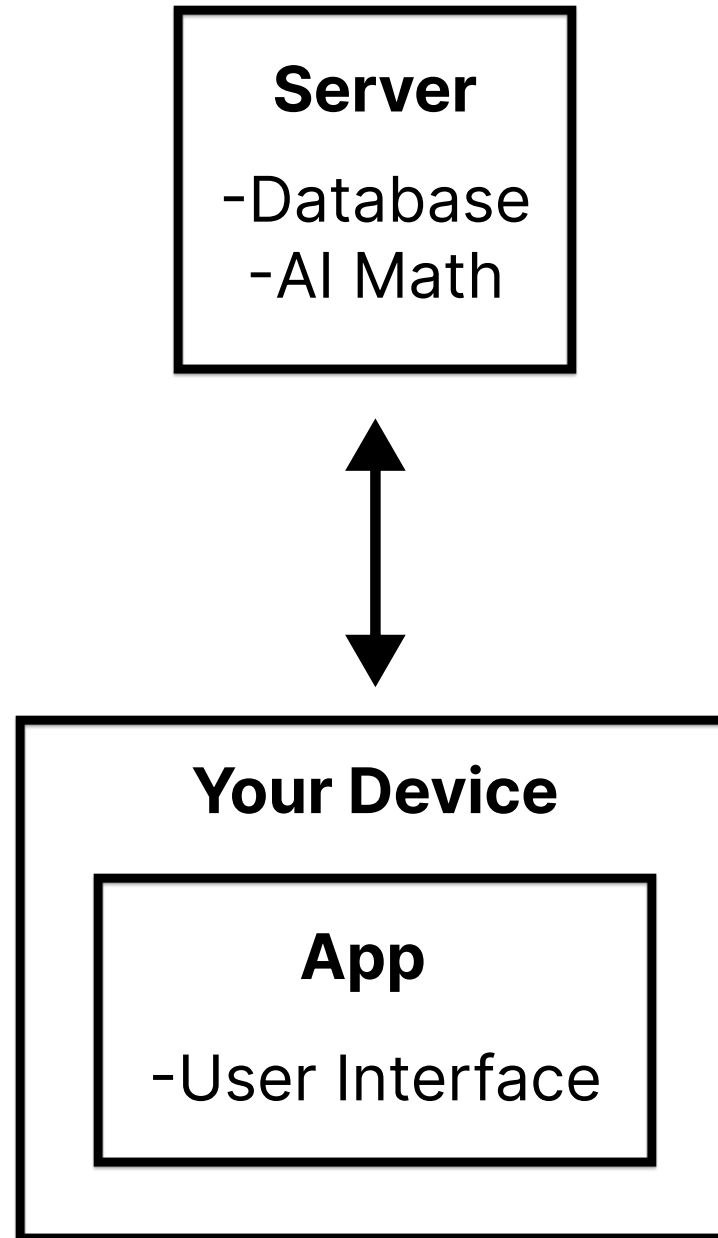
Funding support:

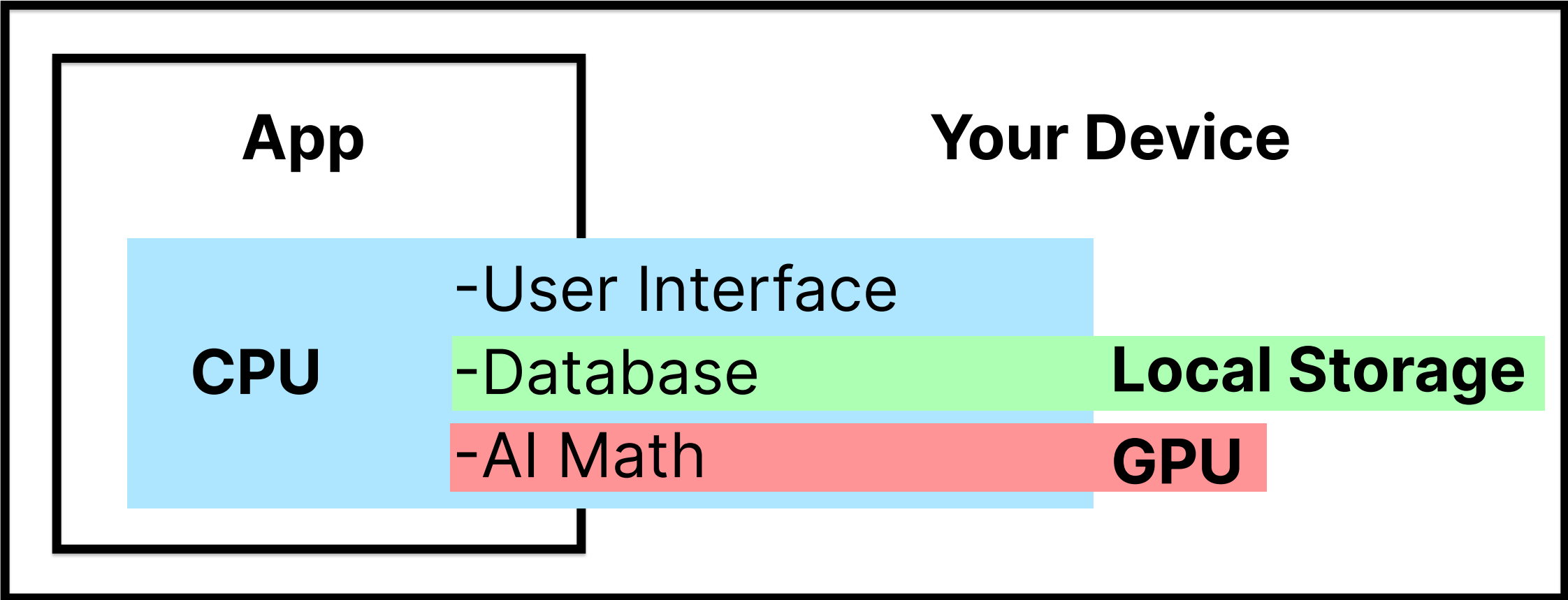
- UVA Distinguished Fellowship
- NSF NRT 1842490
- NSF GRFP 1829004
- Virginia CCI CVN GR102900

We thank Dylan Do, Zack Landsman, Michael Lin, Alex Myers, Eric Nguyen, Jacob Rice, Allison Sawyer, Olivia Seto, and Alice Warner for assistance in selecting and arranging HumanEval tasks.



Appendix





Web Browser

Your Device


Thread 1 -User Interface **CPU**

Thread 2 -Database **OPFS Storage**

Thread 3 -AI Math **GPU**

Transformers.js: State-of-the-art Machine Learning for the web



Chrome for Dev... 
764K subscribers

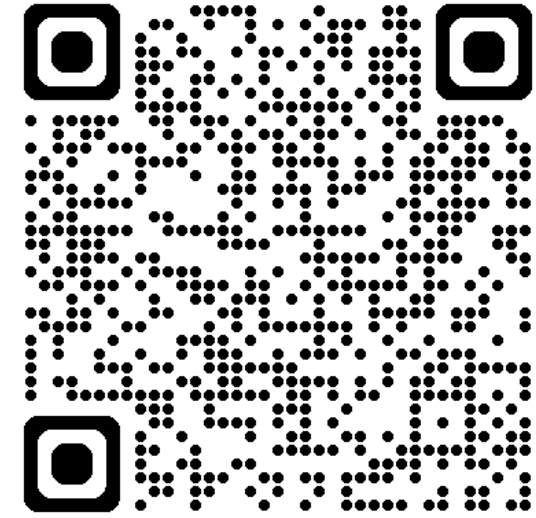
Subscribe

 369

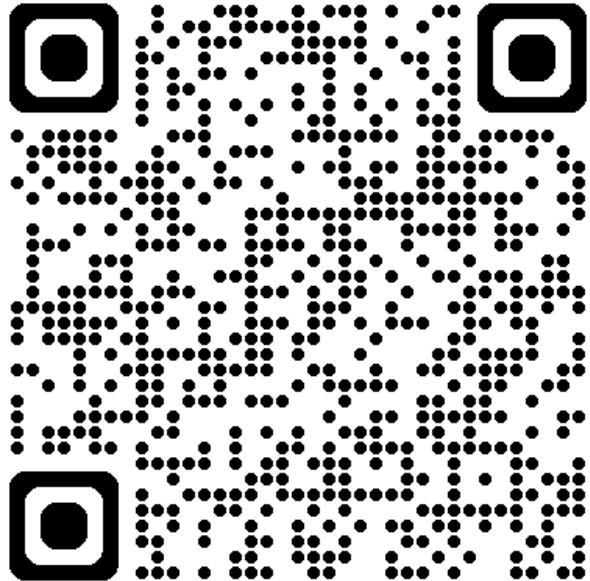


 Share

6,684 views 21 hours ago



<https://n.gardella.cc/rag/>




AI for Coding: In-Browser LLMs & Continue.dev x Ollama

 n

niqwis

Analytics

Edit video

 0

16 views Streamed live on Oct 22, 2024

Table 1

	Solo	with AI	Change
Score	$\tilde{x} = 2$	$\tilde{x} = 2.5$	$\widetilde{\Delta x} = 1$
Δ Valence	$\bar{x} = -.088$	$\bar{x} = .118$	$\overline{\Delta x} = 0.206$
Δ Arousal	$\tilde{x} = 0$	$\tilde{x} = -.5$	$\widetilde{\Delta x} = -.5$
TLX Mental	$\bar{x} = 58.382$	$\bar{x} = 49.412$	$\overline{\Delta x} = -8.971$
TLX Physical	$\tilde{x} = 5$	$\tilde{x} = 2.5$	$\widetilde{\Delta x} = 0$
TLX Temporal	$\bar{x} = 54.559$	$\bar{x} = 49.265$	$\overline{\Delta x} = -5.294$
TLX Performance	$\bar{x} = 61.912$	$\bar{x} = 55.882$	$\overline{\Delta x} = -6.029$
TLX Effort	$\tilde{x} = 62.5$	$\tilde{x} = 55$	$\widetilde{\Delta x} = -5$
TLX Frustration	$\bar{x} = 47.059$	$\bar{x} = 44.853$	$\overline{\Delta x} = -2.206$
Self-Grade	$\bar{x} = 7.647$	$\bar{x} = 7.912$	$\overline{\Delta x} = .265$



Table 2

	Statistic	<i>p</i>	<i>d</i>
Score	z Wilcoxon = 3.012	.001 *	.368
Δ Valence	$t(16) = .800$.436	.098
Δ Arousal	z Wilcoxon = -1.329	.191	$-.162$
TLX Mental	$t(16) = -2.487$.024 *	$-.304$
TLX Physical	z Sign Test = $-.289$.773	$-.035$
TLX Temporal	$t(16) = -1.338$.200	$-.163$
TLX Performance	$t(16) = -1.248$.230	$-.152$
TLX Effort	z Wilcoxon = -2.023	.043 *	$-.247$
TLX Frustration	$t(16) = -.563$.581	$-.069$
Self-Grade	$t(16) = .765$.455	.093



Tables 3 & 4

		1st Trial	2nd Trial	Change
Solo	Score	$\tilde{x} = 2$	$\tilde{x} = 2$	$\widetilde{\Delta x} = 0$
	Self-Grade	$\tilde{x} = 8$	$\tilde{x} = 7.25$	$\widetilde{\Delta x} = 0$
with AI	Score	$\tilde{x} = 2$	$\tilde{x} = 3$	$\widetilde{\Delta x} = 1$
	Self-Grade	$\bar{x} = 7.118$	$\bar{x} = 8.706$	$\overline{\Delta x} = 1.588$

		Statistic	<i>p</i>	<i>d</i>
Solo	Score	$z_{\text{Wilcoxon}} = .090$	1.000	.016
	Self-Grade	$z_{\text{Wilcoxon}} = -.517$.594	-.091
with AI	Score	$z_{\text{Wilcoxon}} = 2.029$	0.046*	.348
	Self-Grade	$t(16) = 2.567$	0.021*	.440

