Live Exploration of AI-Generated Programs

Kasra Ferdowsi

with Ruanqianqian (Lisa) Huang*, Michael B. James, Nadia Polikarpova, and Sorin Lerner

UC San Diego

Overview

1. Motivation & Background:
    a. Grounded Copilot
    b. Live Programming
2. LEAP: Live Exploration of AI-Generated Code
3. User Study
4. Findings
    a. Validating suggestions
    b. Over-/Under-reliance
    c. Cognitive Load
    d. Impressions

Slides based on Polikarpova 2023.

Overview

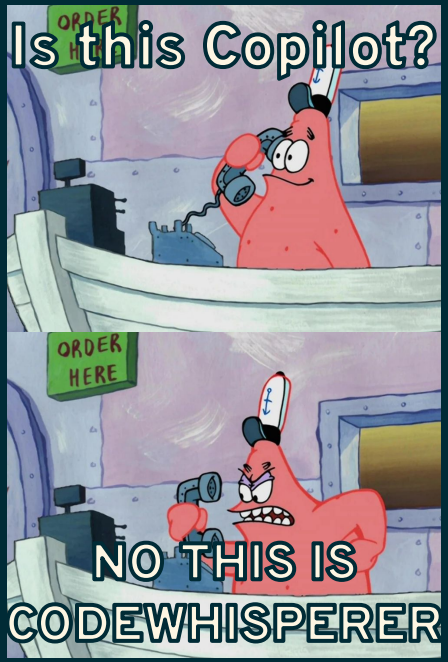
1. Motivation & Background:
    a. Grounded Copilot
    b. Live Programming
2. LEAP: Live Exploration of AI-Generated Code
3. User Study
4. Findings
    a. Validating suggestions
    b. Over-/Under-reliance
    c. Cognitive Load
    d. Impressions

Slides based on Polikarpova 2023.

LLMs for Code Generation*

GitHub
Copilot

OpenAI
ChatGPT



* For experienced programmers.

Background

1. Grounded Copilot: How Programmers Interact with Code-Generating Models.
   Shraddha Barke, Michael B. James, and Nadia Polikarpova. 2023.
2. Understanding the Usability of AI Programming Assistants.
   Jenny T. Liang, Chenyang Yang, and Brad A. Myers. 2023.
3. Reading Between the Lines: Modeling User Behavior and Costs in AI-Assisted Programming.
   Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. 2022.
4. Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models.
   Priyan Vaithilingam, Tianyi Zhang, and Elena Glassman. 2022.

Background

In summary, programmers using AI-generated code…

- Spend significant time *validating* code suggestions,
- Have trouble evaluating the correctness of generated code,
- Choose validation strategies based on *time cost*, and so
- Both *under- and over-rely* on AI code suggestions.

Background

"User interactions with Copilot can be classified into two modes—***acceleration*** and ***exploration***—akin to the two systems of thought in dual-process theories of cognition"*

* Grounded Copilot: How Programmers Interact with Code-Generating Models

Shraddha Barke, Michael B. James, and Nadia Polikarpova. 2023.

Grounded Copilot

Acceleration	vs.	Exploration
unintentional	**Prompting**	intentional with comments / invoke side panel
"pattern matching"	**Validation**	explicit validation via elimination / execution / documentation
unit of focus (sub-expression / statement)	**Scope**	entire function + multiple alternatives
unwilling to edit	**Mismatch Tolerance**	willing to edit / debug / "rip apart" / cherry-pick

Grounded Copilot

Acceleration	vs.	Exploration
unintentional	**Prompting**	intentional with comments / invoke side panel
"pattern matching"	**Validation**	explicit validation via elimination / execution / documentation
unit of focus (sub-expression / statement)	**Scope**	entire function + multiple alternatives
unwilling to edit	**Mismatch Tolerance**	willing to edit / debug / "rip apart" / cherry-pick


Grounded Copilot
Acceleration
unintentional
"pattern matching"
unit of focus
(sub-expression / statement)
unwilling to edit
vs.
Prompting
Validation
Scope
Mismatch
Tolerance

Live Programming

Demo!

Research Question:

Does *Live Programming* offer a good interaction for *exploring* AI-generated code?

Overview

1. Motivation & Background:
    a. Grounded Copilot
    b. Live Programming
2. LEAP: Live Exploration of AI-Generated Code
3. User Study
4. Findings
    a. Validating suggestions
    b. Over-/Under-reliance
    c. Cognitive Load
    d. Impressions

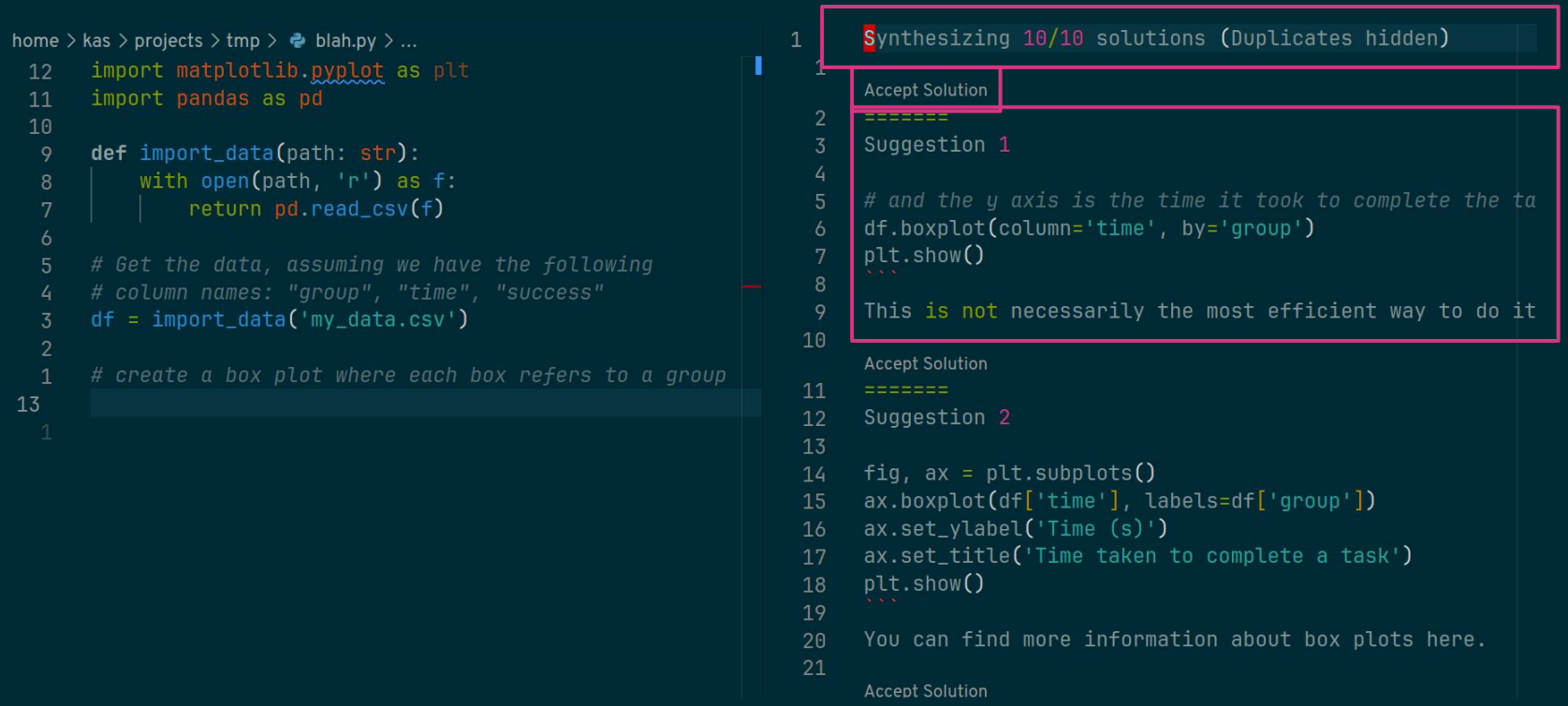
Slides based on Polikarpova 2023.

Overview

1. Motivation & Background:
    a. Grounded Copilot
    b. Live Programming
2. LEAP: Live Exploration of AI-Generated Code
3. User Study
4. Findings
    a. Validating suggestions
    b. Over-/Under-reliance
    c. Cognitive Load
    d. Impressions

Slides based on Polikarpova 2023.

Copilot's Side Panel



Demo!

Overview

1. Motivation & Background:
    a. Grounded Copilot
    b. Live Programming
2. LEAP: Live Exploration of AI-Generated Code
3. User Study
4. Findings
    a. Validating suggestions
    b. Over-/Under-reliance
    c. Cognitive Load
    d. Impressions

Slides based on Polikarpova 2023.

Overview

1. Motivation & Background:
    a. Grounded Copilot
    b. Live Programming
2. LEAP: Live Exploration of AI-Generated Code
3. User Study
4. Findings
    a. Validating suggestions
    b. Over-/Under-reliance
    c. Cognitive Load
    d. Impressions

Slides based on Polikarpova 2023.

Research Questions

How does Live Programming effect...

1. Code Correctness
2. Over-/Under-Reliance on AI
3. Cognitive Load
4. Users' Impressions

Experimental Conditions

no-PB

AI suggestions

+

Manually Invoked
Terminal Output

PB

AI Suggestions

+

Projection Boxes

Tasks

API-Heavy

Pandas

clean dataframe and compute stats using pandas

Algorithmic

Bigrams

find the most frequent bigram in a string

Fixed-Prompt

Open-Prompt

Box Plot

overlay scatter plot over boxplot using matplotlib

String Rewriting

parse rewrite rules and apply to a string

Participants

n = 17

Occupation:

- 15 academia
- 2 industry

Python Usage:

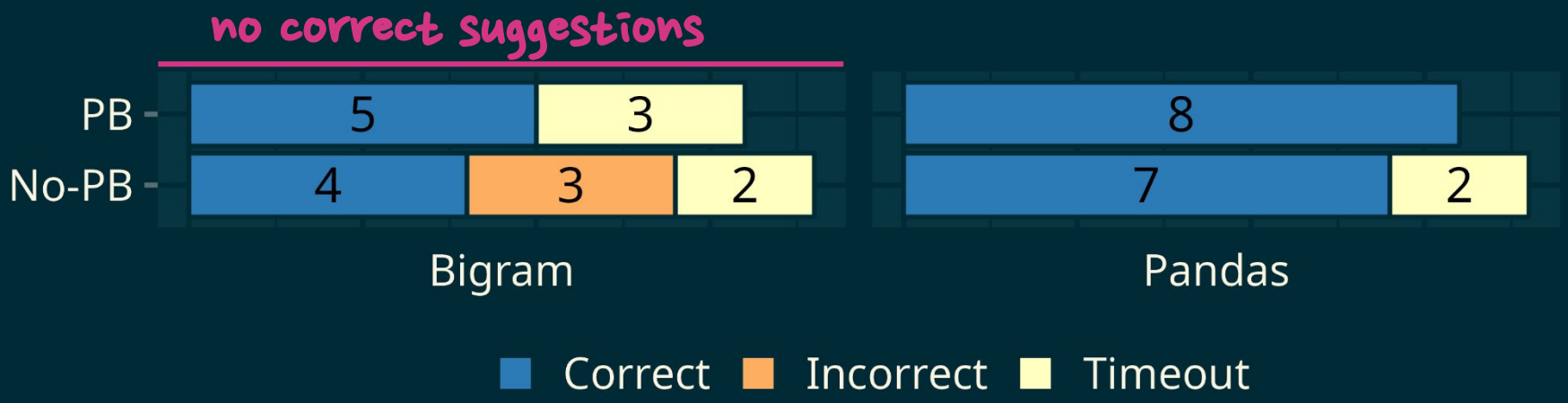
- 2 occasionally
- 8 regularly
- 7 almost every day

Overview

1. Motivation & Background:
    a. Grounded Copilot
    b. Live Programming
2. LEAP: Live Exploration of AI-Generated Code
3. User Study
4. Findings
    a. Validating suggestions
    b. Over-/Under-reliance
    c. Cognitive Load
    d. Impressions

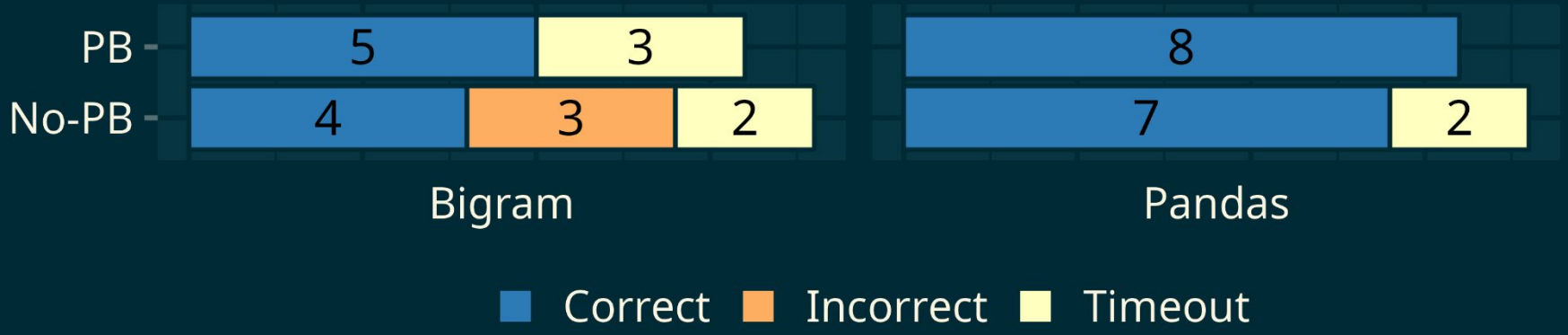
Slides based on Polikarpova 2023.

RQ1: Correctness



LEAP helps validate suggestions!
(But does not help fix incorrect ones)

RQ2: Over-/Under-reliance



6 no-PB vs 0 PB participants **mis-judged** correctness of their solutions

RQ2: Over-/Under-reliance

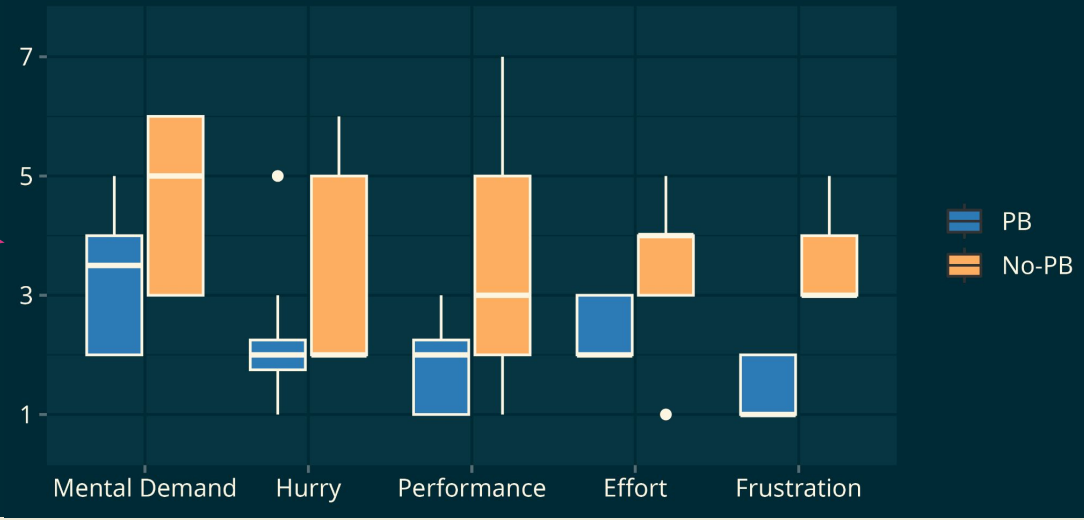
"it was **easy to understand** the behavior of a code suggestion because the little boxes on the side allowed for you to preview the results." (P3)

"it **saved me the effort** of writing multiple print statements." (P1)

LEAP reduces over-/under-reliance on AI,
by lowering the cost of validation.

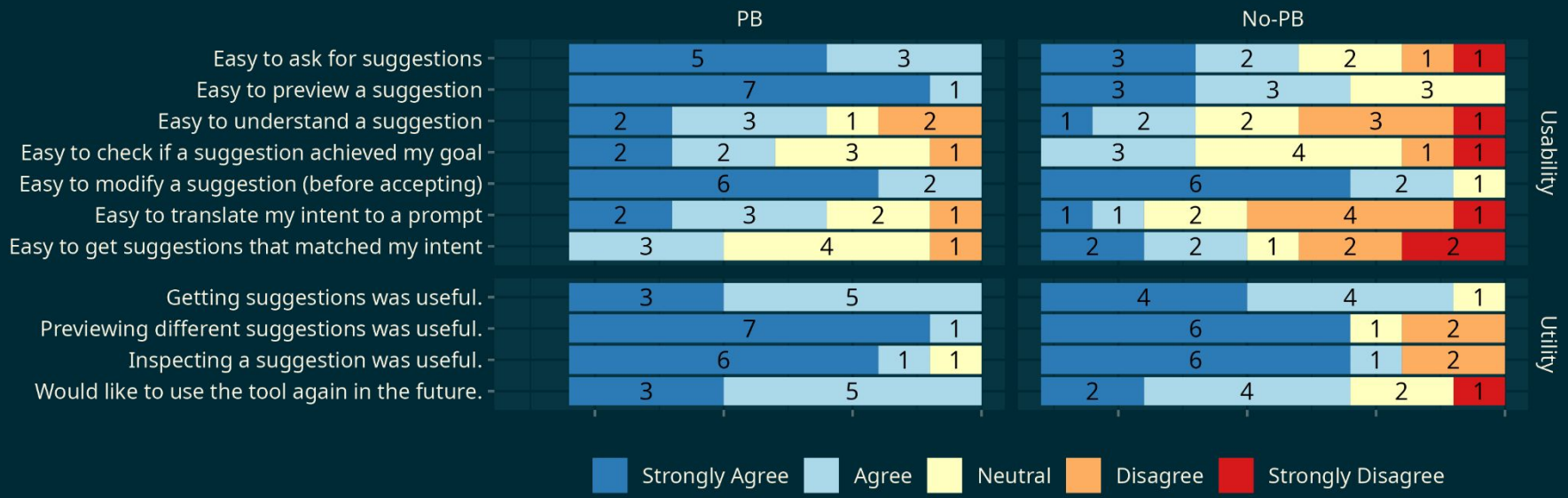
RQ3: Cognitive Load

NASA TLX metrics on the Pandas task



LEAP significantly reduced the cognitive load of exploring AI suggestions *on tasks amenable to validation by execution.*

RQ4: Users' Impressions



LEAP was more *usable* and more *useful*.

Overview

1. Motivation & Background:
    a. Grounded Copilot
    b. Live Programming
2. LEAP: Live Exploration of AI-Generated Code
3. User Study
4. Findings
    a. Validating suggestions
    b. Over-/Under-reliance
    c. Cognitive Load
    d. Impressions

Slides based on Polikarpova 2023.

Summary

1. Background:
    a. Live Programming, for
    b. Exploration of AI-Generated Code
2. LEAP: Projection Boxes + Copilot-like interface
3. User Study w/ 17 participants
4. Found that LEAP…
    a. Helps with validating suggestions,
    b. Reduces Over-/Under-reliance,
    c. Improves Cognitive Load, and
    d. Leaves a positive impressions on participants.