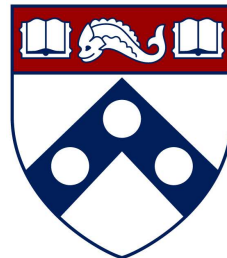


An Empirical Study of **Anonymous, Unmoderated, and Online** **Peer-to-Peer** Programming **Tutoring Conversations**

Priscila Santiesteban, **Emma Shedden**, Madeline Endres, Westley Weimer

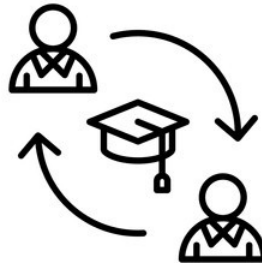


Penn
UNIVERSITY *of* PENNSYLVANIA

We learn CS outside of the classroom



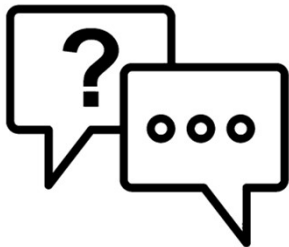
MOOCS:
Khan Academy



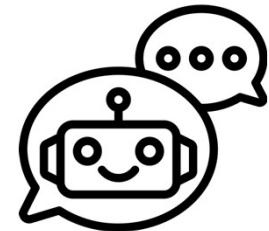
Peer tutoring:
PythonTutor



Chat rooms:
Discord



Forums:
Stack Overflow



AI assistants:
ChatGPT

Why study PythonTutor?

- Education research platform
- 10+ million users
- 10+ years online
- Novice-oriented
- Free to use, anonymous,



Total # unique IP addresses per country

Ten Million Users and Ten Years Later: Python Tutor's Design Guidelines for Building Scalable and Sustainable Research Software in Academia (P Guo, UIST 2021)

PythonTutor has tested a new functionality:

```
1 x = ['a', 'b', 'a', 'a', 'b']
2
3 def foo(ls):
4     pass # FIXME
5
6 foo(x)
```

Visualize execution

Python 3.11 

Get live help! [NEW]

Stop chat session

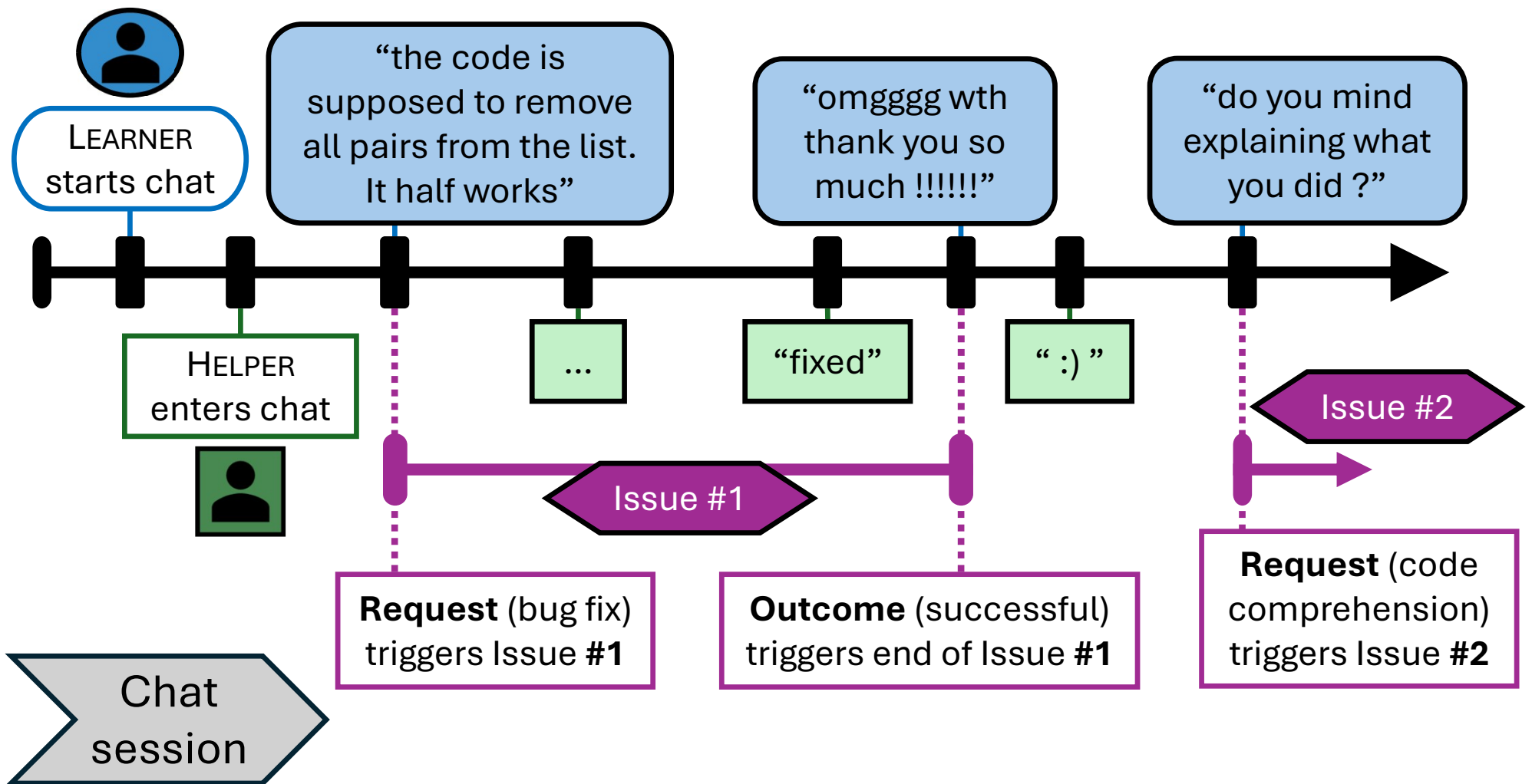
Chat

the code is supposed to remove all pairs from the list. It half works

...

LEARNER

HELPER



What we did:

- We are interested in the dynamics of peer tutoring conversation.
 - PythonTutor is a research education platform that supports this feature.
- We analyze 108 peer tutoring sessions (9K messages).
 - Most exhibit positive communication behaviors and outcomes.
- We find these analysis methods effective:
 - Qualitative analysis, two ways: conversation structure and content
 - GLMER analysis of annotated data

Who uses peer tutoring, and why?

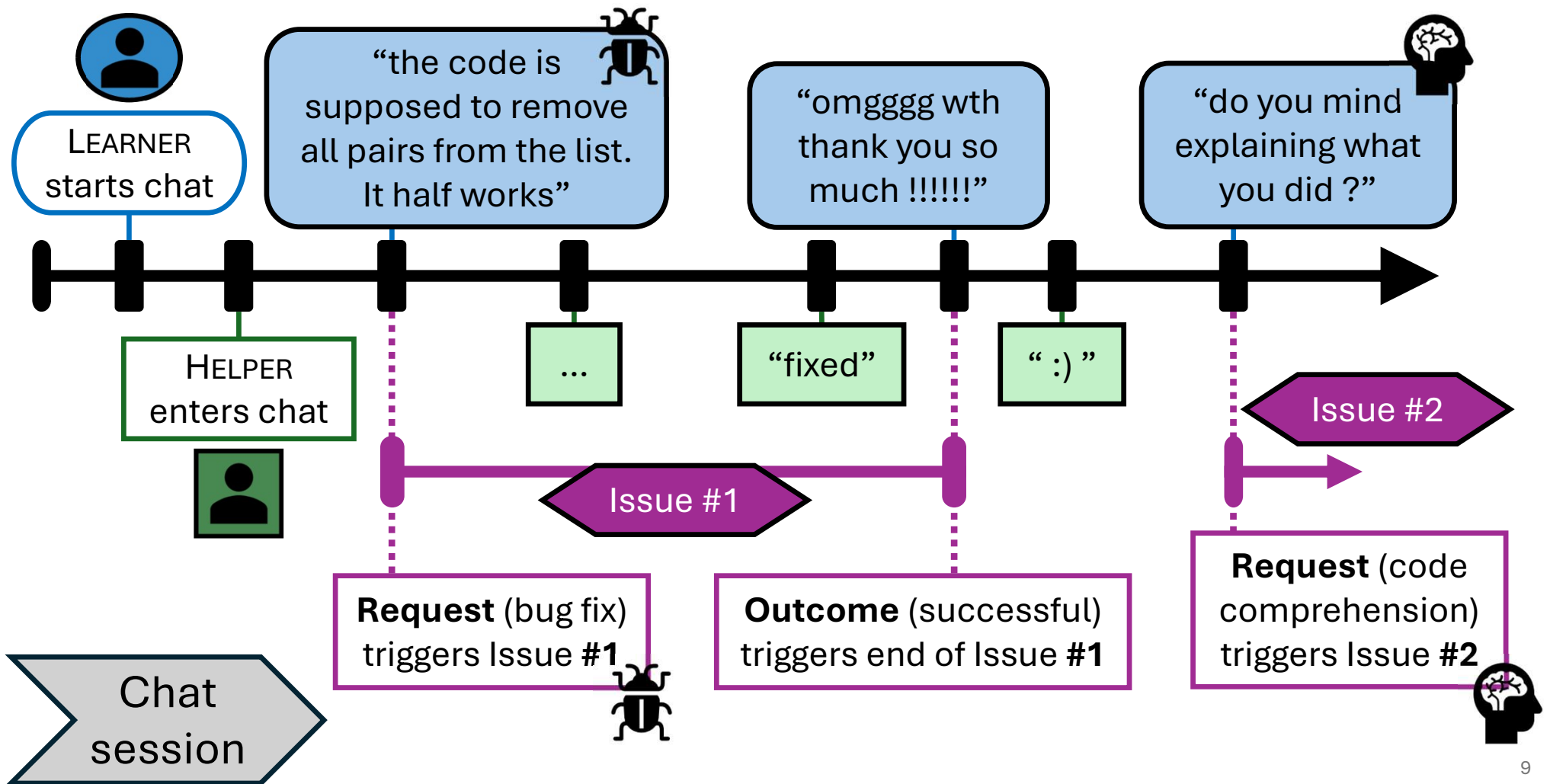
RQ1

88.0% of LEARNERS are Python novices

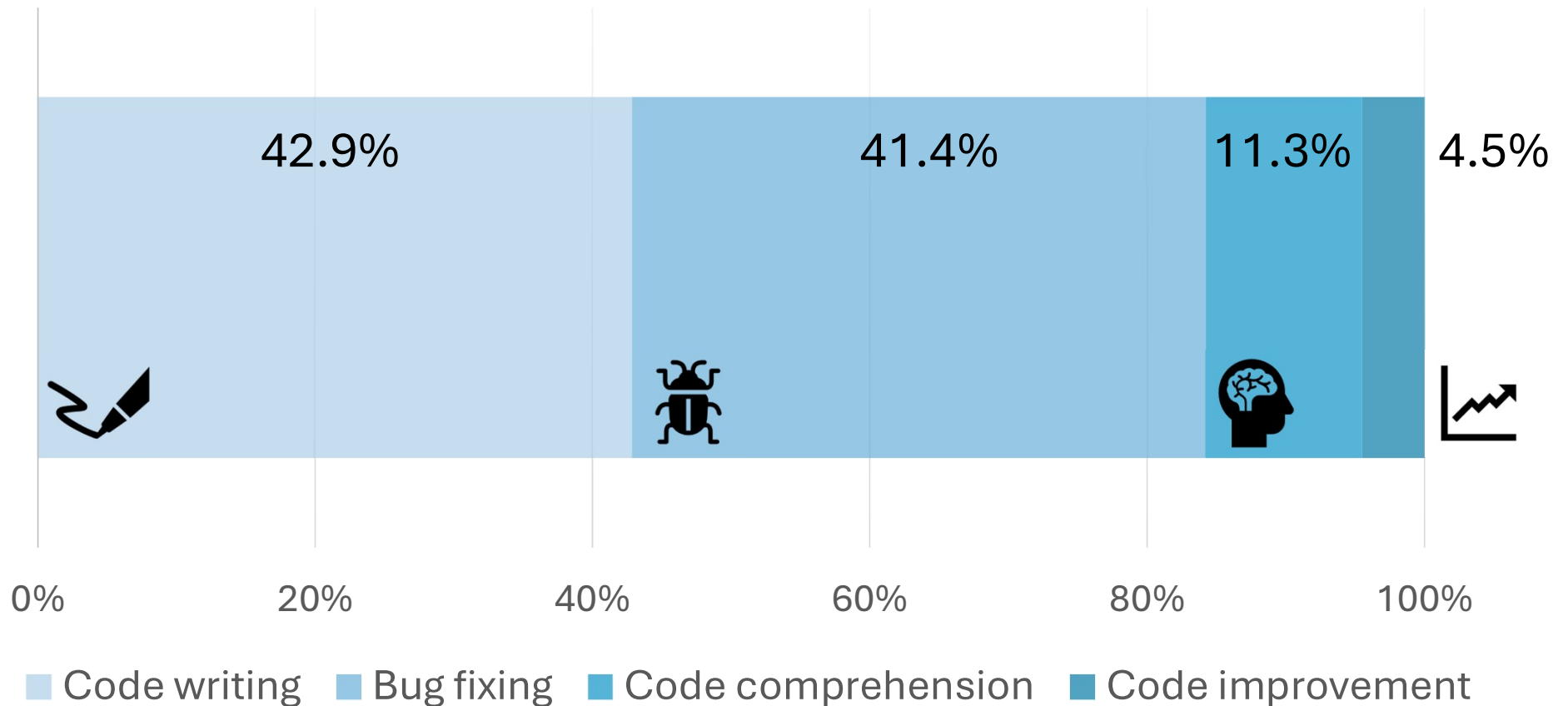
- They present CS1-level coding problems
 - Also, some LEARNERS call themselves novices, or say they don't understand a CS1 concept
- 25.9% of LEARNERS are enrolled in a traditional CS course
 - PythonTutor is a secondary resource

“the reading isn't explaining the python code very well”

“I went to office hours today and they somehow only confused me more”



They want help *writing* and *debugging* code



What happens in peer tutoring conversations?

RQ2

Users focus on code synthesis

We consider 14 topics of conversation:

- **Proposed new code**  is 13× as common as the others
- **Errors**  (location and message) are 0.21× as common



“instead using list I
recommend using dict”



“this is a question from an exam. im
having trouble understanding...”

Relating the *what* and the *how*

vs. the null

HELPERS
go above
and beyond

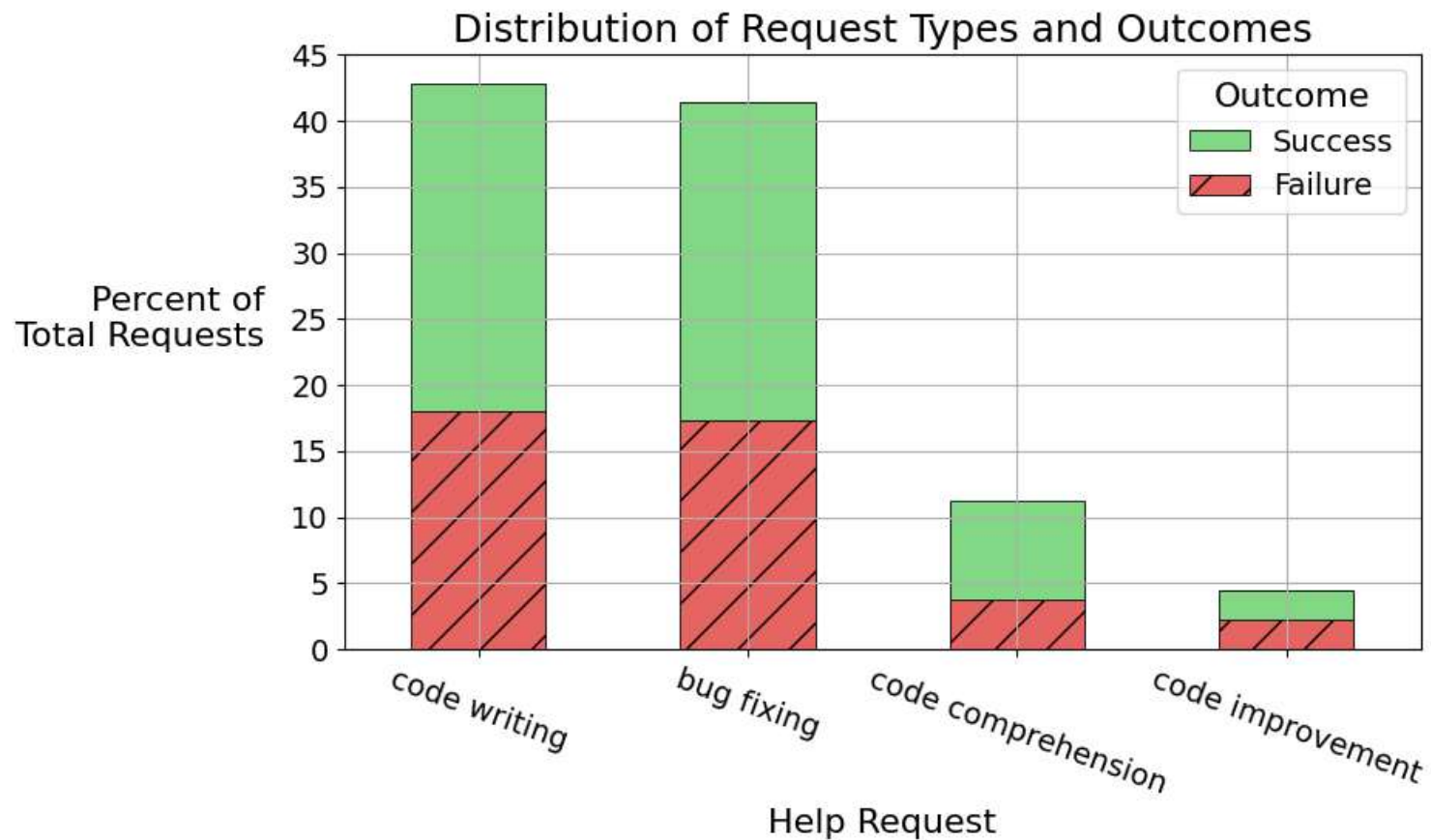
Topic	Method	Rate Ratio ×
Proposed new code 	Implement 	7.66
Coding concept 	Teach (extension) 	5.26
Learning resource 	Teach (extension) 	4.08

Null (rejected): Topics and methods of communication are independent

**When are LEARNERS satisfied with
a peer tutoring experience?**

RQ3

Success rates are similar across requests



Communication is indicative of outcome

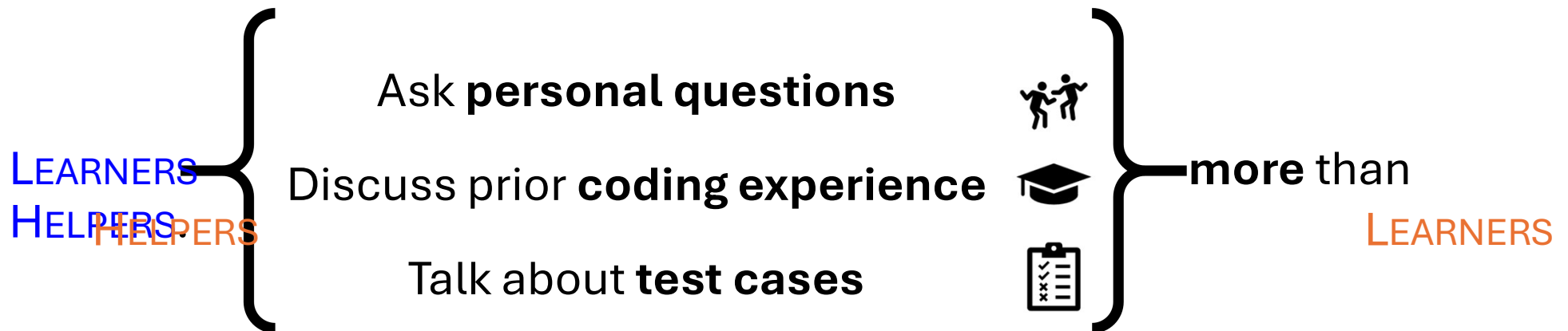
Successful interactions exhibit:

- 20% more direct implementation 
- 40% more gratitude 
- 29% less frustration 
- 28% less expressing that the user feels lost 

This suggests opportunities for real-time positive intervention.

Success depends on **who** says what

In **successful** conversations,
unsuccessful



In summary:

- RQ1
 - LEARNERS are novices who lack support from structured courses.
 - Despite being untrained and unmoderated, HELPERS achieve generally positive student satisfaction.
 - Details of program errors are under-communicated.
- RQ2
 - Code synthesis is most common
 - Concepts appear when HELPERS go above and beyond LEARNERS' asks.
- RQ3
 - Hands-on coding sessions are more successful.
 - Outcomes relate to *who* enacts certain behaviors.

See our paper and GitHub for more results and methods! [links/qr code]